

PORTADA Tapa del Libro

Application System/400

Guía del usuario RPG/400

Versión 3 Release 1.0

Número de Documento SC10-9426-00

Número de Programa
5763-RG1

AVISOS Avisos

+--- **Avisos** -----+
| Antes de utilizar esta información y el producto que soporta, debe |
| leer la información general que se encuentra bajo el apartado |
| "Advertencias" en el tema PORTADA_1. |
+-----+

VERSION Nota de Versión
Primera Edición (Junio 1994)

Esta publicación es la traducción del original inglés *RPG/400 User's Guide*, SC09-1816-00.

Esta edición es aplicable al programa bajo licencia IBM* ILE* RPG/400* (Programa 5763-RG1), Versión 3 Release 1 Modificación 0, así como a todos los releases y modificaciones posteriores a menos que se indique lo contrario en nuevas ediciones. Asegúrese de que la edición que utiliza sea la apropiada para el nivel del producto.

Solicite las publicaciones a través del representante de IBM o de la sucursal de IBM de su localidad. En la dirección que figura más abajo no hay existencias de publicaciones.

Se proporciona un formulario para comentarios del lector al final de esta publicación. Si no encuentra el formulario, puede dirigir sus comentarios a:

IBM S.A.
Centro de Traducción y Publicaciones
Avda. Diagonal, 571
08029 Barcelona
España

También puede enviar sus comentarios por fax a:

Desde España: (93) 209 11 16
Desde otros países: 34 3 209 11 16

Cuando envía información a IBM, otorga a IBM un derecho no exclusivo para utilizar o distribuir la información de la forma que crea apropiada sin incurrir en ninguna obligación con el remitente.

© Copyright International Business Machines Corporation 1994.
Reservados todos los derechos.

CONTENIDO	Contenido
PORTADA	Tapa del Libro
AVISOS	Avisos
VERSION	Nota de Versión
CONTENIDO	Contenido
PORTADA_1	Advertencias
PORTADA_1.1	Información de interfaz de programación
PORTADA_1.2	Marcas registradas y marcas de servicio
PORTADA_2	Acerca de este manual
PORTADA_2.1	Quién debe utilizar este manual
PORTADA_2.2	Cómo interpretar diagramas de sintaxis
1.0	Capítulo 1. Una introducción al RPG/400 y al sistema AS/400
1.1	El sistema OS/400
1.2	El lenguaje de control del AS/400
1.2.1	Mandatos del lenguaje de control utilizados con frecuencia
1.3	Entorno Sistema/38 en el sistema AS/400
1.4	Programas de utilidad y lenguajes del AS/400
1.4.1	El Programa de Utilidad para Entrada del Fuente
1.4.2	La ayuda para el diseño de pantallas
1.4.3	El lenguaje de consulta estructurada
1.4.3.1	Restricciones
1.5	Diseño de un programa en RPG/400
1.5.1	Diseño de la salida
1.5.2	Diseño del proceso
1.5.3	Diseño de la entrada
1.6	Programación estructurada en el lenguaje de programación RPG/400
1.6.1	Operación secuencial
1.6.2	Bifurcación condicional
1.6.2.1	Estructura If ELSE
1.6.2.2	Estructura SELEC
1.6.2.3	Otras estructuras de bifurcación condicional
1.6.3	Repetición de una operación
1.6.3.1	Operación DO
1.6.3.2	Operación DOW (Hacer mientras)
1.6.3.3	Operación DOU (Hacer hasta)
1.6.4	Resumen de códigos de operación de programación estructurada
1.7	Diseño de aplicaciones
1.7.1	Diseño de un programa único
1.7.2	Diseño de programa modular
1.7.3	Ejemplos de Diseño de aplicaciones
2.0	Capítulo 2. Entrada de especificaciones en RPG/400
2.1	Las especificaciones en RPG/400
2.1.1	La especificación de control
2.1.2	Especificaciones de descripción de archivo
2.1.3	Especificaciones adicionales
2.1.4	Especificaciones del contador de líneas
2.1.5	Especificaciones de entrada
2.1.6	Especificaciones de cálculo
2.1.7	Especificaciones de salida
2.2	Entrada de un programa
3.0	Capítulo 3. Compilación de un programa en RPG/400
3.1	Mandato crear programa en RPG400 (CRTRPGPGM)
3.1.1	Utilización del mandato CRTRPGPGM
3.1.1.1	Elementos de las líneas de mandatos de CRTRPGPGM
3.1.1.2	Entrada de elementos desde la pantalla de mandatos de CRTRPGPGM
3.1.1.3	Entrada de ciertos parámetros solamente
3.1.1.4	Entrada de valores de parámetros solamente
3.2	Mandato CRTRPGPGM
3.3	Compilación en el Entorno del Sistema/38
4.0	Capítulo 4. Mensajes de error, pruebas y depuración
4.1	Utilización, visualización e impresión de mensajes
4.1.1	Utilización de mensajes
4.1.2	Mensajes de señalización de Systems Application Architecture
4.1.3	Visualización e impresión de mensajes
4.2	Cómo ejecutar un programa en RPG/400
4.2.1	Soporte Salvar-mientras-activo
4.3	Utilización de una biblioteca de pruebas
4.4	Utilización de puntos de interrupción
4.4.1	Ejemplo de utilización de puntos de interrupción
4.4.2	Consideraciones al utilizar puntos de interrupción
4.5	Utilización de un rastreo
4.5.1	Ejemplo de utilización de un rastreo
4.5.2	Consideraciones al utilizar un rastreo
4.6	Utilización de códigos de operación DEBUG
4.7	Utilización del vucio con formato en RPG/400
4.8	Manejo de excepciones/errores
5.0	Capítulo 5. Consideraciones generales sobre archivos
5.1	Independencia de dispositivo/dependencia de dispositivo
5.2	Spooling
5.2.1	Spool de salida
5.3	Archivos descritos externamente y descritos por programa
5.4	Comprobación de nivel
5.5	Bloqueo de archivo por un programa en RPG/400
5.6	Bloqueo de registro por un programa en RPG/400

5.7	Desagrupación de bloques de registros de entrada y agrupación en bloques de registros de salida
5.8	Compartimiento de una vía de datos abierta
5.9	Utilización del mandato de language de control RCLSRC
5.10	Especificaciones para archivos descritos externamente
5.10.1	Especificaciones de descripción de archivo
5.10.2	Cambio de nombre de formatos de registro
5.10.3	Ignorado de formatos de registro
5.10.4	Campos de coma flotante
5.10.5	Alteración temporal o adición de funciones del RPG/400 a una descripción externa
5.10.6	Especificaciones de salida
5.11	Archivos descritos por programa
5.12	Archivos de impresora
5.12.1	Desbordamiento de página
5.12.2	Indicadores de desbordamiento
5.12.3	Lógica de obtención de desbordamiento
5.12.4	Opción PRTCTL (Control de impresión)
5.13	Archivo secuencial
5.14	Archivo especial
6.0	Capítulo 6. Control de compromiso
6.1	Utilización del control de compromiso
6.1.1	Comienzo y Fin del Control de Compromiso
6.1.2	Especificación de archivos para control de compromiso
6.1.3	Operaciones de control de compromiso
6.1.4	Bloqueo de control de compromiso
6.1.5	Control de compromiso en el ciclo del programa
6.1.6	Ejemplo de utilización del control de compromiso
7.0	Capítulo 7. Utilización de archivos DISK
7.1	Archivos DISK descritos externamente
7.1.1	Especificaciones de formato de registro
7.1.2	Vía de acceso
7.1.3	Claves válidas para un registro o archivo
7.1.3.1	Argumentos de búsqueda válidos
7.1.3.2	Referencia a una clave parcial
7.1.4	Métodos de proceso para archivos DISK descritos externamente
7.2	Archivos DISK descritos por programa
7.2.1	Archivo indexado
7.2.1.1	Argumentos de búsqueda válidos
7.2.2	Archivos secuenciales
7.2.3	Archivo de direcciones de registros
7.2.3.1	Registros de límites
7.2.3.2	Números relativos de registro
7.2.4	Archivo descrito externamente como archivo descrito por programa
7.3	Métodos para proceso de archivos DISK
7.3.1	Proceso por número relativo de registro
7.3.2	Proceso consecutivo
7.3.3	Proceso secuencial por clave
7.3.4	Proceso secuencial entre límites
7.3.5	Ejemplos de proceso por clave
7.4	Operaciones de archivo válidas
8.0	Capítulo 8. Utilización de archivos WORKSTN
8.1	Función para las comunicaciones entre sistemas
8.2	Archivos WORKSTN descritos externamente
8.2.1	Proceso de un archivo WORKSTN descrito externamente
8.2.2	Indicadores de tecla de función en archivos de dispositivo de pantalla
8.2.3	Teclas de mandato en archivos de dispositivo de pantalla
8.3	Proceso de archivos WORKSTN
8.3.1	Operación EXFMT
8.3.2	Operación READ
8.3.3	Operación WRITE
8.3.4	Archivo WORKSTN
8.3.5	Subarchivos
8.3.5.1	Utilización de subarchivos
8.4	Archivo WORKSTN descrito por programa
8.4.1	Archivo WORKSTN descrito por programa con un nombre de formato
8.4.1.1	Especificaciones de salida
8.4.1.2	Especificaciones de entrada
8.4.1.3	Especificaciones de cálculo
8.4.1.4	Consideraciones adicionales
8.4.2	Archivo WORKSTN descrito por programa sin un nombre de formato
8.4.2.1	Archivo de entrada
8.4.2.2	Archivo de salida
8.4.2.3	Archivo combinado
8.5	Archivos de múltiples dispositivos
8.6	Ejemplos de archivo WORKSTN
8.6.1	Programa de ejemplo 1-Consulta
8.6.2	Programa de ejemplo 2-Entrada de datos con actualización del maestro
8.6.3	Programa de ejemplo 3-Mantenimiento
8.6.4	Programa de ejemplo 4-Proceso de subarchivo WORKSTN
8.6.5	Programa de ejemplo 5-Consulta por código postal y búsqueda por nombre
8.6.6	Programa de ejemplo 6-Archivo WORKSTN descrito por programa con nombre de formato FORMATT
8.6.7	Programa de ejemplo 7-Línea de comienzo variable
8.6.8	Programa de ejemplo 8-Operación de lectura con límite de tiempo de espera excedido
9.0	Capítulo 9. Formatos de campos de datos y estructuras de datos
9.1	Formato de campos en archivos

9.1.1	Formato decimal empaquetado
9.1.2	Formato decimal con zona
9.1.3	Formato binario
9.1.3.1	Archivo descrito por programa
9.1.3.2	Archivo descrito externamente
9.1.4	Signos
9.1.4.1	Formatos externos
9.1.4.2	Formato interno
9.2	Estructuras de datos
9.2.1	Formato de subcampos de estructura de datos en almacenamiento
9.2.2	Especificaciones de sentencias de estructuras de datos
9.2.2.1	Reglas para la especificación de sentencias de estructuras de datos
9.2.3	Estructuras de datos de apariciones múltiples
9.3	Estructuras de datos especiales
9.3.1	Estructura de datos de área de datos
9.3.2	Estructura de datos de información de archivo
9.3.3	Estructura de datos de estado del programa
9.4	Especificaciones de subcampos de estructuras de datos
9.4.1	Reglas para especificaciones de subcampos
9.5	Ejemplos de estructura de datos
10.0	Capítulo 10. Constantes con nombre, inicialización y tipos de datos SAA
10.1	Constantes con nombre
10.1.1	Reglas para constantes con nombre
10.2	Inicialización
10.2.1	Subrutina de inicialización (*INZSR)
10.2.2	Códigos de operación CLEAR y RESET
10.2.3	Inicialización de estructuras de datos
10.2.4	Consideraciones especiales para la inicialización de estructuras de datos
10.2.5	Reglas para la inicialización de subcampos
10.2.6	Inicialización y el ciclo del programa
10.2.7	Ejemplos de inicialización
10.3	Tipos de datos SAA
10.3.1	Campos de longitud variable
10.3.2	Campos de fecha, hora e indicación de la hora
10.3.3	Soporte de tipo de datos gráficos DBCS
10.3.4	Soporte de valores nulos
10.3.5	Manejo de errores para tipos de datos SAA
11.0	Capítulo 11. Comunicación con objetos en el sistema
11.1	Llamada a otros programas
11.1.1	CALL (Llamada a un programa)
11.1.2	PLIST (Identificar una lista de parámetros) y PARM (Identificar parámetros)
11.1.2.1	Reglas para la especificación de PLIST
11.1.2.2	Reglas para la especificación de PARM
11.1.3	Soporte gráfico del OS/400
11.1.4	FREE (Desactivar un programa)
11.2	Llamada a subrutinas especiales
11.2.1	Subrutina de recuperación de mensajes (SUBR23R3)
11.2.2	Soporte de la interfaz común de programación SAA
11.2.3	Traslado de datos de doble byte entre corchetes y supresión de caracteres de control (SUBR23R3)
11.2.4	Traslado de datos de doble byte entre corchetes y adición de caracteres de control (SUBR23R3)
11.3	Retorno desde un programa llamado
11.3.1	Una terminación normal
11.3.2	Una terminación anormal
11.3.3	Retorno sin una terminación
11.4	Áreas de datos
11.4.1	Área de datos de parámetros de inicialización del programa (PIP)
12.0	Capítulo 12. El generador automático de informes
12.1	Impresión por grupos
12.1.1	Especificaciones
12.1.2	Ejemplos
12.2	Especificaciones de la sentencia /COPY
12.2.1	Modificación de especificaciones copiadas
12.2.1.1	Modificación de especificaciones de descripción de archivos
12.2.1.2	Modificación de especificaciones de campos de entrada
12.3	Formato de generador de informes
12.3.1	Espaciado y Salto
12.3.2	Ubicación de encabezamientos y campos
12.3.2.1	Encabezamientos de página
12.3.2.2	Reasignación de formato de encabezamientos de página *AUTO
12.3.2.3	Cuerpo del informe
12.3.2.4	Desbordamiento de líneas de impresión D/T-*AUTO
12.4	Especificaciones generadas
12.4.1	Cálculos generados
12.4.2	Especificaciones de salida generadas
12.5	Ayudas de programación
12.6	Utilización de CRTRPTPGM para compilar un programa generador automático de informes
12.6.1	Utilización del mandato CRTRPTPGM
12.6.2	Mandato CRTRPTPGM
12.7	Ejemplos de utilización del generador automático de informes
12.7.1	EJEMPLO 1 - Informe de ventas
12.7.2	EJEMPLO 2 - Informe de ventas con tres niveles de totales
12.7.3	EJEMPLO 3 - Informe de ventas con indicación de grupo
12.7.4	EJEMPLO 4 - Informe de ventas con totales en columnas cruzadas
12.7.5	EJEMPLO 5 - Informe de ventas utilizando especificaciones copiadas

12.7.6	EJEMPLO 6 - Alteración temporal de especificaciones de entrada copiadas
13.0	Capítulo 13. Programas de ejemplo RPG/400
13.1	Lista de comprobación de los ejemplos del programa
13.1.1	Diseño de la base de datos
13.1.1.1	Archivo maestro de empleados
13.1.1.2	Archivo maestro de proyectos
13.1.1.3	Archivo maestro de códigos de razón
13.1.2	Archivos históricos de transacciones
13.1.3	Archivo de control del área de datos
13.1.4	Mantenimiento del archivo maestro
13.1.5	Mantenimiento del archivo de control del área de datos
13.1.6	Entrada del archivo de tiempos
13.1.7	Actualización semanal del archivo de tiempos
13.1.8	Información y actualización mensual del archivo de entrada de tiempos
13.2	Definición de campo de base de datos
13.2.1	Archivo maestro de referencias de la base de datos - REFMST
13.2.2	Archivo de control del área de datos - CTLFIL
13.2.3	Archivo maestro de empleados - EMPMST
13.2.4	Archivo maestro de proyectos - PRJMST
13.2.5	Archivo maestro de códigos de razón - RSNMST
13.2.6	Archivo de entrada de transacciones semanal - TRWEEK
13.2.7	Archivo de entrada de transacciones mensual - TRMNTH
13.3	Diseño del menú de información de tiempos
13.4	Mantenimiento del archivo maestro
13.4.1	Pantalla Mantenimiento del Archivo Maestro - PRG01FM
13.4.1.1	Formato SELECT - Selección de mantenimiento
13.4.2	Selección del maestro de empleados - Formato EMPSEL
13.4.3	Mantenimiento del maestro de empleados - Formato EMPMNT
13.4.4	Selección del maestro de proyectos - Formato PRJSEL
13.4.5	Mantenimiento del maestro de proyectos - Formato PRJMNT
13.4.6	Selección del maestro de códigos de razón - Formato RSNSEL
13.4.7	Mantenimiento del maestro de códigos de razón - Formato RSNMNT
13.4.8	Descripciones de datos de mantenimiento del archivo maestro - PRG01FM
13.4.9	Programa en RPG/400 de mantenimiento de archivos maestros - PRG01
13.5	Mantenimiento del archivo de control
13.5.1	Mantenimiento del archivo de control - PRG02FM
13.5.2	Descripciones de datos del mantenimiento del archivo de control - PRG02FM
13.5.3	Programa en RPG/400 de Mantenimiento del archivo de control - PRG02
13.6	Entrada de transacciones del archivo de tiempos
13.6.1	Entrada de transacciones de información de tiempos - PRG03FM
13.6.1.1	Pantalla de Selección de Empleados
13.6.2	Descripciones de datos de entrada de transacciones de información de tiempos - PRG03FM
13.6.3	Programa en RPG/400 de entrada de transacciones de información de tiempos - PRG03
13.7	Actualización semanal del archivo de tiempos
13.7.1	Programa RPG/400 de Edición de entradas del archivo de tiempos - PRG05
13.7.2	Diseño del informe semanal de transacciones de empleados - PRG09
13.7.3	Actualización del archivo maestro e Informe semanal de transacciones - PRG09
13.8	Proceso mensual
13.8.1	Actualización e información mensual del archivo de tiempos
13.8.2	Diseño del informe de resumen de la información de tiempos de empleados - PRG06RP
13.8.3	Descripciones de datos del informe de resumen de empleados -PRG06RP
13.8.4	Programa RPG/400 de informe de resumen de empleados - PRG06
13.8.5	Diseño del informe de resumen del proyecto de información de tiempos - PRG07RP
13.8.6	Descripciones de los datos del informe de resumen del proyecto - PRG07RP
13.8.7	Programa RPG/400 de informe de resumen del proyecto - PRG07
13.8.8	Diseño del informe de resumen de información de tiempos de códigos de razón - PRG08RP
13.8.9	Descripciones de datos del informe de resumen de códigos de razón - PRG08RP
13.8.10	Programa RPG/400 de informe de resumen de códigos de razón - PRG08
13.8.11	Programa RPG/400 de actualización y borrado mensual del archivo maestro - PRG04
13.9	Proceso de final de año
A.0	Apéndice A. Información de servicio del programa generador automático de informes y del con
A.1	Visión general del compilador
A.1.1	Fases del compilador
A.1.2	Principales áreas de datos del compilador
A.1.3	Organización de los mensajes de error del compilador
A.1.4	Subrutinas en tiempo de ejecución
A.2	Opciones de depuración del compilador
A.2.1	Valor *SOURCE para el parámetro OPTION
A.2.2	Valor *XREF para el parámetro OPTION
A.2.3	Valor *DUMP para el parámetro OPTION
A.2.4	Valor *LIST para el parámetro GENOPT
A.2.5	Valor *ATR para el parámetro GENOPT
A.2.6	Valor *XREF para el parámetro GENOPT
A.2.7	Valor *DUMP para el parámetro GENOPT
A.2.8	Valor *PATCH para el parámetro GENOPT
A.2.9	Valor *OPTIMIZE para el parámetro GENOPT
A.2.10	Parámetro ITDUMP
A.2.11	Parámetro SNPDUMP
A.2.12	Parámetro CODELIST
A.2.13	Parámetro PHSTRC
A.3	Ejemplos de utilización de opciones de depuración del compilador
A.4	Diseño de la IRP
A.5	Programa generador automático de informes
B.0	Apéndice B. RPG/400 y funciones del RPG II del AS/400 compatibles con el Sistema/36

Guía del usuario RPG/400
Contenido

B.1	Mejoras en el lenguaje
C.0	Apéndice C. Comunicación de datos
C.1	Manejo de excepciones y errores con archivos ICF
C.2	Recuperación de errores de comunicaciones
D.0	Apéndice D. Archivos de gestión de datos distribuidos (DDM)
E.0	Apéndice E. Opción del entorno del Sistema/38 del compilador RPG
E.1	Diferencias entre el RPG III del Sistema/38 y la Opción del entorno del Sistema/38 del com
E.2	Diferencias entre la Opción del entorno del Sistema/38 del compilador RPG y el compilador
E.3	Tipos de archivo soportados por cada compilador
F.0	Apéndice F. Ejemplos de la utilización de matrices
G.0	Apéndice G. Glosario de abreviaturas
BACK_1	Bibliografía
INDICE	Índice
COMENTARIOS	Hoja de Comentarios

PORTADA 1 Advertencias

Las referencias que hace esta publicación a productos, programas o servicios IBM, no implican que IBM piense ponerlos a la venta en todos los países en los que opera IBM. Las referencias a productos, programas o servicios IBM no afirman ni implican que únicamente pueda utilizarse el producto, programa o servicio IBM. Puede utilizarse cualquier producto programa o servicio funcionalmente equivalente, que no infrinja ningún derecho de propiedad intelectual de IBM, en lugar del producto, programa o servicio IBM. La evaluación y verificación de la operación conjunta con otros productos, excepto aquellos expresamente designados por IBM, son responsabilidad del usuario.

IBM puede tener patentes o solicitudes de patente pendientes que se refieren a temas de este documento. El hecho de que se le proporcione este documento no le otorga ninguna licencia sobre dichas patentes. Puede enviar sus preguntas sobre licencias por escrito a IBM Director of Licensing, IBM Corporation, 208 Harbor Drive, Stamford, Connecticut, USA |06904-2501.

Los cambios o adiciones de texto se indican con una línea vertical (|) a la izquierda del cambio o adición.

Esta publicación contiene ejemplos de datos e informes que se utilizan diariamente en operaciones comerciales. Para ilustrarlos lo más completamente posible, los ejemplos incluyen nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier parecido con nombres y direcciones utilizados por una empresa comercial real es pura coincidencia.

Subtemas

PORTADA_1.1 Información de interfaz de programación

PORTADA_1.2 Marcas registradas y marcas de servicio

PORTADA_1.1 Información de interfaz de programación

El manual RPG/400 Guía del Usuario está pensado para ayudarle a crear programas RPG/400. RPG/400 Guía del Usuario presenta información sobre *interfaces de programación de uso general y otras ayudas adicionales* proporcionada por el compilador RPG/400.

Las interfaces de programación de uso general permiten al cliente escribir programas que soliciten o reciban servicios del compilador RPG/400.

PORTADA_1.2 Marcas registradas y marcas de servicio

Los términos siguientes, indicados por un asterisco (*), que se utilizan en esta publicación, son marcas registradas de IBM Corporation en Estados Unidos u otros países:

Application System/400	AS/400
IBM	ILE
Operating System/2	Operating System/400
OS/2	OS/400
RPG/400	SAA
Systems Application Architecture	SQL/400
400	

PORTADA_2 Acerca de este manual

Este manual es una guía para el lenguaje de programación RPG/400* en un sistema AS/400 que utilice el sistema Operating System/400* (OS/400*). El compilador RPG/400 es un compilador de Systems Application Architecture* (SAA*) que se adhiere a los convenios SAA.

Los temas cubiertos en este manual incluyen:

- Diseño de programas en RPG/400
- Codificación de programas en RPG/400
- Entrada y compilación de programas en RPG/400
- Prueba y depuración de programas en RPG/400
- Estudio de programas de muestra y ejemplos codificados en RPG/400.

Este manual puede hacer referencia a productos que se anuncian pero todavía no están disponibles.

Puede que necesite consultar otros manuales de IBM* para obtener información más detallada sobre un tema concreto. La publicación *Guía de Publicaciones*, GC10-9237 (GC41-9678), proporciona información sobre todos los manuales de la biblioteca del AS/400*. Para obtener una lista de publicaciones relacionadas, consulte "Bibliografía" en el tema BACK_1.

Subtemas

PORTADA_2.1 Quién debe utilizar este manual

PORTADA_2.2 Cómo interpretar diagramas de sintaxis

PORTADA_2.1 Quién debe utilizar este manual

Este manual está pensado para gente con conocimientos básicos sobre el proceso de datos y el lenguaje de programación RPG/400. También está diseñado para guiar al programador en la utilización de programas en RPG/400 y compiladores del sistema AS/400. A menudo se mencionan operaciones y especificaciones de RPG/400. Para obtener una descripción detallada de los códigos de operación y las especificaciones de RPG/400, consulte el manual *RPG/400 Reference*, SC09-1817.

Antes de utilizar este manual, debe estar familiarizado con cierta información:

- Debe saber cómo utilizar el soporte de gestión de datos para trabajar con archivos, estaciones de pantalla, impresoras, cintas y disquetes, así como el soporte en spool. Esta información está contenida en la publicación *Guía para la Gestión de Datos*.
- Debe estar familiarizado con su estación de pantalla (también conocida como estación de trabajo) y sus controles. Algunos elementos de la pantalla y ciertas teclas del teclado son estándares con independencia del sistema de software que se esté ejecutando en el estación de pantalla o el sistema de hardware al que la estación de pantalla esté conectada. A continuación aparecen algunas de estas teclas:
 - Teclas de movimiento de cursor
 - Teclas de mandatos
 - Teclas de salida de campo
 - Teclas de inserción y supresión
 - Tecla de restauración de error.

Esta información está contenida en la publicación *Guía para Nuevos Usuarios*, SC10-8881 (SC41-8211).

- Debe saber cómo operar la estación de pantalla cuando está conectada al sistema IBM AS/400 y ejecuta el software del AS/400. Esto significa tener conocimientos acerca del sistema operativo y del Lenguaje de Control (CL) para realizar tareas como por ejemplo:
 - Iniciar y terminar una sesión en AS/400
 - Operar de forma interactiva con las pantallas
 - Utilizar la ayuda
 - Entrar mandatos de control
 - Llamar a programas de utilidad
 - Responder a mensajes.

Para mayor información sobre el lenguaje de control, consulte las siguientes publicaciones del IBM AS/400:

- *CL Guía del Programador*
- *Control Language Reference*

- Debe estar familiarizado con el ciclo de programas RPG/400, con la forma en que los indicadores afectan al ciclo de programas y con la manera de codificar entradas en las hojas de especificaciones del RPG/400.

Los programas de ejemplo de aplicación contenidos en este manual se han hecho a escala de tal forma que pueda utilizar la publicación *RPG Debugging Template*, GX21-9129 para comprobar los programas.

Estas consideraciones generales sobre el lenguaje de programación RPG/400 se explican en un curso de codificación en RPG/400. Para obtener información más detallada sobre el lenguaje de programación RPG/400 consulte la publicación *RPG/400 Reference*.

PORTADA_2.2 Cómo interpretar diagramas de sintaxis

Los diagramas de sintaxis de este manual utilizan los convenios siguientes:

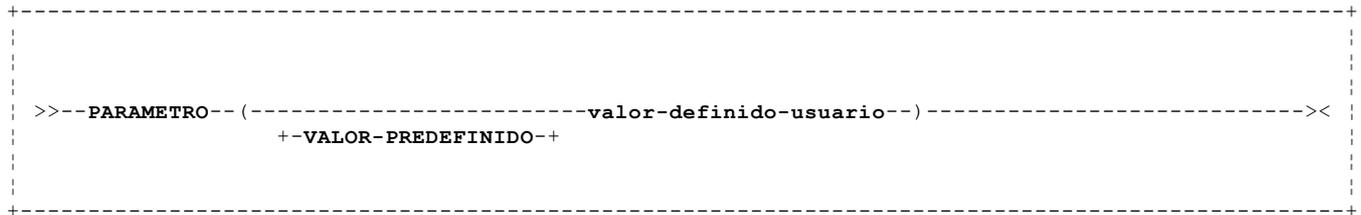


Figura 1. Estructura de un Diagrama de sintaxis

Lea el diagrama de sintaxis de izquierda a derecha, de arriba a abajo, siguiendo la ruta de la línea.

Dos puntas de flecha seguidas por una línea continua indica el principio del diagrama de sintaxis.

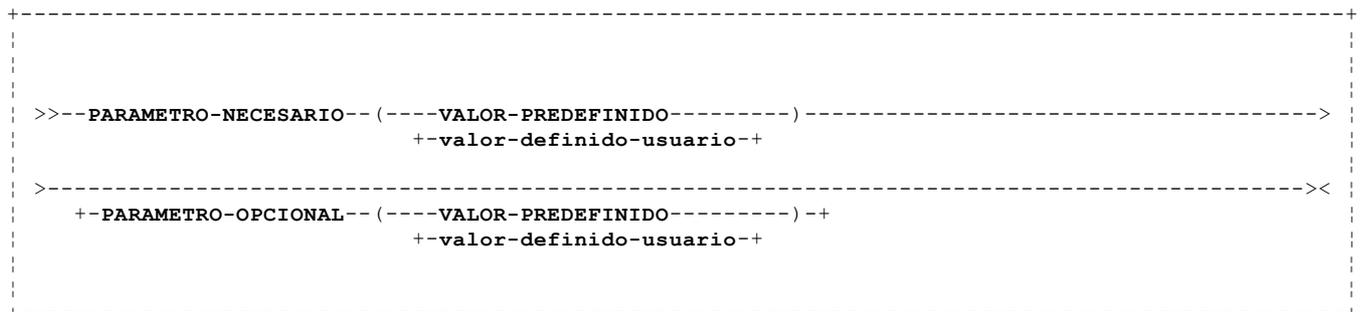
Una línea continua que finaliza con una punta de flecha a la derecha y una punta de flecha a la izquierda indica el final del diagrama de sintaxis.

Una línea continua que lleva a una punta de flecha a la derecha indica que la sintaxis de la sentencia continúa en la línea siguiente.

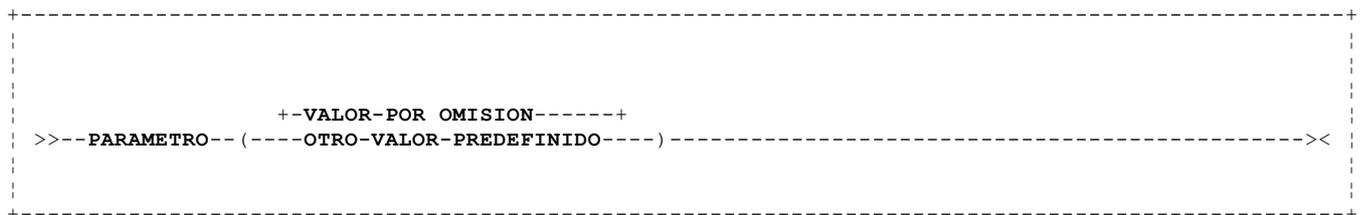
Una punta de flecha a la derecha que lleva a una línea continua indica que continúa una sentencia de la línea anterior.

Un parámetro o valor entre abrir y cerrar paréntesis indica que el parámetro o valor debe entrarse entre paréntesis.

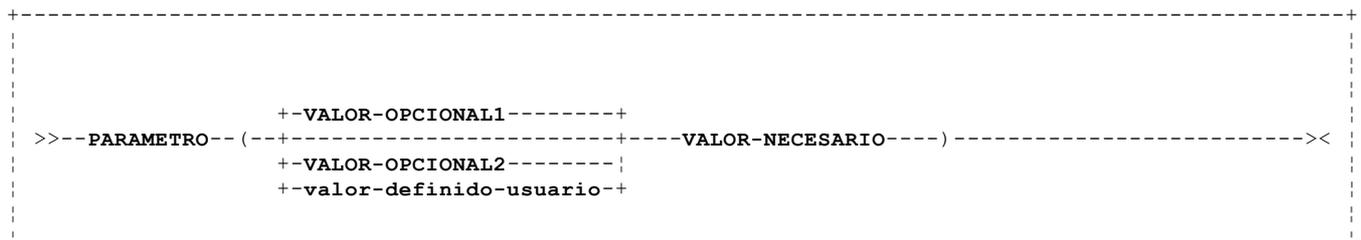
Los **parámetros necesarios** aparecen en la línea base y deben entrarse. Los **parámetros opcionales** aparecen por debajo de la línea base y no tienen que entrarse. En el ejemplo siguiente, el usuario debe entrar el PARAMETRO-NECESARIO y un valor para el mismo, pero no tiene que entrar el PARAMETRO-OPCIONAL o un valor para el mismo.



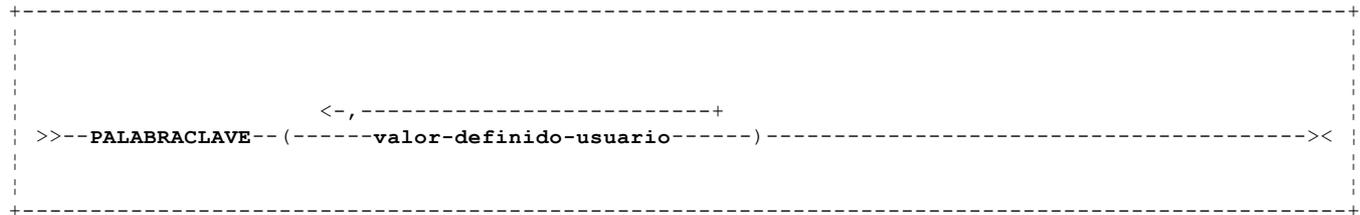
Los **valores por omisión** aparecen por encima de la línea base y no tienen que entrarse. Se utilizan cuando no se especifica un parámetro. En el ejemplo siguiente, el usuario puede entrar VALOR-POR-OMISION, OTRO-VALOR-PREDEFINIDO o nada. Si no se entra nada, se supone que es VALOR-POR-OMISION.



Los **valores opcionales** se indican por una línea en blanco. La línea en blanco indica que no tiene que entrarse un valor del primer grupo (VALOR-OPCIONAL1, VALOR-OPCIONAL2, *valor-definido-usuario*). Por ejemplo, basándose en la sintaxis siguiente, el usuario podría entrar: PALABRACLAVE (VALOR-NECESARIO).



+-----+
Los **valores repetidos** pueden especificarse para algunos parámetros. La ,
del ejemplo siguiente indica que cada *valor-definido-usuario* debe
separarse por una coma.



1.0 Capítulo 1. Una introducción al RPG/400 y al sistema AS/400

El lenguaje de programación RPG/400 está diseñado para facilitar la creación de aplicaciones de software para negocios.

El RPG es un lenguaje en evolución. Está disponible en cada máquina que le da soporte una versión algo diferente del RPG. El sistema AS/400 es el más reciente de estos sistemas informáticos. Debe saber que, además de ofrecer una nueva versión mejorada del RPG, el sistema AS/400 también da soporte a la versión anterior de RPG disponible en el sistema Sistema/38 y en el sistema Sistema/36. Para obtener más información, consulte Apéndice B, "RPG/400 y funciones del RPG II del AS/400 compatibles con el Sistema/36", y Apéndice E, "Opción del entorno del Sistema/38 del compilador RPG".

Este capítulo proporciona una visión general de los siguientes temas:

- El sistema OS/400 y el lenguaje de control
- Funciones del RPG/400 en el sistema AS/400
- El Entorno del Sistema/38 en el sistema AS/400
- Programas de utilidad y lenguajes disponibles
- El ciclo de programación RPG/400
- Diseño de programas en RPG/400
- Programación estructurada en programas RPG/400
- Diseño de aplicaciones.

Subtemas

- 1.1 El sistema OS/400
- 1.2 El lenguaje de control del AS/400
- 1.3 Entorno Sistema/38 en el sistema AS/400
- 1.4 Programas de utilidad y lenguajes del AS/400
- 1.5 Diseño de un programa en RPG/400
- 1.6 Programación estructurada en el lenguaje de programación RPG/400
- 1.7 Diseño de aplicaciones

1.1 El sistema OS/400

El sistema operativo que controla todas las interacciones con el sistema AS/400 se denomina sistema Operating System/400 (OS/400). Desde la estación de trabajo, el sistema OS/400 le permite :

- Iniciar y terminar una sesión
- Operar de forma interactiva con las pantallas
- Utilizar la información de ayuda en línea
- Entrar mandatos de control y procedimientos
- Responder a mensajes
- Gestionar archivos
- Ejecutar programas de utilidad y otros programas

En la publicación *Guía de Publicaciones* encontrará una lista completa de las publicaciones sobre el sistema OS/400.

1.2 El lenguaje de control del AS/400

Puede manipular el sistema OS/400 mediante el lenguaje de control (CL). Puede operar de forma interactiva con el sistema entrando o seleccionando mandatos CL. El sistema AS/400 visualiza en muchos casos una serie de mandatos CL o parámetros de mandato adecuados para la situación en la pantalla. Entonces puede seleccionar el mandato o los parámetros que desee.

Subtemas

1.2.1 Mandatos del lenguaje de control utilizados con frecuencia

1.2.1 Mandatos del lenguaje de control utilizados con frecuencia

La tabla siguiente lista algunos de los mandatos CL utilizados más frecuentemente así como el motivo por el que podría utilizarlos.

Tabla 1. Funciones del RPG/400 y mandatos CL asociados		
Función RPG/400	Mandatos del lenguaje de control asociados y su utilización	
Llamar	CALL nombre de programa	Ejecutar un programa en RPG/400
	CALL QCL	Acceder al entorno Sistema/38
Control de compromiso	CRTJRN	Preparar utilización de control de compromiso.
	CRTJRNRCV	Preparar utilización de control de compromiso.
	ENDCMTCTL	Comunicar al sistema que desea finalizar el control de compromiso.
	JRNPF	Preparar utilización de control de compromiso.
	STRCMTCTL	Comunicar al sistema que desea comenzar el control de compromiso.
Comunicaciones	CRTICFDEVE	Crear dispositivo ICF
	OVRICFDEVE	Alterar temporalmente dispositivo ICF
Compilar	CRTRPGPGM	Crear programa en RPG
	CRTRPPTGM	Crear programa generador de informe automático
Proceso consecutivo	OVRDBF	Alterar temporalmente con archivo de base de datos
Area de datos de especificación de control	CRTDTAARA	Crear área de datos
	DSPDTAARA	Visualizar área de datos
Depurar	ADDBKP	Añadir punto de interrupción
	ADDTRC	Añadir rastreo
	DSPBKP	Visualizar punto de interrupción
	STRDBG	Arrancar depuración
Códigos de edición	CRTEDTD	Crear descripción de edición (para códigos de edición definidos por el usuario)
	DSPDTAARA	Visualizar área de datos
Archivos de impresora	CRTPRTF	Crear archivo de impresora
	OVRPRTF	Alterar temporalmente archivo de impresora
Editor del sistema	STRSEU	Arrancar programa de utilidad para entrada de fuente

El lenguaje de control y todos los mandatos el mismo aparecen descritos detalladamente en el manual *CL Reference*.

1.3 Entorno Sistema/38 en el sistema AS/400

El sistema AS/400 ofrece una función mejorada con relación al Sistema/38. Como muchos programas en lenguaje RPG/400 están escritos para el Sistema/38, y muchos programadores ya están familiarizados con el Sistema/38, el sistema AS/400 también da soporte a estos programas del Entorno del Sistema/38. El mandato de CL CALL QCL modifica la pantalla del sistema AS/400 para que semeje una pantalla del Sistema/38. Esta se conoce con el nombre de Entorno del Sistema/38. Cuando se encuentre en este entorno, puede entrar y compilar programas en RPG/400 como si estuviera utilizando un Sistema/38. Los convenios de denominación de archivos son los mismos que en Sistema/38. También puede entrar mandato CL del AS/400 en el Entorno del Sistema/38. Puede entrar mandatos del Entorno del Sistema/38 desde el sistema AS/400 utilizando mandatos de calificación de bibliotecas. El mandato QSYS38/CRTTRPGPGM llama al compilador RPG III del Entorno del Sistema/38. Para obtener más información sobre el Entorno del Sistema/38, consulte el manual *System/38 Environment Programmer's Guide/Reference*.

Puede utilizar el Programa de Utilidad para Entrada del Fuente (SEU) para entrar el programa fuente del RPG/400 de forma interactiva. Entre el mandato CL STRSEU para llamar al SEU. Si especifica el parámetro TYPE(RPG) en este mandato, se llama al comprobador sintáctico del RPG/400, el cual detecta los errores de sintaxis del RPG/400, sentencia a sentencia, mientras se entra el programa fuente. Otra alternativa sería entrar un programa fuente en disquetes y subir el programa a un archivo fuente.

```
+--- Advertencia -----+
|
| Para informarse sobre cómo utilizar el RPG III en el Entorno del
| Sistema/38, consulte:
|
| □ Apéndice E, "Opción del entorno del Sistema/38 del compilador RPG"
|   en el tema E.0
|
| □ la publicación Sistema/38 RPG III Reference Manual and
|   Programmer's Guide SC21-7725.
|
| Para obtener información acerca de los dispositivos y mandatos del
| Sistema/38 consulte los manuales adecuados de la biblioteca del
| Sistema/38.
|
+-----+
```

1.4 Programas de utilidad y lenguajes del AS/400

El sistema AS/400 ofrece dos programas de utilidad y un lenguaje que le puede resultar útil para programar. Se trata del programa de utilidad Ayuda para el Diseño de Pantallas (SDA), el Programa de Utilidad para Entrada del Fuente (SEU), y el Lenguaje de Consulta Estructurada (SQL).

Subtemas

- 1.4.1 El Programa de Utilidad para Entrada del Fuente
- 1.4.2 La ayuda para el diseño de pantallas
- 1.4.3 El lenguaje de consulta estructurada

1.4.1 *El Programa de Utilidad para Entrada del Fuente*

Utilice el SEU para entrar su código en el sistema. El SEU también proporciona un comprobación de sintaxis exhaustiva. Para obtener más información sobre el SEU, consulte la publicación *SEU User's Guide and Reference*.

1.4.2 La ayuda para el diseño de pantallas

El SDA es un programa de utilidad que facilita la creación de las pantallas que el programa necesita. Para obtener más información sobre la SDA, consulte la publicación *SDA User's Guide and Reference*.

1.4.3 El lenguaje de consulta estructurada

El sistema AS/400 le permite insertar sentencias de SQL/400 dentro de programas de RPG/400. Las sentencias SQL/400 se introducen en una especificación de cálculo. La sintaxis se muestra en la Figura 2. Debe observar las reglas siguientes:

- El delimitador de inicio **/EXEC SQL** debe introducirse en las columnas 7-15, con la barra inclinada en la columna 7.
- Las sentencias SQL/400 pueden comenzar en la misma línea que el delimitador de inicio.
- Las sentencias SQL/400 pueden continuarse en cualquier número de líneas de continuación posteriores. El delimitador de la línea de continuación es el **+** en la columna 7.
- Las sentencias SQL/400 no pueden sobrepasar la columna 74.
- El delimitador de fin **/END-EXEC** debe entrarse en las columnas 7-15, con la barra inclinada en la columna 7, en una línea separada. Este señala el final de las sentencias SQL/400. Se debe entrar solo, sin sentencias SQL/400 que lo sigan.

```
C
C      |
C      |
C/EXEC SQL      (el delimitador de inicio)
C+
C+      (líneas de continuación que contienen sentencias en SQL )
C+
.
.
.
C/END-EXEC      (el delimitador de fin)
C      |
C      |
C      |
```

Figura 2. Sintaxis para la entrada de sentencias de SQL/400 en un programa en RPG/400.

Debe entrar un mandato diferente para procesar las sentencias en SQL/400.

Consulte las publicaciones *SQL/400* Guía del Programador* y *Programación: Structured Query Language Reference* si desea información sobre cómo codificar las sentencias SQL/400.

Subtemas

1.4.3.1 Restricciones

1.4.3.1 Restricciones

En el lenguaje de programación RPG/400, las sentencias SQL/400 no pueden especificarse en el miembro fuente al que se hace referencia en una sentencia /COPY.

No debe utilizar sentencias SQL/400 en un programa generador de informes automático de RPG. En vez de eso, debe utilizar el mandato CRTRPTPGM para procesar los programas generadores de informes automáticos de RPG y salvar el fuente de RPG/400 que se haya generado. El informe automático generará el fuente de RPG/400, al que puede añadir sentencias de SQL/400. Para procesar las sentencias en SQL/400 y generar programa objeto de RPG, debe utilizar el preprocesador de SQL/400. Si las sentencias SQL/400 se preprocesan utilizando el preprocesador generador de informes automático de RPG/400, los resultados pueden no ser los esperados.

Consulte el manual *SEU User's Guide and Reference* para obtener información sobre cómo describe el SEU la comprobación de sintaxis del SQL/400 así como los manuales *SQL/400* Guía del Programador* y *Programming: Structured Query Language Reference* para obtener más información sobre el preprocesador de SQL/400.

1.5 Diseño de un programa en RPG/400

El diseño de un programa incluye:

- Decidir la salida de programa que necesita
- Decidir el proceso que producirá la salida que necesita
- Decidir la entrada que requiere y que está disponible para el programa.

Esta secuencia puede parecer invertida, ya que empieza por el resultado (la salida) y acaba por el principio (la entrada). Diseñar la salida primero es como saber dónde ir antes de empezar un viaje: le ayuda a decidir la mejor forma de llegar.

Subtemas

- 1.5.1 Diseño de la salida
- 1.5.2 Diseño del proceso
- 1.5.3 Diseño de la entrada

1.5.1 Diseño de la salida

El programa proporciona registros de salida. El usuario debe decidir qué hacer con estos registros. Generalmente tendrá tres opciones (o una combinación de las tres opciones):

- Puede visualizarlos.
- Puede imprimirlos.
- Puede almacenarlos.

Si desea visualizar los registros de salida en la estación de pantalla, debe decidir qué información desea visualizar y cómo desea que se disponga. Para definir cómo desea que se dispongan las pantallas, utilice el formulario para el diseño de pantallas. Entonces puede utilizar el programa de utilidad SDA para crear sus propias pantallas. Para obtener más información sobre la SDA, consulte la publicación *SDA User's Guide and Reference*.

Si desea imprimir los registros de salida en la pantalla, tiene que decidir la información que desea imprimir (qué campos y qué registros) y cómo desea que se disponga esta información en el informe impreso. Para indicar cómo quiere que se disponga el informe impreso, utilice el formulario para el diseño de impresoras.

Si desea conservar los registros de salida en almacenamiento, debe decidir qué información desea conservar y cómo desea organizar los campos en los registros de salida.

Después de diseñar todos los registros de salida, codifique los registros de las especificaciones de salida y de las especificaciones de descripción de archivo de RPG/400.

1.5.2 *Diseño del proceso*

El diseño del proceso significa la planificación y el cálculo que produce la salida necesaria. Al diseñar el proceso, debe fijarse en cómo trabaja el ciclo de programas de RPG/400. El ciclo de programas de RPG/400 controla ciertas operaciones de lectura y escritura que se efectúan en cada registro. Como resultado de ello, el ciclo de programas determina en parte cómo procesar los datos.

1.5.3 *Diseño de la entrada*

Después de decidir la salida que necesita y los cálculos que producirán la salida necesaria, el paso siguiente es determinar de dónde procederán los datos de entrada para el programa. Pueden proceder de uno o más archivos del sistema, de una o más estaciones de pantalla del sistema, de uno o más sistemas distintos o de una combinación de estas fuentes. Debe conocer los nombres utilizados para los archivos de entrada, la ubicación de los campos en los registros de entrada, la secuencia de los tipos de registro, los formatos de los datos numéricos y los indicadores utilizados. Después de conocer esta información, puede describir sus registros de entrada en las especificaciones de entrada del RPG/400.

1.6 Programación estructurada en el lenguaje de programación RPG/400

La programación estructurada es una forma de diseño y codificación que hace que los programas sean fáciles de entender, depurar y modificar.

Las tres estructuras utilizadas en cada programa del sistema son:

- Operación secuencial
- Bifurcación condicional
- Repetición de una operación basándose en cierta condición.

Idealmente, un programa estructurado es un jerarquía de módulos que pueden tener un único punto de entrada y un único punto de salida. El control se transfiere hacia abajo a través de la estructura sin necesidad de bifurcaciones incondicionales hacia los niveles más altos de la estructura.

A continuación se muestra cómo pueden utilizarse las tres estructuras en el lenguaje de programación RPG/400.

Subtemas

- 1.6.1 Operación secuencial
- 1.6.2 Bifurcación condicional
- 1.6.3 Repetición de una operación
- 1.6.4 Resumen de códigos de operación de programación estructurada

1.6.1 Operación secuencial

La operación secuencial significa cualquier serie de instrucciones que se procesan una después de otra, sin transferir el control a otra parte del programa.

1.6.2 Bifurcación condicional

Subtemas

1.6.2.1 Estructura If ELSE

1.6.2.2 Estructura SELEC

1.6.2.3 Otras estructuras de bifurcación condicional

1.6.2.1 Estructura If ELSE

Un ejemplo de una bifurcación condicional IF-THEN-ELSE, en lenguaje simple es:

```

IF (si) hace frío,

THEN (entonces) cogeré el abrigo;

ELSE, (si no) dejaré el abrigo en casa.
    
```

La Figura 3 es un gráfico de flujo de una bifurcación condicional.

IMAGEN 1

Figura 3. Gráfico de flujo de una bifurcación condicional

En el lenguaje de programación RPG/400, la estructura IF-THEN-ELSE se realiza mediante los códigos de operación **IFxx**, **ELSE**, y **END**. La Figura 4 muestra un diseño de una bifurcación condicional utilizando los códigos de operación **IFxx**, **ELSE** y **END**.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* En este ejemplo, si CENTR es igual a Y o si CENTR es igual a N,
C* se desactiva el indicador 52 moviendo '0' a *IN52. Si CENTR no es
C* igual a Y ni a N, se activa el indicador 52 moviendo '1' a *IN52.
C* La sentencia END finaliza el grupo IF/THEN/ELSE.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C          CENTR      IFEQ 'Y'
C          CENTR      OREQ 'N'
C          MOVE '0'          *IN52
C          ELSE
C          MOVE '1'          *IN52
C          END
    
```

Figura 4. Diseño de una bifurcación condicional utilizando las operaciones IF/ELSE/END

1.6.2.2 Estructura SELEC

Un ejemplo de una estructura de bifurcación condicional SELEC-WHEN-OTHER en lenguaje simple sería:

```
SELEC
WHEN (cuando) haga calor
  me pondré un sombrero
  iré a la playa
WHEN (cuando) haga frío
  me podré la chaqueta
OTHERwise (en caso contrario) no saldré a la calle
```

La Figura 5 es un diagrama de flujo de una bifurcación condicional SELEC-WHEN-OTHER.

IMAGEN 2

Figura 5. Diagrama de flujo de una bifurcación condicional SELEC-WHEN-OTHER

En el lenguaje de programación RPG/400, se utiliza la estructura SELEC-WHEN-OTHER mediante los códigos de operación SELEC, WHxx, y OTHER. La Figura 6 muestra la bifurcación condicional utilizando los códigos de operación SELEC, WHxx y OTHER.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgcomentarios+++
C*
C* Si X es igual a 1 entonces realizar las operaciones de la
C* secuencia 1; si X no es igual a 1, entonces si Y=2 y X<10 realizar
C* las operaciones de la secuencia 2. Si ninguna condición es cierta,
C* entonces realizar las operaciones de la secuencia 3.
C*
C          SELEC
C          X          WHEQ 1
C          :
C          :          sec 1
C          Y          WHEQ 2
C          X          ANDLT10
C          :
C          :          sec 2
C          OTHER
C          :
C          :          sec 3
C          ENDSL
C*
```

Figura 6. Bifurcación condicional utilizando las operaciones SELEC/WHxx/OTHER

1.6.2.3 Otras estructuras de bifurcación condicional

Hay otras tres formas de crear bifurcaciones condicionales:

- La operación **CASxx**
- La operación **GOTO** y los indicadores de condicionamiento
- La operación **CABxx**.

También puede crear una bifurcación a una subrutina con la operación **EXSR** y los indicadores de condicionamiento.

1.6.3 Repetición de una operación

El lenguaje de programación RPG/400 realiza tres estructuras de repetición-Hacer, Hacer mientras y y Hacer hasta-mediante los códigos de operación **DOWxx**, **DOUxx** y **DO** y el código de operación **END**.

Subtemas

1.6.3.1 Operación DO

1.6.3.2 Operación DOW (Hacer mientras)

1.6.3.3 Operación DOU (Hacer hasta)

1.6.3.1 Operación DO

La Figura 7 es un diagrama de flujo de una operación DO y la Figura 8 ilustra la codificación.

IMAGEN 3

Figura 7. Diagrama de flujo de una operación DO

Una operación DO funciona de la siguiente forma:

1. Establece el campo de índice (campo del resultado) en el valor inicial (factor 1).
2. Comprueba si el valor del campo de índice es mayor que el valor final (factor 2).

Si el valor del campo de índice es mayor que el valor final, se transfiere el control a la siguiente sentencia **END**.

3. Si el valor de campo de índice no es mayor que el valor final, se procesan las operaciones entre la sentencia **DO** y **END**.
4. En **END**, se incrementa el valor del campo de índice según el valor especificado en el factor 2 de la sentencia **END**, o en 1 si no se especifica ningún incremento.
5. El control se transfiere al paso 2 anterior.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* El siguiente ejemplo ilustra una operación DO. Como el factor
C* 1 de la sentencia DO está en blanco, el valor inicial de Y es 1, y
C* como el factor 2 de la sentencia END está en blanco, el valor de
C* incremento de Y es 1. El factor 2 de la sentencia DO contiene el
C* valor 10, que es el valor final de la rutina DO.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          Z-ADD1          X          20
C          DO 10          Y          30
C          X          LOKUPTABA          TABR          50
C 50          TABR          MULT 1.04          RATE          72
C          MOVE '1'          *IN90
C          EXCPT
C          MOVE '0'          *IN90
C          ADD 1          X
C          END
    
```

Figura 8. Diseño de la operación DO utilizando códigos de operación DO y END

1.6.3.2 Operación DOW (Hacer mientras)

Si comprueba la condición en primer lugar y a continuación procesa las operaciones, la estructura se llama DOW (Hacer mientras). Un ejemplo de una operación DOW (Hacer mientras) es:

1. Comparar una suma con 5.
2. Si la suma es menor que 5, añadir 1 a la suma.
3. Repetir los pasos 1 y 2 hasta que la suma sea igual o mayor que 5.

La Figura 9 es un diagrama de flujo de una operación DOW (Hacer mientras) y la Figura 10 ilustra la codificación de una operación DOW.

IMAGEN 4

Figura 9. Diagrama de flujo de un operación DOW (Hacer mientras)

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* El código siguiente determina si se ha llenado el subarchivo. Si
C* el indicador 32 está desactivado (*IN32 igual a 0), la operación
C* DOWEQ se procesa hasta que el resto del subarchivo se rellena con
C* registros en blanco.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          *IN32      DOWEQ'0'
C          MOVE *BLANKS  STATUS
C          MOVE *BLANKS  PRCDEX
C          MOVE *BLANKS  RSCDEX
C          Z-ADD0        EHWRKX
C          Z-ADD0        ACDATX
C          Z-ADD0        TFRRN
C          ADD 1         RECNO
C          WRITEMPFIL          32
C          END
C* El END precedente señala el final de la operación DOW (Hacer
mientras).
```

Figura 10. Diseño de una operación DOW (Hacer mientras) utilizando el código de operación DOWxx

Observe en la Figura 10 (Hacer mientras) que el programa comprueba en primer lugar si la condición es cierta. Si es así, se procesa el código existente entre las operaciones **DOW** y **END**. A continuación el programa retrocede para comprobar de nuevo si la condición es cierta aún y se repite todo el ciclo. Si la condición ya no es cierta, el control se transfiere a la instrucción que sigue inmediatamente a la operación **END**.

1.6.3.3 Operación DOU (Hacer hasta)

Si procesa las operaciones en primer lugar y a continuación comprueba la condición, la estructura se llama DOU (Hacer hasta). Un ejemplo de una operación DOU (Hacer hasta) es:

1. Añadir 1 a una suma.
2. Comparar una suma con 5.
3. Si la suma es menor que 5, repita los pasos 1 y 2 .

La Figura 11 es un diagrama de flujo de una operación DOU (Hacer hasta) y la Figura 12 muestra la codificación.

IMAGEN 5

Figura 11. Diagrama de flujo de un operación DOU (Hacer hasta)

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C*
C* Las líneas de código siguientes realizan una condición DO UNTIL.
C* El programa realiza un bucle entre la sentencia DOUEQ y la
C* sentencia END hasta que se alcance el final de archivo
C* (*IN50 igual a 1).
C          EMPSR      BEGSR
C*
C          *IN50      DOUEQ'1'
C          READ RCEMP                      50
C* Las líneas de código siguientes suman las horas del mes actual a
C* las horas del año a la fecha para el archivo maestro de empleados.
C* Dado que el factor 1 no está especificado en las sentencias, se
C* añade el factor 2 al campo de resultado. Si *INU4 está activado, se
C* ejecuta la sesión para el final del año y se mueven las horas del
C* año actual a las horas del año anterior.
C          ADD  EPHRC      EPHRY
C          ADD  EPNRC      EPNRY
C  U4      MOVE EPHRY      EPHRP
C  U4      MOVE EPNRY      EPNRP
C* El código siguiente borra los campos de las horas del mes actual
C* poniéndolas a cero y añadiéndoles un 0. Si se activa *INU4, se
C* ejecuta la sesión para el final del año y deben ponerse a cero las
C* horas del año actual.
C          Z-ADD0      EPHRC
C          Z-ADD0      EPNRC
C  U4      Z-ADD0      EPHRY
C  U4      Z-ADD0      EPNRY
C* El código siguiente actualiza el archivo maestro de empleados
C* utilizando el formato RCEMP.
C          UPDATRCEMP
C          END
C* La sentencia END precedente está asociada con la sentencia
C* DOUEQ.
```

Figura 12. Diseño de una operación DOU (Hacer hasta) utilizando el código de operación DOUwxx

1.6.4 Resumen de códigos de operación de programación estructurada

Los códigos de operación de programación estructurada son:

- IFxx** (Si/Entonces)
- ELSE** (Si no, Hacer)
- ENDyy** (Fin)
- DO** (Hacer)
- DOWxx** (Hacer mientras)
- DOUxx** (Hacer hasta)
- ANDxx** (Y)
- ORxx** (O)
- CASxx** (Subrutina de llamada condicional)
- SELEC** (Seleccionar un módulo)
- WHxx** (Cuando) **OTHER** (En caso contrario).
- OTHER** (En caso contrario).

en las cuales **xx** puede ser:

GT	El factor 1 es mayor que el factor 2.
LT	El factor 1 es menor que el factor 2.
EQ	El factor 1 es igual al factor 2.
NE	El factor 1 no es igual al factor 2.
GE	El factor 1 es igual o mayor que el factor 2.
LE	El factor 1 es igual o menor que el factor 2.
Espacios en blanco	El factor 1 no se compara con el factor 2 (Proceso incondicional). Esto sólo es válido para la operación CASxx .

y en las que **yy** puede ser:

CS	Fin de un grupo CAS.
DO	Fin de un grupo DO, DOU, o DOW.
IF	Fin de un grupo IF.
SL	Fin de un grupo SELEC.

1.7 Diseño de aplicaciones

El diseño de aplicaciones incluye determinar si el usuario debe crear un programa para que realice todas las funciones requeridas o debe crear múltiples programas para componer una aplicación.

Subtemas

- 1.7.1 Diseño de un programa único
- 1.7.2 Diseño de programa modular
- 1.7.3 Ejemplos de Diseño de aplicaciones

1.7.1 Diseño de un programa único

En un diseño de un programa único, todas las funciones se realizan dentro del programa. El diseño de un programa único es aplicable tanto a programas interactivos como de proceso por lotes. Se utiliza mejor cuando hay pocas funciones que son relativamente sencillas.

Por ejemplo, un programa de consulta interactiva que acepte un número de cliente de un operador, busque el registro correspondiente en un archivo maestro de clientes y visualice el registro como un programa simple podría tener un diseño de programa único.

IMAGEN 6

Un programa algo más complejo que pudiera tener también un diseño de programa único es un programa de mantenimiento de archivo que permita al operador:

- Hacer consultas dentro de un registro
- Cambiar un registro
- Suprimir un registro
- Añadir un registro

IMAGEN 7

Un ejemplo de un programa de proceso por lotes con un diseño de programa único es un programa que imprima una lista de los pedidos que cada operador haya entrado durante el día.

IMAGEN 8

1.7.2 Diseño de programa modular

El diseño de programa modular incluye la utilización de múltiples programas para realizar múltiples funciones, una por programa. El diseño de programa modular puede aplicarse a programas interactivos y a programas por lotes. Por ejemplo, la aplicación de entrada de pedidos mostrada en la Figura 13 está diseñada para que tenga cuatro programas:

- Un programa principal RPG/400 o CL
- Un programa RPG/400 que solicita el número de cliente y muestra la información del cliente en la pantalla
- Un programa RPG/400 que acepta la entrada de las líneas de detalle del pedido
- Un programa RPG/400 que calcula los totales del pedido.

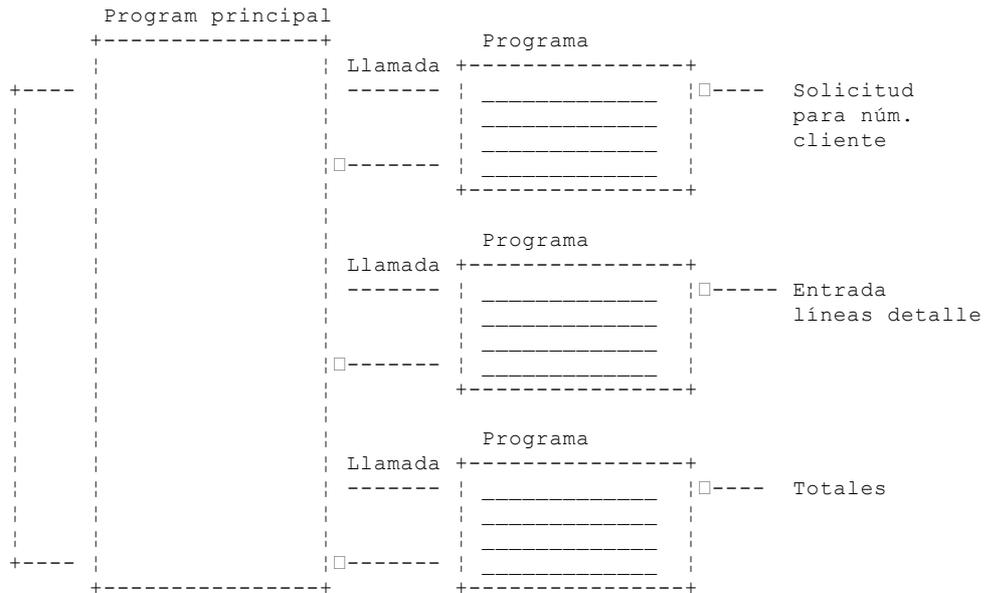


Figura 13. Diseño modular para una aplicación de entrada de pedidos

Un diseño de programa modular tiene varias ventajas potenciales:

- El diseño, la codificación, la comprobación y el mantenimiento de varios programas más pequeños puede ser más fácil que el diseño, la codificación, la comprobación y el mantenimiento de un programa grande y complejo. Con frecuencia esta elección es un asunto de preferencia personal pero muchas veces es beneficioso mantener los programas tan pequeños y simples como sea posible.
- Pueden solicitarse funciones CL desde programas RPG/400 dado que el AS/400 permite que los programas RPG/400 y CL se llamen entre sí.

Un único programa de larga duración puede tener secciones de codificación que se ejecuten de forma poco frecuente. Un diseño modular puede ordenarse para que el código que se utiliza infrecuentemente sólo se llame cuando se necesite.

Una posible desventaja del diseño modular de programas es la necesidad de llamar a programas adicionales. Estas llamadas tardan en codificarse y pueden necesitar una actividad general adicional del sistema para el proceso de los programas.

1.7.3 Ejemplos de Diseño de aplicaciones

A continuación se muestran las descripciones de programas modulares que ilustran algunos enfoques de diseño.

La función de entrada de pedidos mostrada en la Figura 14 tiene tres subfunciones:

- Aceptar la información de cabecera acerca de un pedido
- Aceptar la entrada de la línea de detalle del pedido
- Calcular los totales del pedido

Una forma de diseñar esta aplicación es hacer que un programa principal en CL llame a programas en RPG/400 para que realicen las funciones.



Figura 14. Ejemplo de diseño de aplicaciones para una función de entrada de pedidos

Cada uno de los programas de RPG/400:

- Abre los archivos
- Visualiza una solicitud para información y ayuda al usuario
- Acepta la entrada del usuario
- Procesa la información
- Cierra los archivos
- Retorna al programa principal

El resultado después de que el usuario introduce una pantalla es el siguiente:

1. Se procesa la entrada.
2. Se cierran los archivos.
3. Se devuelve el control al programa principal.
4. El programa principal llama al programa siguiente.
5. Este programa solicita la entrada del usuario.

Todo el proceso de las entradas y salidas de las estaciones de trabajo y todas las aperturas y cierres de los archivos se producen en los programas RPG/400. Por lo tanto, el usuario podría tener que esperar después de introducir una pantalla y antes de ver la siguiente.

En la Figura 15 se muestra una modificación del diseño previo que podría reducir los tiempos de respuesta y hacer un uso más efectivo de los recursos del sistema.



Guía del usuario RPG/400
Ejemplos de Diseño de aplicaciones

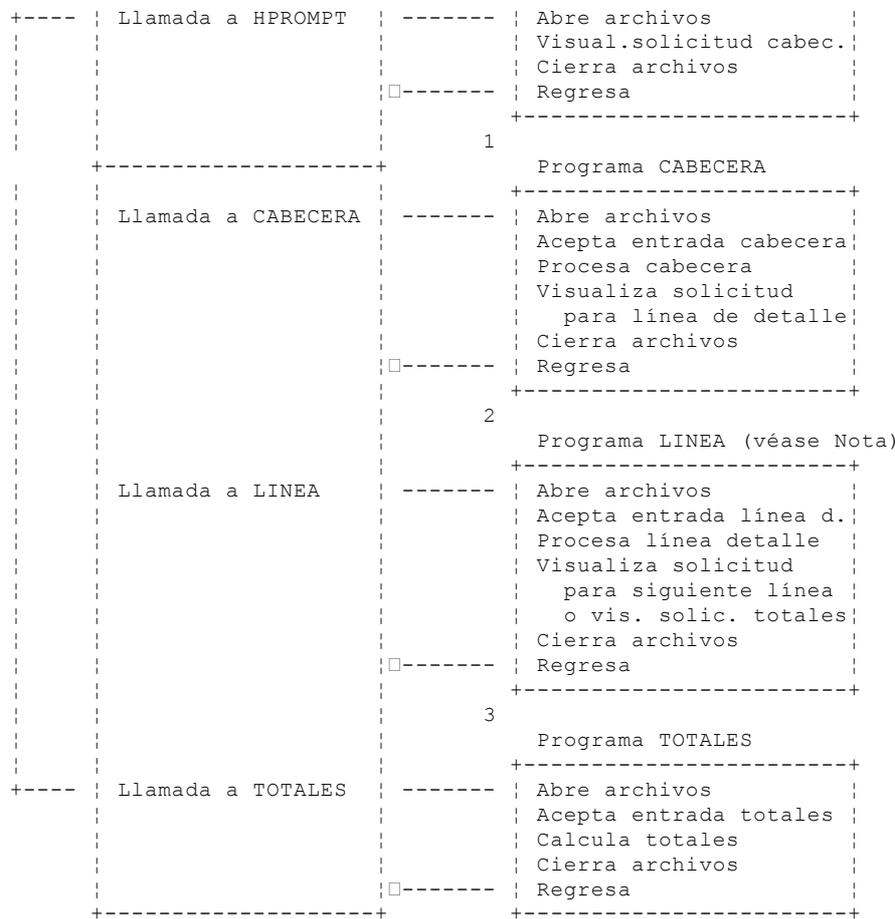


Figura 15. Ejemplo de diseño de aplicación modificado para una función de entrada de pedidos

Nota: En lugar de volver de forma incondicional al programa principal, puede diseñarse el programa **LINEA** para repetir en bucle dentro de sí mismo mientras se introducen las líneas de detalle.

Esta modificación permite que se produzca la entrada de los datos del usuario mientras se arrancan los programas y se abren y cierran los archivos. El solapamiento de la entrada de los datos y un proceso AS/400 se produce en los puntos 1, 2 y 3.

Para los dos ejemplos anteriores de diseño de programa modular, toda la entrada desde y la salida hacia estaciones de trabajo se produce en los programas. Para el ejemplo en la Figura 16, se producen una serie de operaciones en un programa principal de RPG/400.

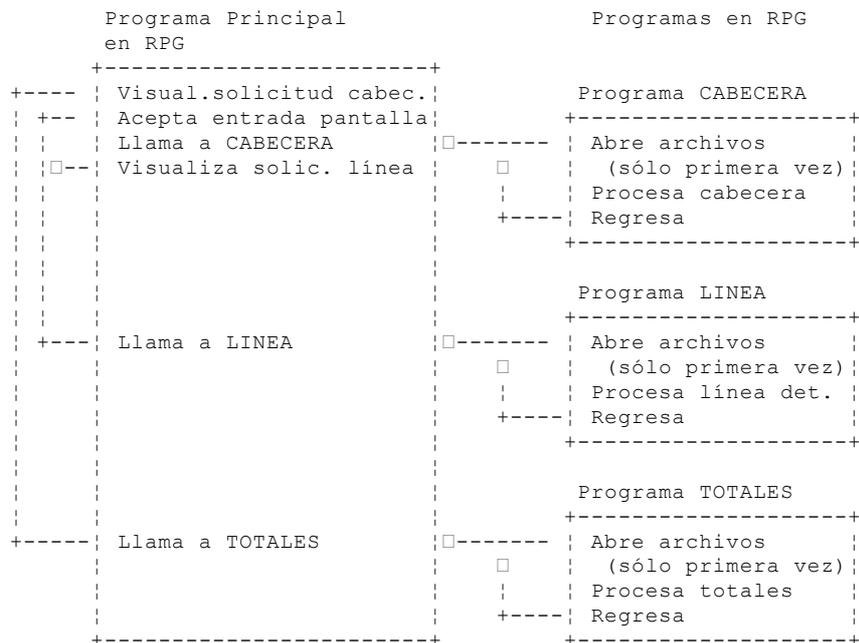


Figura 16. Ejemplo de diseño de aplicaciones con entrada y salida en programa principal

La entrada de la pantalla determina el programa que va a llamarse. Si se lee una cabecera, se llama a **CABECERA** y se transfiere como parámetro el registro de cabecera. Si se lee una línea de detalle, se llama a **LINEA** y se transfiere como parámetro un registro de línea de detalle. Si se lee la información de totales, se llama a **TOTALES** y se transfiere como parámetro un registro de totales.

Los programas dejan los archivos abiertos hasta que finaliza el trabajo, de este modo se elimina el tiempo de apertura y cierre de los archivos. Los programas no finalizan cuando regresan al programa principal.

2.0 Capítulo 2. Entrada de especificaciones en RPG/400

Después de diseñar el programa, deberá escribir las sentencias individuales que combinará en un programa fuente. Estas sentencias están codificadas en las hojas de especificaciones del RPG/400. Cada línea codificada en una hoja de especificaciones representa una sentencia en el programa fuente. Cada hoja de especificaciones contiene 80 columnas. Las cabeceras de las columnas indican la clase de información a codificar en las distintas columnas.

Este capítulo describe los tipos de especificaciones que puede entrar al crear un programa fuente en RPG/400. Este capítulo también describe cómo utilizar un editor de texto, un SEU por ejemplo, para entrar esta información directamente en el sistema y así empezar a crear el programa fuente en línea.

Subtemas

2.1 Las especificaciones en RPG/400

2.2 Entrada de un programa

2.1 Las especificaciones en RPG/400

Hay siete clases de especificaciones en RPG/400. Cuando el programa fuente se compila, estas especificaciones deben estar en la secuencia que sigue:

1. Especificaciones de control
2. Especificaciones de descripción de archivo
3. Especificaciones adicionales
4. Especificaciones del contador de líneas
5. Especificaciones de entrada
6. Especificaciones de cálculo
7. Especificaciones de salida

En este capítulo se describen brevemente cada una de estas especificaciones. La publicación *RPG/400 Reference* proporciona las descripciones detalladas de estas especificaciones.

Los programas en RPG/400 no tienen que utilizar todas las especificaciones. Un programa típico podría utilizar la descripción del archivo, la entrada, el cálculo, etc. y las especificaciones de salida.

Subtemas

- 2.1.1 La especificación de control
- 2.1.2 Especificaciones de descripción de archivo
- 2.1.3 Especificaciones adicionales
- 2.1.4 Especificaciones del contador de líneas
- 2.1.5 Especificaciones de entrada
- 2.1.6 Especificaciones de cálculo
- 2.1.7 Especificaciones de salida

2.1.1 La especificación de control

La especificación de control proporciona al compilador del RPG/400 información sobre el programa y el sistema. Esta incluye:

- El nombre del programa
- El formato de fecha del programa
- Si se utiliza un orden de clasificación alternativo o conversión de archivo.

Nota: La especificación de control es opcional.

2.1.2 Especificaciones de descripción de archivo

Las especificaciones de descripción de archivo describen todos los archivos que el programa utiliza. La información sobre cada archivo incluye:

- El nombre del archivo
- Cómo se utiliza el archivo
- El tamaño de los registros del archivo
- El dispositivo de entrada o salida utilizado para el archivo
- Si el archivo está condicionado por un indicador externo

2.1.3 Especificaciones adicionales

Las especificaciones adicionales describen todos los archivos de direcciones de registros, archivos de tablas y archivos de matriz utilizados en el programa. La información incluye:

- El nombre del archivo, matriz o tabla
- El número de entradas en un registro de entrada de matriz o tabla
- El número de entradas en una tabla o matriz
- La longitud de la entrada de la tabla o matriz.

2.1.4 Especificaciones del contador de líneas
Especificaciones del contador de líneas

Las especificaciones del contador de líneas describen la página o formulario en el que se imprimirá la salida. La información incluye:

- El número de líneas por página
- La línea de la página en la que ocurrirá el desbordamiento.

2.1.5 Especificaciones de entrada

Las especificaciones de entrada describen los registros, campos, estructuras de datos y constantes con nombre que utiliza el programa. Esta información de las especificaciones de entrada incluye:

- El nombre del archivo
- La secuencia de los tipos de registro
- Si se utilizan indicadores de identificación de registro, indicadores de nivel de control, indicadores de relación campo-registro o indicadores de campo
- Si se utilizan estructuras de datos, campos de consulta anticipada, códigos de identificación de registro o campos de comparación
- El tipo de cada campo (alfanumérico o numérico, decimal empaquetado, decimal con zona o en formato binario)
- La ubicación de cada campo en el registro
- El nombre de cada campo en el registro
- Las constantes con nombre.

2.1.6 Especificaciones de cálculo

Las especificaciones de cálculo describen los cálculos a realizar con los datos y el orden de los mismos. Las especificaciones de cálculo también pueden utilizarse para controlar operaciones de entrada y de salida. La información incluye:

- Indicadores de nivel de control y de condicionamiento para la operación especificada
- Los campos o constantes que se utilizarán en la operación
- La operación que se va a procesar
- Si se activan indicadores resultantes después de procesar la operación

2.1.7 Especificaciones de salida

Las especificaciones de salida describen los registros y campos en los archivos de salida y las condiciones bajo las que se procesan las operaciones de salida. La información incluye:

- El nombre del archivo
- El tipo de registro que se va a grabar
- Las instrucciones de espaciado y salto para archivos **PRINTER**
- Los indicadores de salida que condicionan cuándo tiene que grabarse el registro
- El nombre de cada campo en el registro de salida
- La ubicación de cada campo en el registro de salida
- Los códigos y las palabras de edición
- Las constantes que van a grabarse
- El nombre de formato para un archivo **WORKSTN**.

2.2 Entrada de un programa

Después de escribir el programa en RPG/400 en las hojas de especificaciones, debe entrarlo en los archivos fuente del sistema. Puede entrar el programa fuente de dos maneras:

- De forma interactiva, utilizando el SEU:

IMAGEN 9

La publicación *SEU User's Guide and Reference* proporciona una descripción completa sobre cómo entrar o actualizar un programa fuente en RPG/400 utilizando el SEU.

- En un proceso por lotes (es decir, desde disquete) utilizando la función de copia del sistema OS/400 o la función de spooling:

IMAGEN 10

La publicación *Guía para la Gestión de Datos* proporciona más información sobre cómo utilizar la función de copia o la de spooling para la entrada en proceso por lotes del programa fuente.

Nota: Cualquiera que sea el método de entrada del fuente que utilice, en literales, constantes, comentarios, datos de matriz y datos de tabla sólo podrá utilizar letras en minúscula. El resto de la información debe estar en letras mayúsculas.

3.0 Capítulo 3. Compilación de un programa en RPG/400

Existen dos entornos en los que puede compilar programas fuente: El entorno del sistema AS/400 y el del Entorno del Sistema/38. Consecuentemente hay dos formas de compilar programas fuente. Este capítulo describe:

- Utilización del mandato CL CRTRPGPGM para compilar un programa fuente en RPG/400 un entorno de sistema AS/400.
- Utilización de los mandatos CL CALL QCL y CRTRPGPGM para compilar un programa fuente en RPG/400 en el sistema Entorno del Sistema/38.

Este capítulo contiene también la información sobre cómo interpretar un listado del compilador.

Para compilar un programa, debe asegurarse de que la biblioteca QTEMP esté en la lista de bibliotecas. El mandato de CL CRTRPGPGM llama al compilador para que cree un objeto de programa en RPG/400 y un listado. (Si en el programa se utilizan archivos descritos externamente, el sistema OS/400 proporciona al programa información sobre los archivos durante la compilación.) La siguiente figura muestra una visión general del proceso de compilación:

IMAGEN 11

Figura 17. Visión general del proceso de compilación

El compilador comprueba la sintaxis del programa fuente en RPG/400 línea a línea y las interrelaciones entre líneas. Por ejemplo, comprueba que todos los nombres de campos estén definidos y, si un campo tiene múltiples definiciones, que cada definición tenga los mismos atributos.

El compilador del RPG/400 da soporte a una longitud de registro del archivo fuente de 102. Además de los campos normales de número de secuencia, (6 caracteres), última fecha de modificación (6 caracteres) y los datos (80 caracteres), se coloca al final del registro un campo de 10 caracteres que puede contener información adicional. El compilador del no utiliza esta información pero ésta se sitúa en el extremo derecho del listado del compilador. El usuario es responsable de colocar información en ese campo. Si desea utilizar el campo adicional ha de crear un archivo fuente con una longitud de registro de 102. El sistema AS/400 se envía con un archivo fuente en RPG/400 suministrado por IBM llamado QRPGRSRC, el cual tiene una longitud de registro de 92.

Subtemas

- 3.1 Mandato crear programa en RPG400 (CRTRPGPGM)
- 3.2 Mandato CRTRPGPGM
- 3.3 Compilación en el Entorno del Sistema/38

3.1 Mandato crear programa en RPG400 (CRTRPGPGM)

Para compilar un programa fuente en RPG/400 en un programa objeto debe introducir el mandato CL, CRTRPGPGM (Crear programa en RPG/400) para llamar al compilador de RPG/400. Los objetos de programa en RPG/400 se crean con la autorización pública *LIBCRTAUT. Puede que desee modificar esta autorización para mantener una mayor seguridad en el sistema.

Si el compilador del RPG/400 termina por causa de errores, se emite un mensaje de escape **QRG9001**. Un programa CL puede supervisar esta excepción utilizando el mandato CL, MONMSG (Supervisar mensaje). Consulte el Capítulo 4, "Mensajes de error, pruebas y depuración".

El compilador crea y actualiza un área de datos con el estado de la última compilación. A ésta área de datos se le llama RETURNCODE, tiene 400 caracteres de longitud y se sitúa en la biblioteca QTEMP. Puede tener acceso al área de datos RETURNCODE especificando RETURNCODE en el factor 2 de una sentencia ***NAMVAR DEFN**. El área de datos RETURNCODE tiene el formato siguiente:

Tabla 2. Contenido del área de datos RETURNCODE	
Byte	Contenido y significado
1	El carácter 1 significa que se ha creado el programa.
2	El carácter 1 significa que la compilación presenta anomalías debido a errores del compilador.
3	El carácter 1 significa que la compilación presenta anomalías debido a errores del fuente.
4	El carácter 1 significa compilado desde un fuente generado por el generador automático de informes.
5	El carácter 1 significa que no se ha llamado al supervisor de resolución de programas ya que se había seleccionado la opción *NOGEN en mandato CRTRPGPGM.
6-10	Numero de sentencias fuente.
11-12	Nivel de gravedad del mandato.
13-14	Gravedad más alta en el mensaje de diagnóstico.
15-20	Número de errores encontrados en el programa.
21-26	Fecha de compilación.
27-32	Hora de compilación.
33-100	No establecido.
101-110	Nombre del programa.
111-120	Nombre de la biblioteca del programa.
121-130	Nombre del archivo fuente.
131-140	Nombre de la biblioteca del archivo fuente
141-150	Nombre del miembro del archivo fuente.
151-160	Nombre del archivo del listado del compilador
161-170	Nombre de la biblioteca del listado del compilador
171-180	Nombre del miembro del listado del compilador.
181-190	Nombre del archivo fuente del generador automático de informes.
191-200	Nombre de la biblioteca del generador automático de informes.

Mandato crear programa en RPG400 (CRTRPGPGM)

201-210	Nombre del miembro del generador automático de informes.
211-370	No establecido.
371-378	Tamaño de la representación intermedia del programa transferido al supervisor de resolución de programas.
379	No establecido.
380-384	Tiempo total de compilación.
385	No establecido.
386-390	Tiempo utilizado por el compilador.
391-395	Tiempo utilizado por el supervisor de resolución de programas.
396-400	Tiempo usado por el convertidor.

Todos los nombres de objetos especificados en el mandato CRTRPGPGM deben estar compuestos por caracteres alfanuméricos, el primero de los cuales debe ser alfabético. Se permiten todos los convenios de denominación del sistema OS/400. La longitud de los nombres no puede exceder de 10 caracteres. Consulte la publicación *CL Guía del Programador* para obtener una descripción detallada de las reglas de denominación de objetos en OS/400 así como una descripción completa de la sintaxis de mandatos del OS/400.

No es probable que se excedan los límites de tamaño internos del sistema para un programa. Sin embargo, si se exceden los límites, el programa debe volverse a escribir, normalmente en forma de múltiples programas.

Subtemas

3.1.1 Utilización del mandato CRTRPGPGM

3.1.1 Utilización del mandato CRTRPGPGM

Puede llamar al compilador RPG/400 de una de las tres formas siguientes:

- De forma interactiva desde la pantalla de visualización de mandatos CRTRPGPGM utilizando mensajes de solicitud. El usuario solicita la pantalla, ilustrada en la Figura 19 en el tema 3.2 y en la Figura 20 en el tema 3.2, tecleando el mandato CL, CRTRPGPGM y pulsando F4 a continuación.
- Entrando CRTRPGPGM seguido solamente de los parámetros por palabra clave que alteran temporalmente las definiciones por omisión. Esta sentencia se entra de forma interactiva en la línea de mandatos o puede hacerse como parte de una corriente de entrada por lotes.
- Entrando CRTRPGPGM seguido sólo por los valores del parámetro, en la secuencia adecuada. Este método es el utilizado más frecuentemente cuando va a someter las peticiones de compilación como parte de un programa CL. Este método puede utilizarse también de forma interactiva pero el usuario está limitado por el CL a entrar sólo los tres primeros valores del parámetro.

Nota: Puede cambiarse cualquier valor por omisión del mandato CRTRPGPGM o cualquier otro mandato CL utilizando el mandato CL, CHGCMDDFT (Cambiar Valor por Omisión Mandato). Consulte la publicación *CL Reference* para obtener más información.

Subtemas

- 3.1.1.1 Elementos de las líneas de mandatos de CRTRPGPGM
- 3.1.1.2 Entrada de elementos desde la pantalla de mandatos de CRTRPGPGM
- 3.1.1.3 Entrada de ciertos parámetros solamente
- 3.1.1.4 Entrada de valores de parámetros solamente

3.1.1.1 Elementos de las líneas de mandatos de CRTRPGPGM

Las descripciones siguientes hacen referencia a los tres elementos de la línea de mandatos del compilador:

- La palabra del mandato del compilador CL, CRTRPGPGM
- El parámetro, al cual se refiere mediante una palabra clave como PGM, SRCFILE, GENOPT, etc.
- El valor del parámetro. Este puede ser un valor predefinido o un nombre de objeto.

Todos los nombres de objeto deben estar formados por caracteres alfanuméricos. El primer carácter tiene que ser alfabético siempre y la longitud del nombre no puede exceder de diez caracteres. Es posible utilizar todo el convenio de denominación del sistema OS/400.

3.1.1.2 Entrada de elementos desde la pantalla de mandatos de CRTRPGPGM

Teclee **CRTRPGPGM** y pulse F4 a continuación. Esto hace que aparezcan las pantallas de solicitud de CRTRPGPGM. Pulse F10 para obtener parámetros adicionales. Estas pantallas y los valores que puede entrar en ellas aparecen descritas más adelante en este capítulo.

Cada parámetro de la pantalla visualiza un valor por omisión. Mueva el cursor a continuación de los elementos en donde quiere que se aplique el valor por omisión. Teclee sobre los elementos en los que quiera definir un valor u opción diferentes. Si no está seguro sobre cómo definir un parámetro en particular, teclee un interrogante "?" como primer carácter de este campo y pulse Intro para recibir una información más detallada. El interrogante debe ir seguido por un espacio en blanco.

Cuando haya establecido todos los valores satisfactoriamente pulse Intro.

3.1.1.3 *Entrada de ciertos parámetros solamente*

Todos los parámetros del CRTRPGPGM tienen valores por omisión. Teclee simplemente **CRTRPGPGM** seguido solamente por esos parámetros (especificados por una palabra clave) cuyas definiciones por omisión desea alterar temporalmente. Separe los parámetros con espacios; entre valores para cada parámetro encerrando el valor o los valores entre paréntesis.

Por ejemplo, para cambiar el nombre del programa y de la biblioteca y aceptar los valores por omisión para los otros parámetros, entre:

CRTRPGPGM PGM(nueva biblioteca/ nuevo nombre)

3.1.1.4 Entrada de valores de parámetros solamente

Tiene la posibilidad de elegir el introducir valores de parámetros solamente, sin especificar las palabras clave del parámetro. En este caso, dado que no hay palabra clave que indique al sistema el valor que pertenece a cada parámetro, debe entrar todos los valores en la secuencia mostrada a continuación. No necesita entrar todo el conjunto de opciones, pero debe entrar las opciones para todos los parámetros hasta llegar hasta el que le interesa. El sistema utilizará los valores por omisión para los parámetros restantes.

Por ejemplo, para compilar un programa fuente en el archivo QRPGSRC en el miembro ABC de la biblioteca SRCLIB entre:

```
CRTRPGGM QTEMP/ABC SRCLIB/QRPGSRC *PGM
```

Tenga en cuenta que también debe entrar los nombres del programa y de la biblioteca del programa compilado. El sistema reconoce la opción que pertenece a cada parámetro por la posición del valor en la línea de mandatos del compilador. Puede entrar de forma posicional un máximo de tres valores de parámetro.

Para obtener más información sobre los mandatos del sistema AS/400 consulte la publicación *CL Reference*.

3.2 Mandato CRTRPGPGM

Se muestra todo el diagrama de sintaxis para el mandato CRTRPGPGM en Figura 18 en el tema 3.2.

Lea el diagrama de sintaxis de izquierda a derecha, de arriba a abajo, siguiendo la ruta de la línea.

Los mandatos de lenguaje de control (CL), parámetros y palabras clave se pueden entrar con caracteres en mayúsculas o en minúsculas. En este manual se muestran en mayúsculas (por ejemplo, PARAMETRO, VALOR-PREDEFINIDO). Las variables aparecen en letras minúsculas en cursiva (por ejemplo, *valor-definido-usuario*). Las variables son nombres o valores definidos por el usuario.

Para obtener información sobre cómo leer diagramas de sintaxis, consulte "Cómo interpretar diagramas de sintaxis" en el tema PORTADA_2.2.

```

-----
Job: B,I Pgm: B,I REXX

>>--CRTRPGPGM-----
      |               +-*CURLIB/-----+ +-*CTLSPEC-----+ |
      +-PGM--(--+-----+-----+-----+-----+)-+
      |               +-nombre-biblioteca/+ +-nombre-programa/+
>-----
      |               +-*LIBL/-----+ +-QRPGSRC-----+ |
      +-SRCFILE--(--+-----+-----+-----+-----+)-+
      |               +-*CURLIB/-----+ +-nombre-archivo-fuente+
      |               +-nombre-biblioteca/+
                                     (P)
>-----
      |               +-*PGM-----+ |               +-*OPTION--(--| Detalles OPTION +--)-+
      +-SRCMBR--(--+nombre-miembro-archivo-fuente---)-+
>-----
      |               +-*NONE-----+ |               +-*NONE-----+ |
      +-GENOPT--(--| Detalles GENOPT +--)-+ |               +-INDENT--(--+valor-carácter---)-+
>-----
      |               +-*NONE-----+ | |               +-*NOFLAG--+ |
      +-CVTOPT--(--+| Detalles CVTOPT +---)-+ +-SAAFLAG--(--+*FLAG-----)-+
>-----
      |               +-9-----+ | |               +-*LIBL/-----+ +-QSYSVRT-----+
      +-GENLVL--(--+valor-nivel-gravedad---)-+ +-PRTFILE--(--+-----+-----+-----+)-+
      |               +-*CURLIB/-----+ | +-nombre-archivo+
      |               +-nombre-biblioteca/+
>-----
      |               +-*YES--+ | |               +-*CURRENT-----+ | |               +-*USER--+ |
      +-REPLACE--(--+*NO-----)-+ | +-TGTRLS--(--+*PRV-----+---)-+ | +-USRPRF--(--+*OWNER-----)-+
      |               +-nivel-release+
>-----
      |               +-*LIBCRTAUT-----+ | |               +-*SRCMBRTXT-----+ |
      +-AUT--(--+*CHANGE-----+---)-+ | +-TEXT--(--+*BLANK-----+---)-+
      |               +-*USE-----+ | |               +- 'descripción' +
      |               +-*ALL-----+ |
      |               +-*EXCLUDE-----+ |
      |               +-nombre-lista-autorizaciones+
>-----
      |               +-*NO--+ | |               +-*NONE-----+ |
      +-PHSTRC--(--+*YES-----)-+ | |               | <-----+ (1) | |
      |               +-ITDUMP--(--+nombre-fase-----)-+
>-----
      |               +-*NONE-----+ | |               +-*NONE-----+ |
      |               | <-----+ (1) | |               +-CODELIST--(--+*ALL-----+---)-+
      +-SNPDUMP--(--+nombre-fase-----)-+ | |               | <-----+ (1) |
      |               +-nombre-fase-----+
>-----
      |               +-*NO--+ | |               +-*NO--+ |
      +-IGNDECERR--(--+*YES-----)-+ | +-ALWNULL--(--+*YES-----)-+
  
```

Notas:

- (1) Un máximo de 25 repeticiones
- (P) Todos los parámetros anteriores a este punto pueden especificarse por posición

Detalles OPTION:

```

+--*SRC-----+
+--*SOURCE---| +--*XREF---+ +--*GEN---+ +--*NODUMP-+ +--*NOSECLVL-+ +--*NOSRCDBG-+ +--*NOLSTDBG-+
+-----+-----+-----+-----+-----+-----+-----+-----+
+--*NOSRC---| +--*NOXREF-+ +--*NOGEN-+ +--*DUMP---+ +--*SECLVL---+ +--*SRCDBG---+ +--*LSTDBG---+
+--*NOSOURCE-+
  
```

Detalles GENOPT:

```

+--*NOLIST-+ +--*NOXREF-+ +--*NOATR-+ +--*NODUMP-+ +--*NOPATCH-+ +--*NOOPTIMIZE-+
+-----+-----+-----+-----+-----+-----+
+--*LIST---+ +--*XREF---+ +--*ATR---+ +--*DUMP---+ +--*PATCH---+ +--*OPTIMIZE---+
  
```

Detalles CVTOPT:

```

+-----+-----+-----+
+--*DATETIME-+ +--*VARCHAR-+ +--*GRAPHIC-+
  
```

Figura 18. Sintaxis del Mandato CRTRPGPGM

A continuación se muestran unos ejemplos de pantallas de solicitud para el mandato CRTRPGPGM. Las pantallas de ejemplo se proporcionan en conjuntos. La primera pantalla del conjunto describe los valores que puede entrar, la segunda pantalla presenta las palabras clave y los valores por omisión. Puede conmutar entre las pantallas de valores y de palabras clave pulsando F11. El texto que sigue a las pantallas describe esas palabras clave y los valores por omisión.

En la descripción de los parámetros, y luego se resaltan. Los parámetros se presentan en secuencia. Siga esta secuencia si solamente está entrando los valores de los parámetros sin la correspondiente abreviatura de parámetro.

Nota: Consulte en el Apéndice E, "Opción del entorno del Sistema/38 del compilador RPG" una descripción de las diferencias entre compilaciones de programas en RPG/400 y programas RPG III de Entorno del Sistema/38.

```

-----
                          Crear Programa RPG/400 (CRTRPGPGM)

Teclee las opciones, pulse Intro.

Programa . . . . . *CTLSPEC__ Nombre, *CTLSPEC
Biblioteca . . . . . *CURLIB___ Nombre, *CURLIB
Archivo fuente . . . . . QRPGRSRC___ Nombre, QRPGRSRC
Biblioteca . . . . . *LIBL_____ Nombre, *LIBL, *CURLIB
Miembro fuente . . . . . *PGM_____ Nombre, *PGM
Nivel gravedad generación . . . 9_____ 0-99
Texto 'descripción'. . . . . *SRCMBRTXT_____

-----

                          Parámetros Adicionales

Opciones de listado fuente . . . _____ *SOURCE, *NOSOURCE, *SRC...
+ para más valores _____
Opciones de generación . . . . . _____ *LIST, *NOLIST, *XREF...
+ para más valores _____
Sangrado de listado fuente . . . *NONE Valor del carácter, *NONE
                                          Más...
F3=Salir F4=Solicitud F5=Renovar F12=Cancelar F13=Cómo usar esta pantalla
F24=Más teclas
  
```

```

-----

                          Crear Programa RPG/400 (CRTRPGPGM)

Teclee las opciones, pulse Intro.

Programa . . . . . PGM *CTLSPEC_
  
```

```

Biblioteca . . . . . *CURLIB__
Archivo fuente . . . . . SRCFILE QRPGSRC__
Biblioteca . . . . . *LIBL____
Miembro fuente . . . . . SRCMBR *PGM_____
Nivel gravedad generación . . . GENLVL 9_____
Texto 'descripción'. . . . . TEXT *SRCMBRTX_____
-----
                                Parámetros Adicionales
-----
Opciones de listado fuente . . . OPTION _____
                                + para más valores _____
Opciones de generación . . . . . GENOPT _____
                                + para más valores _____
Sangrado de listado fuente . . . INDENT *NONE
-----
F3=Salir F4=Solicitud F5=Renovar F12=Cancelar F13=Cómo usar esta pantalla Más...
F24=Más teclas
-----

```

Figura 19. Primer Conjunto de Pantallas de Solicitud de CRTRPGPGM

PGM

Especifica el nombre de biblioteca y de programa por el que se conocerá el programa compilado en RPG/400. Si no se especifica ninguna biblioteca, el programa creado se almacena en la biblioteca actual.

***CTLSPEC**

Se utiliza el nombre del programa especificado entre las posiciones 75 y 80 de la especificación de control.

Si el nombre del programa no está especificado en la especificación de control, pero el programa fuente es de un archivo de base de datos, se utiliza el nombre del miembro, especificado por el parámetro SRCMBR como el nombre del programa. Si el fuente no es del archivo de base de datos, el programa toma el nombre por omisión RPGOBJ.

nombre-programa

Entre el nombre con el que se conocerá el programa.

***CURLIB**

El programa compilado se almacena en la biblioteca actual. Si no ha especificado ninguna biblioteca actual, se utiliza QGPL.

nombre-biblioteca

Entre el nombre de la biblioteca donde ha de almacenarse el programa.

SRCFILE

Especifica el nombre del archivo fuente que contiene el programa fuente en RPG/400 que se ha de compilar, así como la biblioteca en la que se encuentra el archivo fuente.

QRPGSRC

El archivo fuente por omisión QRPGSRC contiene el programa fuente en RPG/400 que se ha de compilar.

nombre-archivo-fuente

Entre el nombre del archivo fuente que contiene el programa fuente en RPG/400 que se ha de compilar.

***LIBL**

El sistema busca en la lista de bibliotecas para encontrar la biblioteca en la que se encuentra el archivo fuente.

***CURLIB**

Se utiliza la biblioteca actual para buscar el archivo fuente. Si no ha especificado ninguna biblioteca actual, se utiliza QGPL.

nombre-biblioteca

Entre el nombre de la biblioteca donde se almacena el programa.

SRCMBR

Especifica el nombre del miembro del archivo fuente que contiene el programa fuente en RPG/400 que se ha de compilar. Este parámetro sólo puede especificarse si el archivo fuente denominado en el parámetro SRCFILE es un archivo de base de datos.

***PGM**

Utiliza el nombre especificado por el parámetro *PGM como nombre del miembro del archivo fuente. El programa compilado tendrá el mismo nombre que el miembro del archivo fuente. Si no se especifica ningún nombre de programa en el parámetro *PGM, el mandato utiliza el primer miembro creado en o añadido al archivo fuente como el nombre del miembro fuente.

nombre-miembro-archivo-fuente

Entre el nombre del miembro que contiene el programa fuente en RPG/400.

GENLVL

Especifica si se genera o no un objeto del programa dependiendo de la gravedad de los errores encontrados. Con este parámetro puede especificarse un valor del nivel seguridad correspondiente al nivel de seguridad de los mensajes emitidos durante la compilación. Si se producen errores en un programa con un nivel de seguridad inferior a 30, y si se especifica un nivel de seguridad mayor que el del programa para este parámetro, el programa se compila; sin embargo, es posible que el programa contenga errores que causen resultados imprevistos al ejecutar el programa. Para errores de programa iguales o mayores que gravedad 30, la compilación del programa puede que finalice o puede que el objeto de programa no se genere, con independencia del valor de este parámetro. No se recomienda especificar un valor mayor que 30 para este parámetro.

9

No se generará un objeto del programa si tiene mensajes de error con un nivel de gravedad mayor o igual a 9.

valor-nivel-seguridad:

Entre un número, del 0 al 99.

Nota: El valor del nivel de seguridad de los mensajes compilados en RPG/400 no sobrepasa el 50.

TEXT

Permite al usuario entrar texto que describa brevemente el programa y sus funciones. El texto aparece cuando se visualiza la información del programa.

***SRCMBRTXT**

Se utiliza el texto del miembro fuente.

***BLANK**

No aparece ningún texto.

'descripción'

Entre el texto que describa brevemente el programa y su función. El texto puede tener 50 caracteres como máximo y debe ir entre apóstrofes. Los apóstrofes no forman parte de la cadena de 50 caracteres. No se necesitan apóstrofes si entra el texto en la pantalla de solicitud.

OPTION

Especifica las opciones que se utilizarán al compilar el programa fuente. Puede especificar algunas o todas las opciones en el orden

que desee. Separe las opciones por un espacio en blanco.

***SOURCE**

Genera un listado fuente, formado por la entrada del programa en RPG/400 y todos los errores de hora de compilación.

***NOSOURCE**

No se genera ningún listado fuente. Si se especifica ***NOSOURCE**, el sistema supone que el usuario tampoco desea un listado de referencias cruzadas y especifica ***NOXREF** también.

La abreviatura aceptada para ***SOURCE** es ***SRC** y para ***NOSOURCE** es ***NOSRC**.

***XREF**

Genera un listado de referencias cruzadas y una tabla de información para el campo de clave (cuando sea adecuado) para el programa fuente.

Nota: Si también desea especificar ***NOSOURCE** o ***NOSRC**, debe especificar explícitamente ***XREF** o se supone ***NOXREF**.

***NOXREF**

No se genera ningún listado de referencias cruzadas.

Nota: Si también se especifica ***NOSOURCE** o ***NOSRC**, se altera temporalmente el valor por omisión normal (***XREF**) y ***NOXREF** es el valor por omisión.

***GEN**

Crea un objeto de programa que puede ejecutarse después de compilar el programa.

***NOGEN**

No crea un objeto de programa.

***NODUMP**

No vuelca áreas de datos importantes cuando se produce un error durante la compilación.

***DUMP**

Vuelca las áreas de datos importantes cuando se produce un error durante la compilación.

***NOSECLVL**

No imprime el texto de mensajes de segundo nivel en la línea siguiente al texto de los mensajes de primer nivel.

***SECLVL**

Imprime texto de mensajes de segundo nivel.

***NOSRCDBG**

No genera información sobre errores de nivel de fuente y depuración.

***SRCDBG**

Genera información sobre errores de nivel de fuente y depuración. Presenta un archivo de sucesos aunque el compilador se complete sin ningún error.

***NOLSTDBG**

No genera información sobre errores y depuración.

***LSTDBG**

Genera la visualización de un listado y información sobre errores y depuración necesaria para la visualización del listado.

Nota: Sólo puede utilizar las opciones ***NOSRCDBG**, ***SRCDBG**, ***NOLSTDBG** y ***LSTDBG** si utiliza el producto AD/Cycle CoOperative Development Environment/400 para compilar los programas. Si especifica una o más de estas opciones pero

no tiene instalado el producto AD/Cycle CODE/400, el compilador del RPG/400 no continuará el proceso y se emitirá un mensaje de error. Para obtener más información sobre estas opciones, consulte la publicación *CODE Debug Tool User's Guide and Reference*, SC09-1622.

GENOPT

Especifica las opciones que se utilizarán para crear el objeto del programa: la impresión de la representación intermedia de un programa (IRP), un listado de referencias cruzadas de los objetos definidos en el IRP, un listado de los atributos del IRP y la plantilla del programa. También puede especificar opciones en el parámetro GENOPT para reservar un área de parches de un programa y para mejorar un programa para que se ejecute con más eficacia. Estos resultados puede ser útiles si se produce un problema al intentar ejecutar el programa compilado. Puede especificar algunas o todas las opciones en el orden que desee. Separe los valores mediante un espacio en blanco. Para observar una descripción del parámetro GENOPT y de la información que proporciona, consulte el "Opciones de depuración del compilador" en el tema A.2 en Apéndice A, "Información de servicio del programa generador automático de informes y del compilador RPG".

***NOOPTIMIZE**

No procesa la optimización del programa.

***OPTIMIZE**

Procesa la optimización del programa. Con ***OPTIMIZE**, el compilador genera un programa proporcionándole un proceso más eficaz que posiblemente necesitará un almacenamiento menor. Si especifica ***OPTIMIZE** podrá aumentar considerablemente el tiempo que se necesita para crear un programa. Los objetos de programa de salida pueden optimizarse con el mandato CL CHGPGM.

INDENT

Especifica si se generará o no el listado del programa fuente de RPG/400 con el sangrado de las operaciones estructuradas para mejorar su lectura. También especifica los caracteres que se utilizarán para marcar las cláusulas de operaciones estructuradas.

***NONE**

El compilador generará un listado sin sangrado.

valor-carácter

El listado fuente se sangra para las cláusulas de operaciones estructuradas. La alineación de las sentencias y de las cláusulas se marca utilizando los caracteres de su elección. Puede elegir cualquier serie de caracteres de 2 caracteres de longitud como máximo. Si desea utilizar un blanco en su serie de caracteres, debe teclearlo entre apóstrofes.

Nota: El sangrado puede no aparecer como esperaba si hay errores en el programa RPG/400.

El segundo conjunto de pantallas de solicitud que aparece en Figura 20 proporciona más valores y palabras clave que puede entrar en el mandato CRTRPGPGM.

Crear Programa RPG/400 (CRTRPGPGM)		
Teclee las opciones, pulse Intro.		
Teclee opciones de conversión.	*NONE _____	*NONE, *DATETIME, *VARCHAR...
+ para más valores		
Señalización SAA.	*NOFLAG _____	*NOFLAG, *FLAG
Imprimir archivo	QSYSPT _____	Nombre
Biblioteca	*LIBL _____	Nombre, *LIBL, *CURLIB
Sustituir programa	*YES _____	*YES, *NO
Release destino.	*CURRENT _____	*CURRENT, *PRV, V2R1M0...
Perfil de usuario.	*USER _____	*USER, *OWNER
Autorización	*LIBCRTAUT _____	Nombre, *LIBCRTAUT, *ALL...
Seguimiento de fases	*NO _____	*NO, *YES

```

Vuelco de texto intermedio . . . *NONE_____
-----
Vuelco breve . . . . . *NONE_____
-----
Listado de códigos . . . . . *NONE_____
-----
                                                                 Más...
F3=Salir  F4=Solicitud  F5=Renovar  F12=Cancelar  F13=Cómo usar esta pantalla
F24=Más teclas
    
```

```

                                Crear Programa RPG/400 (CRTRPGPGM)

Teclee las opciones, pulse Intro.

Teclee opciones de conversión. . CVTOPT          *NONE_____
                                + para más valores
Señalización SAA. . . . . *NOFLAG          *NOFLAG, *FLAG
Imprimir archivo . . . . . QSYSPT_____ Nombre
                                *LIBL_____
Biblioteca . . . . .
Sustituir programa . . . . . REPLACE          *YES
Release destino. . . . . TGTRLS          *CURRENT
Perfil de usuario. . . . . USRPRF          *USER_
Autorización . . . . . AUT          *LIBCRTAUT
Seguimiento de fases . . . . . PHSTRC          *NO_
Vuelco de texto intermedio . . *NONE_____
-----
Vuelco breve . . . . . *NONE_____
-----
Listado de códigos . . . . . *NONE_____
-----
                                                                 Más...
F3=Salir  F4=Solicitud  F5=Renovar  F12=Cancelar  F13=Cómo usar esta pantalla
F24=Más teclas
    
```

Figura 20. Segundo conjunto de pantallas de solicitud de CRTRPGPGM

CVTOPT

Especifica cómo el compilador RPG/400 maneja los tipos de datos de la base de datos de la fecha, hora e indicación de la hora y los tipos de datos de longitud variable que se recuperan de los archivos descritos externamente. Consulte "Tipos de datos SAA" en el tema 10.3 para obtener una descripción detallada de cómo soporta el compilador del RPG/400 los tipos de datos SAA.

***NONE**

Se ignoran los tipos de datos de la base de datos de la fecha, hora, indicación de la hora y de longitud variable y no se pueden acceder en el programa en RPG/400. Se emite un mensaje informativo en tiempo de compilación, de gravedad 0, cuando un formato de registro contiene campos ignorados.

***DATETIME**

Especifica que los tipos de datos de la base de datos de la fecha, hora e indicación de la hora se tienen que declarar como campos de caracteres de longitud fija y que se pueden acceder en el programa en RPG/400.

***VARCHAR**

Especifica que los tipos de datos de la base de datos de longitud variable se tienen que declarar como campos de caracteres de longitud fija y se pueden acceder en el programa en RPG/400.

***GRAPHIC**

Especifica que los tipos de datos gráficos del juego de caracteres de doble byte (DBCS) se tienen que declarar como campos de caracteres de longitud fija y se pueden acceder en el programa en RPG/400.

Nota: Elija los dos parámetros, *VARCHAR y *GRAPHIC, si desea que los tipos de datos gráficos DBCS de longitud variable se declaren en el programa.

SA AFLAG

Especifica si se producirá la señalización de las especificaciones que no están soportadas por el SAA RPG. Para más información sobre las señalizaciones en SAA y sobre cómo y cuándo utilizarlas, consulte "Mensajes de señalización de Systems Application Architecture" en el tema 4.1.2.

***NOFLAG**

No se producirá la señalización.

***FLAG**

Se producirá la señalización. Se listarán mensajes bajo la cabecera Resumen de mensajes de SAA. No se emitirán mensajes para una especificación si se emite un mensaje de gravedad 30 o superior para esa especificación.

PRTFILE

Especifica el nombre del archivo en el que ha de colocarse el listado del compilador y la biblioteca en la que se encuentra este archivo. Si especifica un archivo cuya longitud de registro sea menor de 132, se perderá la información.

QSYSPRT

Si no especifica el nombre de un archivo, el listado del compilador se sitúa en el archivo suministrado por IBM, QSYSPRT. El archivo QSYSPRT tiene una longitud de registro de 132.

nombre-archivo

Introduzca el nombre del archivo en el que ha de situarse el listado del compilador.

***LIBL**

El sistema busca en la lista de bibliotecas para encontrar la biblioteca en la que se encuentra el archivo.

***CURLIB**

Se utiliza la biblioteca actual para buscar el archivo. Si no ha especificado una biblioteca actual, se utilizará QGPL.

nombre-biblioteca

Introduzca el nombre de la biblioteca en donde se encuentra el archivo.

REPLACE

Especifica si un nuevo objeto de programa objeto se creará si ya existe un programa con el mismo nombre en la biblioteca especificada.

***YES**

Se creará un nuevo objeto de programa y cualquier objeto de programa que exista con el mismo nombre en la biblioteca especificada se trasladará a la biblioteca QRPLOBJ.

***NO**

No se creará ningún nuevo objeto de programa si un objeto de programa existe con el mismo nombre en la biblioteca especificada.

TGTRLS

Especifica el nivel de release del sistema operativo en el cual se intenta utilizar el objeto que se está creando.

Puede especificar un nivel de release exacto con el formato VxRxMx, donde Vx es la versión, Rx es el release y Mx es el nivel de modificación.

Nota: Para utilizar el objeto en el sistema destino, tiene que salvar el objeto en el nivel de release destino especificado en el mandato de creación y después restaurarlo en el sistema

destino.

***CURRENT**

El objeto se va a utilizar en el release del sistema operativo que se se ejecuta actualmente en el sistema.

***PRV**

El objeto se va a utilizar en el release anterior con nivel de modificación 0 del sistema operativo.

nivel-release

Especifica el release con el formato VxRxMx. El objeto puede utilizarse en un sistema con el release especificado o con cualquier release posterior al sistema operativo instalado.

Los valores válidos dependen de la versión actual, release y nivel de modificación y cambian con cada nuevo release.

USRPRF

Especifica bajo qué perfil de usuario ha de ejecutarse el programa compilado en RPG/400. Este perfil controla los objetos que puede utilizar el programa (incluyendo la autorización que tiene el programa para cada objeto). Si el programa ya existe, el parámetro USRPF no se actualizará al nuevo perfil.

***USER**

El programa se ejecuta bajo el perfil de usuario del usuario del programa.

***OWNER**

El programa se ejecuta bajo el perfil de usuario del propietario y el usuario del programa. Se utilizan los conjuntos de autorizaciones de objetos colectivas de los dos perfiles para buscar y acceder a objetos mientras se ejecuta el programa. Los objetos que se creen durante el programa serán propiedad del usuario del programa.

Nota: El parámetro USRPRF refleja las necesidades de seguridad del sistema. Los recursos de seguridad disponibles para el sistema AS/400 aparecen descritos en detalle en las publicaciones *Seguridad Manual de Consulta* y *CL Reference*.

AUT

Especifica la autorización que se está otorgando a los usuarios que no tienen una autorización específica para el objeto, que no están en la lista de autorizaciones y cuyo grupo de usuarios no tiene autorización específica para el objeto. Puede alterarse la autorización para todos los usuarios o para usuarios específicos después de la creación del programa con los mandatos CL, Otorgar autorización sobre objeto (GRTOBJAUT) o Revocar autorización sobre objeto (RVKOBJAUT). Para obtener más información sobre estos mandatos, consulte la publicación *CL Reference*.

***LIBCRTAUT**

La autorización de uso público sobre un objeto se toma de la palabra clave CRTAUT de la librería destino (la librería que contiene el objeto). El valor se determina cuando se crea el objeto. Si el valor CRTAUT de la biblioteca cambia después de su creación, el nuevo valor no afectará a los objetos existentes.

***CHANGE**

El público tiene autorización operativa sobre objetos y todos las autorizaciones sobre datos del programa compilado. Cualquier usuario puede ejecutar, depurar, cambiar y realizar funciones básicas en el programa.

***USE**

El público puede ejecutar el programa compilado pero no puede depurarlo ni modificarlo.

***ALL**

El público tiene autorización total sobre el programa.

***EXCLUDE**

El público no puede utilizar el programa.

nombre de lista-autorización

Nombre de una lista de autorización a la que está añadida el programa. Para obtener una descripción de la lista de autorización y cómo crearla consulte la publicación *CL Reference*.

Nota: Utilice el parámetro AUT para reflejar los requisitos de seguridad del sistema. Los recursos de seguridad disponibles se describen con detalle en el manual *Seguridad Manual de Consulta*.

PHSTRC

Especifica si se proporciona información en el listado sobre las fases del compilador. Consulte el Apéndice A, "Información de servicio del programa generador automático de informes y del compilador RPG" para una explicación detallada de este parámetro.

***NO**

No proporciona información acerca de las fases del compilador.

***YES**

Proporciona información acerca de las fases del compilador.

ITDUMP

Este parámetro especifica si ha de imprimirse durante la compilación un listado dinámico de texto intermedio de una o más de las fases especificadas cada vez que se crea un registro IT. Este parámetro especifica también si ha de imprimirse un flujo de rutinas principales que se ejecutan en una o más de las fases especificadas. Consulte el Apéndice A, "Información de servicio del programa generador automático de informes y del compilador RPG" para obtener una explicación detallada de este parámetro.

***NONE**

No se produce ningún vuelco de texto intermedio.

nombre-fase

Entre los dos últimos caracteres del nombre de la fase. Consulte el Apéndice A, "Información de servicio del programa generador automático de informes y del compilador RPG" para obtener una explicación detallada de este parámetro.

SNPDUMP

Especifica si se imprimirán las áreas de datos principales después de ejecutar una o varias fases especificadas. Consulte el Apéndice A, "Información de servicio del programa generador automático de informes y del compilador RPG" para obtener una explicación detallada de este parámetro.

***NONE**

No se produce vuelco breve.

nombre-fase

Entre los dos últimos caracteres del nombre de la fase. Consulte el Apéndice A, "Información de servicio del programa generador automático de informes y del compilador RPG" para obtener una explicación detallada de este parámetro.

CODELIST

Especifica si ha de imprimirse un listado dinámico del IRP durante la compilación de una o más fases especificadas del programa fuente. Consulte el Apéndice A, "Información de servicio del programa generador automático de informes y del compilador RPG" para obtener una explicación detallada de este parámetro.

***NONE**

No se produce un listado de la codificación para cada una de las fases de generación de código que se ejecutan.

***ALL**

Produce un listado de la codificación para cada una de las fases de generación de código que se ejecutan.

nombre-fase

Entre los dos últimos caracteres del nombre de la fase. Consulte el Apéndice A, "Información de servicio del programa generador automático de informes y del compilador RPG" para obtener una explicación detallada de este parámetro.

El tercer conjunto de pantallas de solicitud que aparece en Figura 21 proporciona más valores y palabras clave que el usuario puede entrar para el mandato CRTRPGPGM.

```
-----+-----
                          Crear Programa RPG/400 (CRTRPGPGM)
-----+-----

Teclee las opciones, pulse Intro.

Ignorar errores datos decimales . .   *NO_           *NO, *YES
Permitir valores nulos . . . . .   *NO_           *NO, *YES

                                          Final
F3=Salir  F4=Solicitud  F5=Renovar  F12=Cancelar  F13=Cómo usar esta pantalla
F24=Más teclas
-----+-----
```

```
-----+-----
                          Crear Programa RPG/400 (CRTRPGPGM)
-----+-----

Teclee las opciones, pulse Intro.

Ignorar errores datos decimales . .   IGNDDECERR      *NO_
Permitir valores nulos. . . . .   ALWNULL         *NO_

                                          Final
F3=Salir  F4=Solicitud  F5=Renovar  F12=Cancelar  F13=Cómo usar esta pantalla
F24=Más teclas
-----+-----
```

Figura 21. Tercer Conjunto de Pantallas de Solicitud CRTRPGPGM

IGNDECERR

Especifica si el programa ha de ignorar los errores de datos decimales detectados por el sistema.

***NO**

No ignora los errores de datos decimales. Cuando se intenta una operación en un campo numérico que contiene datos decimales que no son válidos, se produce una excepción de programa. Los errores de datos decimales sólo se detectarán en los campos definidos en formato decimal empaquetado. Para obtener más información sobre el formato decimal empaquetado, consulte el Capítulo 11, "Comunicación con objetos en el sistema" en el tema 11.0.

***YES**

Ignora los errores de datos decimales. El efecto de los errores de datos decimales sobre el proceso no puede predecirse fácilmente. El compilador sólo genera un mensaje de error en el listado del compilador para comunicar al usuario que esta opción está especificada. Los resultados incorrectos que se producen durante la ejecución del programa, cuando se ha especificado esta opción, son responsabilidad del usuario.

ALWNULL

Especifica si un programa en RPG/400 aceptará valores nulos de campos que permitan valores nulos en un archivo de solo entrada descrito externamente. Se emite un mensaje en tiempo de compilación, de gravedad 0 cuando un formato de registro contiene campos que permiten valores nulos. Consulte "Soporte de valores nulos" en el tema 10.3.4 para obtener una descripción detallada sobre cómo el compilador del RPG/400 da soporte a campo con posibilidad de nulos.

***NO**

Especifica que el programa en RPG/400 no procesará los campos de valores nulos de archivos descritos externamente. Si intenta recuperar un registro que contenga valores nulos, el programa en RPG/400 no puede acceder a ningún dato del registro y se produce un error de correlación de datos.

***YES**

Especifica que un programa en RPG/400 aceptará campos de valores nulos para un archivo sólo de entrada descrito externamente. Cuando se recupera un registro que contiene valores nulos de un archivo de sólo entrada descrito externamente, no se produce ningún error de correlación de datos y los valores por omisión de la base de datos se colocan en los campos que contienen valores nulos.

3.3 Compilación en el Entorno del Sistema/38

También puede compilar un programa fuente en RPG/400 desde el Entorno del Sistema/38. Deberá llamar al compilador con los mismos mandatos que si utilizara el entorno nativo AS/400 (CRTRPGPGM para llamar al compilador de RPG/400 y CRTRPTPGM para llamar a la función de generación de informe automático). Para compilar un programa desde el Entorno del Sistema/38, utilice el mandato CL, CALL QCL para llamar al Entorno del Sistema/38, antes de introducir el mandato CRTRPGPGM. También puede entrar los mandatos de Entorno del Sistema/38, desde el entorno nativo mediante mandatos de calificación de bibliotecas. El mandato QSYS38/CRTRPGPGM llama al compilador RPG III del Entorno del Sistema/38.

Para obtener más información sobre las diferencias entre el programa RPG/400 en el entorno AS/400 y en el Entorno del Sistema/38, consulte el Apéndice E, "Opción del entorno del Sistema/38 del compilador RPG".

Para obtener más información sobre cómo programar en el Entorno del Sistema/38, consulte la publicación *System/38 RPG III Reference Manual and Programmer's Guide*.

4.0 Capítulo 4. Mensajes de error, pruebas y depuración

Este capítulo describe los mensajes de error que puede recibir desde el compilador del RPG/400, explicando el significado de los mismos así como la forma de visualizarlo e imprimirlos. Este capítulo describe también cómo probar y depurar un programa en RPG/400 utilizando las funciones proporcionadas por el compilador del RPG/400 y el sistema OS/400.

Sistema OS/400	RPG
<input type="checkbox"/> Biblioteca de pruebas	<input type="checkbox"/> código de operación DEBUG
<input type="checkbox"/> Puntos de interrupción	<input type="checkbox"/> código de operación DUMP
<input type="checkbox"/> Rastreo	

Las funciones del sistema OS/400 le permiten utilizar mandatos de CL para probar programas, protegiendo al mismo tiempo los archivos de producción y permitiéndole operaciones de observación y depuración mientras se ejecuta un programa. Consulte la publicación *CL Reference* para obtener más información sobre cómo utilizar los mandatos de CL.

No es necesario un código fuente especial para utilizar las funciones del sistema OS/400. Las funciones del compilador del RPG/400 pueden utilizarse independientemente de las funciones del sistema OS/400 o en combinación con ellas para:

- Depurar un programa
- Producir un vuelco con formato de los valores de los indicadores y del contenido de los campos, estructuras de datos, matrices y tablas.

Se necesita codificación fuente especial para utilizar los códigos de operación **DEBUG** y **DUMP** del RPG/400. También puede obtener un vuelco con formato como respuesta a un mensaje en tiempo de ejecución.

La estructura de datos de la información del archivo y la estructura de datos del estado del programa pueden proporcionar información adicional sobre la depuración. Estas estructuras de datos se describen más adelante en este capítulo. A continuación presentamos la información acerca del tratamiento de excepciones/errores.

Subtemas

- 4.1 Utilización, visualización e impresión de mensajes
- 4.2 Cómo ejecutar un programa en RPG/400
- 4.3 Utilización de una biblioteca de pruebas
- 4.4 Utilización de puntos de interrupción
- 4.5 Utilización de un rastreo
- 4.6 Utilización de códigos de operación **DEBUG**
- 4.7 Utilización del vuelco con formato en RPG/400
- 4.8 Manejo de excepciones/errores

4.1 Utilización, visualización e impresión de mensajes

Subtemas

4.1.1 Utilización de mensajes

4.1.2 Mensajes de señalización de Systems Application Architecture

4.1.3 Visualización e impresión de mensajes

4.1.1 Utilización de mensajes

Este manual hace referencia a los mensajes que se reciben durante la compilación y en tiempo de ejecución. Estos mensajes se visualizan en la pantalla o se imprimen en la lista del compilador. Este producto no contiene ningún manual de mensajes.

Cada mensaje está formado por tres partes como mínimo, tal como se muestra en el siguiente mensaje de ejemplo:

```
+-----+
|
|  A  10
|
|  B  Mensaje: La sintaxis de la entrada de Identificación
|     del programa no es válida. Se toma por omisión RPGOBJ.
|
|  C  Causa: La entrada de Identificación del programa
|     (posiciones 75-80) de una especificación de control tiene una sintaxis no
|     válida: el primer carácter no es alfabético o no está justificado por
|     la izquierda, o contiene blancos intercalados o caracteres especiales.
|     Se toma por omisión RPGOBJ.
|
|     Recuperación: Especifique RPGOBJ o una entrada
|     válida (posiciones 75-80) para la opción de Identificación del programa.
|     Vuelva a compilar.
|
+-----+
```

A Un número que indica la gravedad del error. El valor del nivel de seguridad de los mensajes en tiempo de compilación del RPG/400 no sobrepasa el 50.

Gravedad Significado

- | | |
|----|--|
| 00 | Un mensaje informativo visualizado durante la entrada, compilación y ejecución. No se ha detectado ningún error y no es necesaria ninguna acción correctiva. |
| 10 | Un mensaje de aviso visualizado durante la entrada, compilación y ejecución. Este nivel indica que se ha detectado un error pero por su escasa importancia no interferirá con la ejecución del programa. Se asume que los resultados de la operación serán correctos. |
| 20 | Un mensaje de error visualizado durante la compilación. Este nivel indica que se ha cometido un error, pero que el compilador está intentando una recuperación que podría producir el código deseado. Puede que el programa no funcione como esperaba el autor. |
| 30 | Un mensaje de error grave visualizado durante la compilación. Este nivel indica que se ha detectado un error demasiado grave para la recuperación automática. La compilación ha terminado, pero el programa no se ejecuta. |
| 40 | Un mensaje de fin de programa o de función de forma anormal visualizado durante la compilación o la ejecución. Este nivel indica un error que obliga a cancelar el proceso. La operación se ha terminado porque era incapaz de manejar datos válidos, o porque el usuario la ha cancelado. |
| 50 | Un mensaje de fin de trabajo anormal visualizado durante la compilación o la ejecución. Este nivel indica un error que obliga a cancelar el trabajo. El trabajo ha terminado porque una función no se ha realizado como era necesario o porque el usuario lo ha cancelado. |
| 99 | Una acción que el usuario debe realizar durante la ejecución. Este nivel indica que es necesaria alguna intervención manual por parte del operador, tal como la introducción de una respuesta, cambiar de disquete, o cambiar el papel de la impresora. |

B El texto que ve en línea o en un listado. Este texto es una breve descripción del problema, generalmente una frase.

C Este texto se imprime en el listado si especifica ***SECLVL** en las opciones en tiempo de compilación. Contiene una descripción ampliada del mensaje y una sección que indica la respuesta correcta del usuario. El valor por omisión que IBM suministra para esta opción es ***NOSECLVL**.

En el tiempo de ejecución, puede entrar **D** para obtener un vuelco con formato RPG/400, **S** para obtener un vuelco del sistema, **C** para cancelar, **G** para continuar el proceso en ***GETIN**, o **F** para obtener un vuelco con formato completo RPG/400.

4.1.2 Mensajes de señalización de Systems Application Architecture

Además de los mensajes descritos anteriormente, el compilador del RPG/400 tiene también un conjunto de mensajes que señalan aquellas características del compilador del RPG/400 a las que el SAA RPG no da soporte. Estos mensajes se solicitan mediante una opción del compilador, **SAAF** descrita en la "Mandato CRTRPGPGM" en el tema 3.2 El valor por omisión para esta opción es ***NOFLAG**. Si selecciona ***FLAG**, encontrará impresos estos mensajes de forma independiente bajo el encabezamiento Resumen de mensajes SAA.

Los mensajes de señalización SAA están pensados para ayudar al programador cuando escribe un código portátil. Si busca un máximo de portabilidad, debe eliminar del programa los códigos señalados. Un programa que tenga solamente mensajes SAA se compilará y ejecutará correctamente en el sistema AS/400. Los mensajes SAA son sólo informativos. Los mensajes de error graves pueden causar la supresión de los mensajes SAA.

Los mensajes SAA se dividen de la misma forma que los demás mensajes descritos aquí. Un mensaje de ejemplo es:

A 0

B **Mensaje:** SAA RPG no da soporte a campos numéricos de más de 15 dígitos.

C **Causa:** El RPG de la interfaz de programación común de Systems Application Architecture no da soporte a campos numéricos de más de 15 dígitos.

Recuperación: Si se requiere la adhesión a SAA en RPG, cambie el programa y vuelva a compilar.

Estos mensajes sólo realizan la señalización en funciones específicas del compilador del RPG/400.

4.1.3 Visualización e impresión de mensajes

Para visualizar o imprimir mensaje concretos, utilice los mandatos DSPMSGF o DSPMSGD. Los mensajes en tiempo de compilación se almacenan en un archivo llamado QRPMSG en la biblioteca QRP. Los mensajes en tiempo de ejecución se almacenan en un archivo llamado QRPMSGGE en la biblioteca QSYS.

En el entorno del Sistema/38, todos los mensajes en tiempo de compilación se encuentran en el archivo QRP3MSG de la biblioteca QRP38. Los mensajes en tiempo de ejecución se encuentran en el archivo QRP3MSGGE de la biblioteca QSYS.

Nota: Si tiene algún comentario o sugerencia relacionada con los mensajes, utilice la Hoja de comentarios del lector incluida en este manual y envíenoslos.

4.2 Cómo ejecutar un programa en RPG/400

Hay muchas formas de ejecutar un programa en RPG/400 dependiendo de cómo se haya escrito el programa y de quién lo esté utilizando. Consulte en el manual *CL Guía del Programador* distintas formas de ejecutar un programa en RPG/400. Las tres formas más comunes de ejecutar un programa en RPG/400 son mediante:

- Una sentencia u operación CALL de un lenguaje de alto nivel
- Un menú orientado a la aplicación
- Un mandato creado por el usuario.

La sentencia CL CALL puede formar parte de un trabajo por lotes, el usuario puede entrarla interactivamente desde una estación de trabajo, o puede estar incluida en un programa CL. Un ejemplo es CALL NOMINA. NOMINA es el nombre de un programa CL o de un programa en RPG/400 que se llama y a continuación se ejecuta. Un programa en RPG/400 puede llamar a otro programa mediante el código de operación CALL. Consulte Capítulo 11, "Comunicación con objetos en el sistema".

Otra forma de ejecutar un programa en RPG/400 es mediante un menú orientado a la aplicación. Puede solicitar un menú orientado a la aplicación y a continuación seleccionar un opción que llama al programa adecuado. La Figura 22 es un ejemplo de un menú orientado a la aplicación:

```
+-----+
|                                     |
|          MENU NOMINA DEL DEPARTAMENTO          |
|                                     |
|    1. Consultar original de empleados          |
|    2. Cambiar original de empleados          |
|    3. Añadir nuevos empleados                |
|    4. Regresar                               |
|                                     |
| Opción:__                                   |
|                                     |
+-----+
```

Figura 22. Ejemplo de un menú orientado a la aplicación

Este menú suele escribirse como un programa en CL en el que cada opción llama a un programa en RPG/400 independiente. Cuando termina un programa en RPG/400, el sistema devuelve el control al programa que lo ha llamado o al usuario. Podría tratarse de un usuario de estación de trabajo, un programa en CL (como por ejemplo el programa de gestión de menús), u otro programa en RPG/400.

También puede crear un mandato para que ejecute un programa en RPG/400 utilizando un definición de mandatos. Consulte la publicación *CL Guía del Programador* para obtener una descripción de cómo definir un mandato. Por ejemplo, puede crear un mandato, PAGA, que llame al programa NOMINA. Un mandato creado por el usuario puede entrarse en un trabajo por lotes, o bien interactivamente, por un usuario de una estación de trabajo.

Subtemas

4.2.1 Soporte Salvar-mientras-activo

4.2.1 Soporte Salvar-mientras-activo

Los programas de aplicación que modifican objetos o datos pueden ejecutarse mientras se salvan los objetos o datos. Consulte el manual *Advanced Backup and Recovery Guide* para obtener una serie de consideraciones de programación relacionadas con el soporte salvar-mientras-activo.

4.3 Utilización de una biblioteca de pruebas

El concepto básico para probar y depurar es el de crear un entorno de pruebas independiente. Los programas que se ejecutan en un entorno operativo normal o en un entorno de pruebas pueden leer, actualizar y grabar registro que se encuentren en bibliotecas de pruebas o de producción. Para evitar que archivos de base de datos de bibliotecas de producción se modifiquen de forma no intencionada, puede especificar la opción **UPDPROD(*NO)** en mandato de CL STRDBG (Empezar depuración).

En el sistema AS/400 puede copiar archivos de producción en las bibliotecas de pruebas o puede crear archivos especiales para las pruebas en esta biblioteca. La copia de prueba de un archivo y la copia de producción pueden tener el mismo nombre si los archivos se encuentran en bibliotecas distintas. Puede utilizar el mismo nombre de archivo de programa para procesos de pruebas y normales.

La Figura 23 muestra un ejemplo de cómo utilizar un entorno de pruebas independiente.

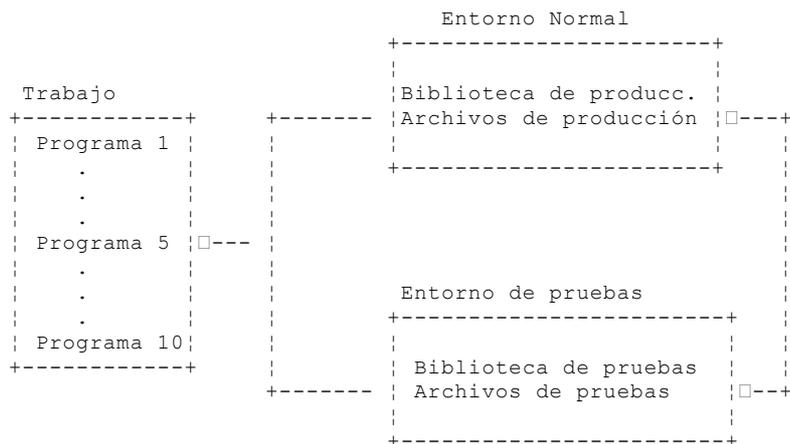


Figura 23. Utilización de un entorno de pruebas independiente

Para realizar pruebas, debe colocar el nombre de la biblioteca de pruebas antes del nombre de la biblioteca de producción en la lista de bibliotecas del trabajo que contiene el programa que se va a probar tal como aparece en la Figura 24.

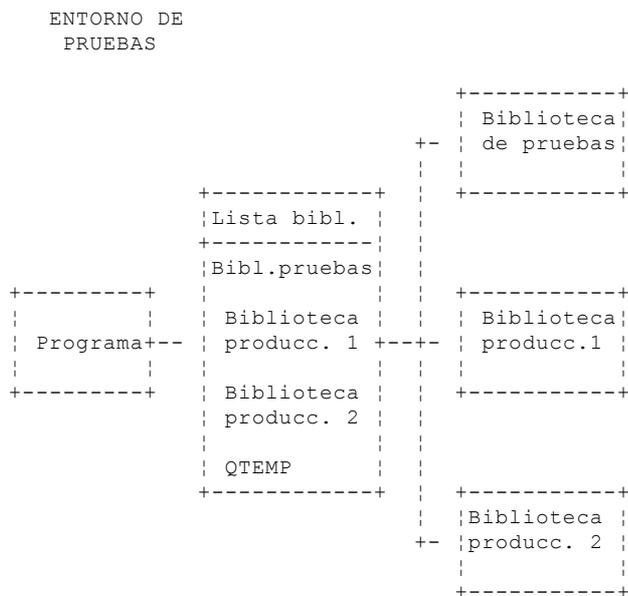


Figura 24. Entorno de pruebas

En la ejecución normal del programa, la biblioteca de producción debe ser la única biblioteca incluida en la lista de bibliotecas para este trabajo. (O sea, no debe incluirse la biblioteca de pruebas). Consulte la Figura 25 a continuación.

NORMAL

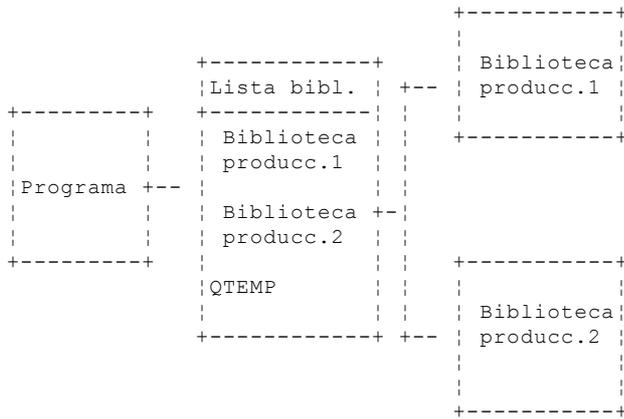


Figura 25. Entorno normal

El programa que se prueba no contiene ninguna sentencia especial que deba probarse. El mismo programa que se prueba puede ejecutarse de forma normal sin necesidad de modificarlo. Todas las funciones de prueba están especificadas en el trabajo que contiene al programa y no en el programa.

TRABAJO

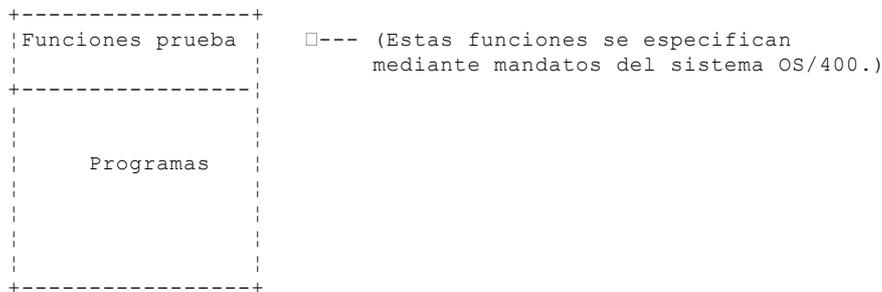


Figura 26. Funciones de prueba

Las funciones de prueba sólo se aplican al trabajo en el que se han especificado. Un programa puede utilizarse a la vez en dos trabajos: en un trabajo que esté en un entorno de prueba y en otro trabajo que esté en un entorno de proceso normal.

Las funciones de prueba del sistema OS/400 le permiten interactuar con un programa mientras éste se ejecuta de forma que pueda observar el proceso. Estas funciones incluyen la utilización de puntos de interrupción y rastreos.

4.4 Utilización de puntos de interrupción

Puede utilizar los puntos de interrupción para detener el programa en un punto de interrupción determinado. Este punto de interrupción puede ser un número de sentencia o una etiqueta del programa. Si utiliza una etiqueta como punto de interrupción en lugar de un número de sentencia, la etiqueta puede estar:

- Sobre una sentencia TAG del programa
- Asociada a un paso del ciclo del programa en RPG/400. Por ejemplo, ***TOTC** indica el comienzo de los cálculos de totales y ***TOTL** indica el comienzo de la salida de totales.
- Asociada a una función realizada por el usuario en un programa en RPG/400. Por ejemplo, **SQRT** indica la función de extracción de la raíz cuadrada.

Cuando se encuentra un punto de interrupción para un trabajo interactivo, el sistema visualiza el punto de interrupción en el que se detiene el programa y, si se solicitan, se visualizan los valores de las variables del programa. Después de obtener esta información (en una pantalla) puede ir a la Pantalla de Entrada de Mandatos y luego introducir mandatos CL para solicitar otras funciones (como por ejemplo visualizar o cambiar una variable, añadir un punto de interrupción o añadir un rastreo).

Cuando encuentre un punto de interrupción en un archivo por lotes, puede llamar a un programa de punto de interrupción. Debe crear este programa de punto de interrupción si desea que gestione la información del punto de interrupción.

Subtemas

- 4.4.1 Ejemplo de utilización de puntos de interrupción
- 4.4.2 Consideraciones al utilizar puntos de interrupción

4.4.1 Ejemplo de utilización de puntos de interrupción

La Figura 27 muestra un listado fuente de un programa de ejemplo en RPG/400, DBGPGM, así como de los mandatos CL que añaden puntos de interrupción en las sentencias 1200 y 1500. El valor de la variable *IN se visualiza cuando se alcanza el punto de interrupción de la sentencia 1200, y el valor de las variables FLD1 y PART se visualiza cuando se alcanza el punto de interrupción de la sentencia 1500.

Mandatos de CL

```
STRDBG  PGM(EJEMPLOS/DBGPGM)
ADDBKP  STMT(1200) PGMVAR((*IN))
ADDBKP  STMT(1500) PGMVAR((CAM1) (PART)) OUTFMT(*HEX)
```

```
5763RG1 V3R0M5 940125          IBM RPG/400
QGPL/DBGPGM          01/25/94 13:42:19      Página  2
NUMERO DE
SECUENCIA *...1...+...2...+...3...+...4...+...5...+...6...+...7...*  IND.  DO  ULTIMA
                                         UTIL. NUM. ACTUAL.
                                         L i s t a d o   d e l   F u e n t e
H
100 FTESTX  IF  F      5          DISK          01/01/94
200 FTESTA  UF  F     10          DISK          01/01/94
300 ITESTX  NS  01          01/01/94
400 I              1   5 PART          01/01/94
500 ITESTA  NS  02          01/01/94
600 I              1   5 FLD1         01/01/94
700 *****
800 *          RUTINA PRINCIPAL
900 *****
1000 C          LOOP          TAG          01/01/94
1100 C          READ TESTX          66          3          01/01/94
1200 C   66          GOTO ENDPGM          01/01/94
1300 C          READ TESTA          67          3          01/01/94
1400 C  N67          MOVE PART          FLD1          01/01/94
1500 C  N67          EXCPTMAST          01/01/94
1600 C  N66          GOTO LOOP          01/01/94
1700 C          ENDPGM          TAG          01/01/94
1800 C          SETON          LR          1          01/01/94
1900 OTESTA  E          MAST          01/01/94
2000 O          FLD1          5          01/01/94
* * * * *   F I N   D E L   F U E N T E   * * * * *
```

Figura 27. Programa de ejemplo en RPG/400 DBGPGM

El primer punto de interrupción se utiliza para indicarle su situación dentro del programa. La Figura 28 muestra las dos pantallas que se crean tras alcanzar el primer punto de interrupción.

```
-----
                          Visualizar Punto de Interrupción
-----
Sentencia/Instrucción . . . . . : 1200 /004A
Programa . . . . . : DBGPGM
Nivel recurrencia . . . . . : 1
Posición inicio . . . . . : 1
Formato . . . . . : *CHAR
Longitud . . . . . : *DCL

Variable . . . . . : *IN
Límites inferior/superior . . . . . : (1:99)
Tipo . . . . . : CHARACTER
Longitud . . . . . : 1
Elemento ----- Valores -----
1 '1' '0' '0' '0' '0' '0' '0' '0' '0' '0'
11 '0' '0' '0' '0' '0' '0' '0' '0' '0' '0'
21 '0' '0' '0' '0' '0' '0' '0' '0' '0' '0'
31 '0' '0' '0' '0' '0' '0' '0' '0' '0' '0'

Pulse Intro para continuar.
F3=Salir programa F10=Entrada mandatos
-----
                          Más...
-----
```

```

                                Visualizar Punto de Interrupción

Sentencia/Instrucción . . . . . : 1200 /004A
Programa . . . . . : DBGPGM
Nivel recurrencia . . . . . : 1
Posición inicio . . . . . : 1
Formato . . . . . : *CHAR
Longitud . . . . . : *DCL

                                41 '0' '0' '0' '0' '0' '0' '0' '0' '0' '0'
                                51 '0' '0' '0' '0' '0' '0' '0' '0' '0' '0'
                                61 '0' '0' '0' '0' '0' '0' '0' '0' '0' '0'
                                71 '0' '0' '0' '0' '0' '0' '0' '0' '0' '0'
                                81 '0' '0' '0' '0' '0' '0' '0' '0' '0' '0'
                                91 '0' '0' '0' '0' '0' '0' '0' '0' '0'

Pulse Intro para continuar.

F3=Salir programa F10=Entrada mandatos
    
```

Figura 28. Pantalla Primer Punto de Interrupción para DBGPGM

La Figura 29 muestra las dos pantallas que se crean tras alcanzar el segundo punto de interrupción.

```

                                Pantalla P. Interrupción

Sentencia/Instrucción . . . . . : 1500 /0060
Programa . . . . . : DBGPGM
Nivel repetición . . . . . : 1
Posición inicio . . . . . : 1
Formato . . . . . : *HEX
Longitud . . . . . : *DCL

Variable . . . . . : FLD1
  Tipo . . . . . : CHARACTER
  Longitud . . . . . : 1
  * . . . + . . . 1 . . . + . . . *...+...1...+.
  404000063F

Variable . . . . . : PART
  Tipo . . . . . : CHARACTER
  Longitud . . . . . : 1

Pulse Intro para continuar.

F3=Salir programa F10=Entrada mandatos
                                Más...
    
```

```

                                Visualizar P. Interrupción

Sentencia/Instrucción . . . . . : 1500 /0060
Programa . . . . . : DBGPGM
Nivel repetición . . . . . : 1
Posición inicio . . . . . : 1
Formato . . . . . : *HEX
Longitud . . . . . : *DCL

  * . . . + . . . 1 . . . + . . . *...+...1...+.
  404000063F
    
```

```
Pulse Intro para continuar.  
F3=Salir programa  F10=Entrada mandatos
```

Figura 29. Pantalla Segundo Punto de Interrupción para DBGPGM

En este punto, puede modificar el valor de una de estas variables para alterar la ejecución del programa. Después de llegar a la Pantalla de Entrada de Mandatos pulsando F10, puede utilizar el mandato de CL CHGPGMVAR (Cambiar variable de programa) para cambiar el valor de una variable.

4.4.2 Consideraciones al utilizar puntos de interrupción

Debería conocer las siguientes características de los puntos de interrupción antes de utilizarlos:

- Si un punto de interrupción forma parte de una sentencia condicionada, la solicitud de punto de interrupción se procesa aunque no se cumpla la condición.
- Si un punto de interrupción se elude mediante una operación **GOTO**, el punto de interrupción no se procesa.
- Algunas sentencias que no pueden procesarse no representan una posición definida en el flujo lógico del programa. Evite colocar puntos de interrupción en operaciones **PLIST**, **PARM**, **KLIST**, **KFLD** y **DEFN**.
- Cuando se solicita un punto de interrupción para una sentencia, el punto de interrupción se produce antes de ejecutar la sentencia.
- Cuando se solicita un punto de interrupción para una sentencia que no se puede procesar, como por ejemplo una operación **TAG**, el punto de interrupción se sitúa en la próxima sentencia.
- Las funciones de punto de interrupción se especifican mediante mandatos de **CL**. Puede utilizar mandatos de **CL** para añadir puntos de interrupción a programas, visualizar información sobre puntos de interrupción, eliminar puntos de interrupción de programas y arrancar un programa después de visualizar un punto de interrupción. Consulte la publicación *CL Reference* para obtener descripciones de estos mandatos así como una descripción más detallada de los puntos de interrupción.
- Los campos de entrada que el programa no utilice no pueden especificarse en el parámetro **PGMVAR** de los mandatos de depuración. Puede visualizar todo el almacenamiento intermedio de entrada y salida de un registro utilizando el nombre de variable **ZZnnBIN** (almacenamiento intermedio de entrada), **ZZnnBOUT** (almacenamiento intermedio de salida). El valor **nn** es el número de secuencia correspondiente al orden en el que los archivos están definidos en las especificaciones. Este número también aparece en la sección de referencias cruzadas del listado del compilador. Por tanto, puede visualizar el almacenamiento intermedio de entrada del segundo archivo del programa especificando **PGMVAR (ZZ02BIN)**.

4.5 Utilización de un rastreo

Puede utilizar un rastreo para registrar las sentencias que se ejecutan en un programa así como los valores de las variables utilizadas en las sentencias.

Para utilizar un rastreo, especifique qué sentencias y variables debe rastrear el sistema. También puede especificar que se rastreen las variables sólo cuando se modifique su valor. Puede especificar el rastreo de una sentencia, de un grupo de sentencias o de todo un programa. Debe solicitar una pantalla de la información sobre el rastreo. La pantalla muestra la secuencia en la que se han ejecutado las sentencias y, si lo solicita, los valores de las variables utilizadas en las sentencias. La Figura 30 muestra la configuración de un rastreo para sentencias de programa y el orden en que se procesan.

Programa	Rastreo		
Sentencia	Orden de proceso	Variables	
1	1	-----	-----
2	6	-----	-----
3	7	-----	-----
4	8	-----	-----
5	6	-----	-----
6	7	-----	-----
7	2	-----	-----
8	6	-----	-----
.	7	-----	-----
.	.		
.	.		

Figura 30. Sentencias del programa y orden de proceso

Subtemas

4.5.1 Ejemplo de utilización de un rastreo

4.5.2 Consideraciones al utilizar un rastreo

4.5.1 Ejemplo de utilización de un rastreo

La Figura 27 en el tema 4.4.1 muestra parte de un listado del programa de RPG/400 DBGPGM. El mandato de CL que añade un rastreo de las sentencias 1000 a la 1800 en el programa es:

```
ADDTRC STMT((1000 1800))
```

La Figura 31 es un ejemplo de una pantalla de la información sobre el rastreo. El mandato de CL que visualiza esta información es:

```
DSPTRCDTA OUTPUT(*)
```

Visualizar Datos Rastreo				
Programa	Sentencia/ Instrucción	Nivel	Recurrencia	Número Secuencia
DBGPGM	1000		1	1
DBGPGM	1200		1	2
DBGPGM	1300		1	3
DBGPGM	1400		1	4
DBGPGM	1500		1	5
DBGPGM	1600		1	6
DBGPGM	1000		1	7
DBGPGM	1200		1	8
DBGPGM	1800		1	9

Pulse Intro para continuar.
F3=Salir F12=Cancelar

Figura 31. Pantalla Datos de Rastreo para DBGPGM

4.5.2 Consideraciones al utilizar un rastreo

Debería conocer las siguientes características de los rastreos antes de utilizarlos:

- Una sentencia condicionada queda registrada en el rastreo aunque no se cumplan las condiciones.
- No se incluyen en el rastreo las sentencias eludidas por operaciones **GOTO**.
- Las funciones de rastreo se especifican mediante mandatos CL en el trabajo que contiene el programa que se va a rastrear. Estas funciones incluyen la adición de solicitudes de rastreo a un programa, la eliminación de solicitudes de rastreo de un programa, la eliminación de los datos acumulados por rastreos anteriores, la visualización de la información sobre el rastreo y la visualización de los rastreos que se han especificado para un programa.
- No se puede visualizar una variable a la cual no se haga referencia en el programa RPG/400.

4.6 Utilización de códigos de operación **DEBUG**

Puede codificar una o varias operaciones **DEBUG** entre los cálculos del RPG/400 para que le ayuden a depurar un programa que no funciona correctamente. Cuando se procesa una operación **DEBUG**, se proporcionan uno o dos registros con información sobre la depuración. El primer registro contiene una lista de todos los indicadores activos en el momento en que se encontró la operación **DEBUG**. El segundo registro es optativo y muestra el contenido de los archivos de resultado especificados para la operación **DEBUG**.

La operación **DEBUG** puede codificarse en uno o varios puntos de las especificaciones de cálculo. Los registros de salida se escriben cada vez que se lleva a cabo la operación **DEBUG**.

Debe conocer las siguientes características del código de la operación **DEBUG** antes de utilizarlo:

- La operación **DEBUG** se ejecuta (se activa) solamente si la posición 15 de la especificación de control contiene un 1.
- Si la operación **DEBUG** está condicionada, sólo se ejecutará si se cumplen las condiciones.
- Si se elude una operación **DEBUG** mediante una operación **GOTO**, la operación **DEBUG** no se ejecutará.

Las pruebas del sistema OS/400 y las funciones de depuración son aplicables a programas que utilizan operaciones **DEBUG**; un punto de interrupción puede encontrarse en una operación **DEBUG** y puede rastrearse una operación **DEBUG**.

4.7 Utilización del vuelco con formato en RPG/400

Para obtener un vuelco con formato (salida impresa del almacenamiento) en RPPG/400 durante la ejecución de un programa, puede codificar uno o más códigos de operación **DUMP** en los cálculos o puede responder con la opción **D** o **F** a un mensaje en tiempo de ejecución. También es posible responder automáticamente a fin de obtener un vuelco. Consulte en la publicación *CL Guía del Programador* el apartado "Lista de respuestas del sistema".

El vuelco con formato incluye el contenido de campos, de estructuras de datos, de tablas y de matrices así como la estructura de datos de información de archivo y la estructura de datos del estado del programa. El vuelco se graba en el archivo denominado QPPGMDMP. (Un vuelco anormal del sistema se graba en el archivo QPSRVDMP).

Si el usuario responde con la opción **F** a un mensaje en tiempo de ejecución del RPG/400, el vuelco también incluye la representación hexadecimal de la vía de datos abierta (ODP, un bloque de control de gestión de datos). Si la posición 15 de la especificación de control contiene un 1, la opción **F** también proporciona un listado de los campos generados por el compilador.

Se proporciona información sobre la estructura de datos de información de archivo (INFDS) para cada archivo en el programa. Sin embargo, en el vuelco no se imprime toda la información contenida en la INFDS. Recuerde que para utilizar en un programa cualquier información de la INFDS, tiene que definir la INFDS en el programa.

Las mismas características que se describen para la operación **DEBUG** son aplicables a la operación **DUMP**.

La Figura 32 muestra un ejemplo de un vuelco con formato del RPG/400.

```
+--- Advertencia -----+
| A continuación se muestran sólo unas páginas seleccionadas de un
| vuelco con formato del RPG/400.
+-----+
```

RPG/400 FORMATTED DUMP

Area de estado dle programa:

Nombre de programa	: QGPL/SAMPLE	A		
Estado de programa	: 00000	B		
Estado anterior	: 00000	C		
Sentencia errónea	: 00000000	D		
Rutina RPG	: *DETC	E		
Número de parámetros	: 000			
Tipo de mensaje		F		
Número de sentencia MI		G		
Información mensaje adicional				
Datos del mensaje				Inform.
Ultimo archivo utilizado	: QSYSVRT		<input type="checkbox"/> -----+	Estado
Ultimo estado de archivo	: 01235			de Programa
Error en entradas PRTCL producido en (C G S D).		H		
Ultima operación en archivo	: OPEN I			
Ultima rutina de archivo	: *INIT			
Ultima sentencia de archivo	: *INIT			
Nombre último registro de archivo			<input type="checkbox"/> -----+	
Nombre de trabajo	: E53		<input type="checkbox"/> -----+	
Nombre de usuario	: QPGMR			
Número de trabajo	: 000811			
Fecha sistema entrado	: 092592			
Fecha inicio	: 092592			
Hora inicio	: 111143			I
Fecha compilación	: 052592			
Hora compilación	: 111125			
Nivel del compilador	: 0001			
Archivo fuente	: QRPGSRC			
Biblioteca	: QGPL			
Miembro	: SAMPLE		<input type="checkbox"/> -----+	

RPG/400 VUELCO CON FORMATO

```
Archivo. . . . . : FILEIN1
Apertura archivo . . . . . : YES
Archivo en Fin de archivo. . . . . : YES
Compromiso activo. . . . . : NO
Estado del archivo . . . . . : 00011
    RPG0011 Fin de archivo (entrada).
Operación de archivo . . . . . : READ R
Rutina de archivo. . . . . : *DETC
Número de sentencia. . . . . : 2500
Nombre de registro . . . . . : FILEA
```

```

Identificador mensaje. . . . . :
Número instrucción MI . . . . . :
Tipo ODP . . . . . : DB
Nombre de archivo. . . . . : FILEIN1
  Biblioteca. . . . . : QGPL
Miembro. . . . . : FILEIN1
Formato registro . . . . . :
Longitud registro primario . . . . . : 45
Longitud registro secundario . . . . . : 0
Longitud bloque entrada. . . . . : 4125
Longitud bloque salida . . . . . : 0
Clase de dispositivo . . . . . : '0000'X
Líneas por página. . . . . : 0
Columnas por línea . . . . . : 0
Número registros en archivo. . . . . : 0
Tipo de acceso . . . . . : ARRIVAL SEQ
Permitir claves duplicadas . . . . . : NO
Archivo fuente . . . . . : NO
Parámetro UFEB . . . . . : 'A20000000000000500000'X
Alteraciones temporales UFEB . . . . . : '00000000000000000000'X
Registros a transferir . . . . . : 74
Número de PUTS . . . . . : 0
Número de GETS . . . . . : 0
Número de PUTS/GETS. . . . . : 0
Número de otras E/S . . . . . : 0
Operación actual . . . . . : '4040'X
Clase de dispositivo . . . . . : '4040'X
Nombre de dispositivo. . . . . :
Longitud último registro . . . . . : 0
Información sobre DDS. . . . . :
Número registro relativo . . . . . : 0
Registros transferidos . . . . . : 0
Número de línea actual . . . . . : 0
  
```

```

Almac. interm. entrada:
0000 80000000 00000000 0007C00D BD000880 0000004A 0037002D 40404040 40404040 *
0020 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0060 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0080 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
00A0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
00C0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
00E0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0100 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0120 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0140 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0160 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
  
```

RPG/400 VUELCO CON FORMATO

```

0180 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
01A0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
01C0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
01E0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0200 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0220 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0240 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0260 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0280 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
02A0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
02C0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
02E0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0300 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0320 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0340 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0360 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0380 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
03A0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
03C0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
03E0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0400 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0420 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0440 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0460 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0480 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
04A0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
04C0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
04E0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0500 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0520 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0540 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0560 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
0580 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
05A0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *
  
```

Utilización del vuelco con formato en RPG/400

05C0	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
05E0	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
0600	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
0620	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
0640	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
0660	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
0680	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
06A0	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
06C0	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
06E0	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
0700	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
0720	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
0740	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
0760	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
0780	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
07A0	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
07C0	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
07E0	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
0800	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
0820	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
0840	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*
0860	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	40404040	*

RPG/400 VUELCO CON FORMATO

```

Archivo. . . . . : QSYSVRT
Apertura archivo . . . . . : YES
Archivo en Fin de archivo. . . . . : YES
Compromiso activo. . . . . : NO
Estado de archivo. . . . . : 01235
      Error en PRCTL producido en (C G S D).      J
Operación de archivo . . . . . : OPEN I
Rutina de archivo. . . . . : *INIT
Número de sentencia. . . . . : *INIT
Nombre de registro . . . . . :
Identificador de mensaje . . . . . :
Número instrucción MI . . . . . :
Tipo ODP . . . . . : SP
Nombre de archivo. . . . . : QSYSVRT
  Biblioteca. . . . . : QSYS
Miembro. . . . . : Q713784701
Formato de registro. . . . . :
Archivo de spool . . . . . : Q04079N001
  Biblioteca. . . . . : QSPL
Número de archivo de spool . . . . . : 19
Longitud registro primario . . . . . : 132
Longitud registro secundario . . . . . : 0
Longitud bloque entrada. . . . . : 0
Longitud bloque salida . . . . . : 132
Clase de dispositivo . . . . . : PRINTER
Líneas por página. . . . . : 10
Columnas por línea . . . . . : 132
Número registros en archivo. . . . . : 0
Tipo de acceso . . . . . : 0
Permitir claves duplicadas . . . . . : NO
Archivo fuente . . . . . : NO
Parámetros UFCB . . . . . : 'A4121000000000000000'X
Alteraciones temporales UFCB . . . . . : '00000000000000000000'X
Número de PUTS . . . . . : 0 □-----+
Número de GETS . . . . . : 0 |-----+ K
Número de PUTS/GETS. . . . . : 0 □-----+
Número de otras E/S . . . . . : 0
Operación actual . . . . . : '4040'X
Clase de dispositivo . . . . . : '4040'X
Nombre de dispositivo. . . . . :
Longitud último registro . . . . . : 0
Información sobre DDS. . . . . :
Número registro relativo . . . . . : 0
Número de línea actual . . . . . : 0

```

Información de Archivo.
 Esta información se rep
 para cada archivo del p
 En la publicación RPG S
 se facilita una descrip
 de estas entradas.

```

Almac. interm. entrada:
0000 E2C1D4D7 D3C54040 4040D8C7 D7D34040 40404040 00000000 00000000 00000000 *SAMPLE
0020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *
0040 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *
0060 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *
0080 00000000 *
Almac. interm. salida:
0000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *
0020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *
0040 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *
0060 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *
0080 00000000 *

```


ALMACENAMIENTO AUTOMATICO PARA PROGRAMA SAMPLE EMPIEZA EN DESPLAZAMIENTO 0015B0 EN AREA ALMACEN. AUTOM
RPG/400 VUELCO CON FORMATO

* * * * * F I N D E V U E L C O E N R P G * * * * *

Figura 32. Vuelco con formato del RPG/400

- A Nombre calificado del programa y biblioteca.
- B Código de estado actual.
- C Código estado anterior
- D Sentencia fuente del RPG/400 incorrecta.
- E Rutina del RPG/400 en la que se ha producido la excepción o error.
- F CPF o MCH para una excepción de máquina.
- G Número de instrucción de máquina.
- H Información sobre el último archivo utilizado en el programa antes de producirse una excepción o error (RPG1235).
- I Información del programa.
- J Error en el archivo.
- K Número de veces que el compilador del RPG/400 ha solicitado E/S del sistema (no el número de operaciones de E/S solicitadas por el programa).
- L La vía de datos abierta se incluye en el vuelco si el usuario responde con una opción F a un mensaje en tiempo de ejecución del RPG/400.
- M También se incluye en el vuelco una lista de los campos generados por el compilador si el usuario responde con la opción F a un mensaje en tiempo de ejecución en RPG/400 y si el programa se ha compilado con un 1 en la posición 15 de la especificación de control.
- N Indicadores generales 1-99 y sus estados actuales (1 es activado, 0 es desactivado).
- O Comienzo de los campos del usuario.
- P Campos con zona incorrectos impresos en hexadecimal.
- Q Estructuras de datos de información de archivos para FILEIN1 y FILEIN2.
- R Estructura de datos de doble aparición.
- S Valores de fecha del sistema.
- T En el mandato CRTRPGPGM se ha especificado **IGNDECERR(*NO)**.
- U Area de datos del estado del programa.
- V Almacenamiento intermedio de entrada para el archivo 02.
- W Este es el número de archivo. Consulte la sección de referencias cruzadas del listado del compilador para obtener el nombre del archivo correspondiente. Los archivos tienen asignado un número de secuencia correspondiente al orden en que están definidos en las especificaciones. De este modo, el número de archivo 03 corresponde al archivo FILEIN2 descrito en este programa.
- X Almacenamiento intermedio de salida para el archivo 03.

4.8 Manejo de excepciones/errores

El compilador del RPG/400 maneja dos tipos de excepciones o errores: excepciones o errores de programa y excepciones o errores de archivo. Algunos ejemplos de excepciones/errores de programa son: división por cero, índice de matriz no válido o **SQRT** (raíz cuadrada) de un número negativo. Algunos ejemplos de excepciones o errores de archivo son tipos de registro no definidos o un error de dispositivo.

La Figura 33 muestra un ejemplo de una estructura de datos de información de archivo (INFDS) y una subrutina de excepciones/errores de un archivo. Para obtener más información sobre el manejo de excepciones/errores mediante el compilador del RPG/400, consulte el manual *RPG/400 Reference*.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*
F* Existen tres archivos definidos en las especificaciones de descripción
F* de archivo. Necesita controlar la lógica del programa si se produce
F* una excepción o error en los archivos TRNFIL o MSTFIL. Por lo tanto,
F* se define una INFDS y una INFSR exclusiva para cada archivo. Estos
F* no se definen para el archivo AUDITFIL.
F*
FNombarchIPEAF....LregLCIAIDePcLaEDispos+.....KSalidaEntradA....U1.*
FTRNFIL OF E K DISK KINFDS FILDS1
F KINFSR ERRRTN
FMSTFIL UF E K DISK KINFDS FILDS2
F KINFSR MSTERR
FAUDITFILOF E K DISK

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I*
I* La ubicación de los subcampos en las estructuras de datos de
I* información de archivo está definida por teclas clave especiales
I* entre las posiciones 44 y 51. Para acceder a estos subcampos
I* predefinidos debe añadir un subcampo entre las posiciones 53 y 58.
I* Si se produce un error o excepción, puede probar la información
I* de la estructura de datos para determinar, por ejemplo, qué error
I* o excepción se ha producido (*STATUS) y en qué operación se ha
I* producido (*OPCODE). Entonces puede utilizar esta información en
I* la subrutina de excepciones/errores para determinar la acción
I* que debe adoptar.
INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
ICPOED1 DS
I.....PDesdeA+++DNomcamL1M1FrP1NeCe...*
I *FILE FIL1
I *FILE ARC1
I *OPCODE OP1
I *STATUS STS1
I *ROUTINE RTN1
ICPOED2 DS
I *FILE ARC2
I *RECORD REC2
I *OPCODE OP2
I *STATUS STS2
I *ROUTINE RTN2

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* En la operación WRITE del registro TRNREC del archivo TRNFIL, se
C* ha especificado un indicador de excepción/error en las posiciones
C* 56 y 57. Este indicador se activa si se produce una excepción o
C* error en esta operación. La operación EXSR llama explícitamente
C* a la subrutina ERRRTN (la subrutina para excepciones y errores del
C* archivo TRNFIL) cuando el indicador 71 está activado.
C* Como el factor 2 de la operación ENDSR para ERRRTN está
C* en blanco, el control se devuelve a la siguiente instrucción
C* secuencial que sigue a la operación EXSR después de haberse
C* ejecutado la subrutina.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C WRITETRNR 71
C 71 EXSR ERRRTN +-----+
C " | Cálculos |
C* +-----+
C*
C* No se especifica ningún indicador de excepción/error en las
C* posiciones 56 y 57 de la operación WRITE del registro AUDITREC del
C* archivo AUDITFIL. No se ha definido ninguna subrutina de
C* excepción/error para este archivo en las especificaciones de
C* descripción del archivo. Por lo tanto, el manejador de errores
C* por omisión del RPG por omisión no maneja ninguna excepción/error
C* que se produzca en esta operación para el archivo AUDITFIL.
C*
```

```

C          WRITEAUDITREC          +-----+
C          "                      |  Cálculos  |
C*         +-----+

```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

C*

C* En las posiciones 56 y 57 no se ha especificado ningún indicador de
C* excepción/error para la operación CHAIN del registro REGMAE
C* del archivo AMAEST. Sin embargo, se ha definido para el archivo
C* una subrutina de excepción/error de archivo (ERRMAE) en
C* las especificaciones de descripción de archivos. Por lo tanto,
C* cuando en la operación CHAIN se produce una condición de
C* excepción/error, distinta de la de registro no encontrado, el RPG
C* transfiere el control a la subrutina ERRMAE. En la operación
C* ENDSR para esta subrutina, el factor 2 contiene un nombre de
C* campo. Esto permite al programador alterar en la propia subrutina
C* el punto de retorno de la subrutina en función de la excepción o
C* error que se ha producido. Dicho campo tiene que contener uno de
C* los valores descritos en el apartado **Subrutina de excepciones/**
C* errores de archivo

```

CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C          CLAMAE          CHAINREGMAE          61
C  61          GOTO NOENC
C          "
C          "
C          ERRMAE          BEGSR          +-----+
C          "                      |  Cálculos  |
C          "                      +-----+
C          ENDSRPUNRET
C          "
C          "
C          RUTERR          BEGSR          +-----+
C          "                      |  Cálculos  |
C          "                      +-----+
C          ENDSR

```

Figura 33. Ejemplo de manejo de excepciones/errores de archivo

La Figura 34 muestra un ejemplo de una subrutina de excepciones/errores de programa.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
IDsnombre..NODsSal-arch++.....NvecLon+.....*
I          SDS
I          *ROUTINE SIT
I          *STATUS ERROR
I          *PARMS PARAM
I          *PROGRAM NOMBR

```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C          *PSSR          BEGSR
C          ERR          COMP 102          20 DIV POR CERO?
C  20          ADD 1          DIVSR
C  20          MOVE '*DETC' RETURN 6
C  N20          MOVE '*CANCL' RETURN
C          ENDSRRETURN

```

Figura 34. Ejemplo de subrutina *PSSR

La estructura de datos de estado del programa está definida en las especificaciones de entrada. Los subcampos predefinidos ***STATUS**, ***ROUTINE**, ***PARMS**, y ***PROGRAM** están especificados y se han asignado nombres a los subcampos.

La subrutina ***PSSR** está codificada en las especificaciones de cálculo. Si se produce una excepción/error de programa, el compilador del RPG/400 pasa el control a la subrutina ***PSSR**. La subrutina comprueba si la excepción o error se ha producido a causa de una operación de división en la cual el divisor sea cero. Si lo ha sido, se activa el indicador 20, se añade 1 al divisor (**DIVSR**) y el literal **'*DETC'**. se mueve al campo **RETURN**. Cuando se mueve el literal al campo **RETURN**, que está especificado en el factor 2 de la operación **ENDSR**, se permite que el usuario controle el punto de retorno dentro de la subrutina. En este ejemplo, el control regresa al principio de la rutina de cálculos de detalle, a menos que la excepción o error no fuese la división por cero. En ese caso, el literal **'*CANCL'** se mueve al campo **RETURN** y se termina el programa.

5.0 Capítulo 5. Consideraciones generales sobre archivos

En este capítulo se describen:

- Las características de independencia y dependencia de dispositivo del programa RPG/400 en el sistema AS/400
- Las funciones de spooling del AS/400
- El detalle con que se definen en el programa RPG/400 los archivos descritos externamente y descritos en el programa
- Las funciones de comprobación de nivel
- El bloqueo de archivo por el programa RPG/400
- El bloqueo de registro por el programa RPG/400
- La agrupación/desagrupación en bloques de registros para mejorar el rendimiento
- El compartimiento de una vía de datos abierta
- Información general sobre la utilización de archivos descritos externamente y cómo puede modificarse esta descripción externa en el programa RPG/400.
- Archivos descritos por programa
- Las funciones del RPG/400 que se relacionan específicamente con un dispositivo PRINTER, un dispositivo SEQ y un dispositivo SPECIAL del RPG/400.

En el sistema AS/400, los archivos están formados por miembros. Estos archivos se organizan en bibliotecas. El convenio de denominación de archivos es **nombre de biblioteca/nombre de archivo**.

Subtemas

- 5.1 Independencia de dispositivo/dependencia de dispositivo
- 5.2 Spooling
- 5.3 Archivos descritos externamente y descritos por programa
- 5.4 Comprobación de nivel
- 5.5 Bloqueo de archivo por un programa en RPG/400
- 5.6 Bloqueo de registro por un programa en RPG/400
- 5.7 Desagrupación de bloques de registros de entrada y agrupación en bloques de registros de salida
- 5.8 Compartimiento de una vía de datos abierta
- 5.9 Utilización del mandato de language de control RCLSRC
- 5.10 Especificaciones para archivos descritos externamente
- 5.11 Archivos descritos por programa
- 5.12 Archivos de impresora
- 5.13 Archivo secuencial
- 5.14 Archivo especial

5.1 Independencia de dispositivo/dependencia de dispositivo

El elemento clave para todas las operaciones de E/S es el archivo. Todos los archivos utilizados en el sistema tienen que estar definidos en el sistema OS/400. El sistema OS/400 mantiene una descripción de cada archivo al que accede un programa cada vez que dicho programa utiliza el archivo.

Las descripciones de archivos del OS/400 se mantienen en línea y sirven como enlace de conexión entre el programa y el dispositivo que se utiliza para las operaciones de E/S. Cuando en un proceso se utiliza el archivo, los datos se leen del dispositivo o se graban en él. En algunos casos este tipo de control de E/S permite al programador cambiar el tipo de archivo (y, en algunos casos, cambiar el dispositivo) utilizado en un programa sin necesidad de cambiar el programa.

En el sistema AS/400, se utiliza el nombre del archivo, que se entra en las posiciones 7 a 14, para hacer referencia al archivo en vez del nombre del dispositivo, que se entra en las posiciones 40 a 46 de las especificaciones de descripción de archivos. El nombre del archivo hace referencia a la descripción de archivo del OS/400 que contiene las especificaciones para el dispositivo real.

IMAGEN 12

El nombre de dispositivo de RPG/400 entre las posiciones 40 y 46 define las funciones que se pueden procesar en el archivo asociado. En el tiempo de compilación, ciertas funciones del RPG/400 sólo son válidas para un nombre de dispositivo de RPG/400 determinado. En este caso, la función del RPG/400 es dependiente de dispositivo. Otro ejemplo de dependencia de dispositivo es el código de operación **EXFMT**, que sólo es válido para un dispositivo WORKSTN.

Para otro ejemplo, suponga que se especifica el nombre de programa **ARCHY** en el programa del RPG/400 con el dispositivo SEQ. El dispositivo SEQ es un tipo de dispositivo independiente. Cuando se ejecuta el programa, el dispositivo de E/S actual se especifica en la descripción de **ARCHY**. Por ejemplo, el dispositivo podría ser PRINTER.

IMAGEN 13

Se pueden utilizar los mandatos del OS/400 para alterar temporalmente un parámetro en la descripción de archivos especificada o para redireccionar un archivo en tiempo de compilación o ejecución. El redireccionamiento de archivos le permite especificar un archivo en tiempo de compilación y otro archivo en tiempo de ejecución:

IMAGEN 14

En el ejemplo anterior, el mandato de CL OVRDBF (Alterar temporalmente con archivo de base de datos) permite la ejecución del programa con un archivo de dispositivo completamente distinto al especificado en el tiempo de compilación.

No todos los redireccionamientos de archivos son válidos. En tiempo de ejecución, las comprobaciones aseguran que las especificaciones del programa en RPG/400 sean válidas para el archivo que se procesa. El sistema OS/400 permite algunos redireccionamientos de archivos aunque el programa contenga dispositivos específicos. Por ejemplo, si el nombre del dispositivo del RPG/400 es PRINTER, y el archivo actual al que está conectado el programa no es una impresora, el sistema OS/400 ignora las especificaciones de espaciado y salto de impresión del RPG/400. Existen otros redireccionamientos de archivo no permitidos por el sistema OS/400 y que causan la terminación del programa. Por ejemplo, si el nombre del dispositivo del RPG/400 es WORKSTN y en el programa se especifica la operación **EXFMT**, el programa se detiene si el archivo real al que está conectado el programa no es un archivo de pantalla o de ICF.

Consulte la publicación *Guía para la Gestión de Datos* para obtener información más detallada sobre los redireccionamientos de archivo válidos así como las alteraciones temporales de los archivos.

5.2 Spooling

El spooling es una función del sistema que coloca los datos en un área de almacenamiento mientras esperan el proceso. El sistema AS/400 proporciona la utilización de funciones de spooling de entrada y de salida. El programa en RPG/400 no detecta la utilización del spooling. El dispositivo físico real desde el que se lee un archivo o en el que se graba un archivo está determinado por el lector de spool o por el transcriptor de spool. Para obtener información más detallada sobre el spooling, consulte la publicación *Guía para la Gestión de Datos*.

Subtemas

5.2.1 Spool de salida

5.2.1 Spool de salida

El spooling de salida es válido para trabajos de proceso por lotes e interactivos. La descripción de archivos que el nombre de archivo especifica en el programa en RPG/400 contiene la especificación para el spooling tal como se indica en el siguiente diagrama:

IMAGEN 15

Puede utilizarse mandatos de alteración temporal en tiempo de ejecución para alterar temporalmente las opciones de spooling especificada en la descripción de archivo, como por ejemplo el número de copias que se va a imprimir. Además el soporte de spooling del AS/400 le permite redireccionar un archivo después de que un programa lo ejecute. Puede dirigir una misma salida impresa a un dispositivo distinto como por ejemplo un disquete.

5.3 Archivos descritos externamente y descritos por programa

Todos los archivos de sistema AS/400 están definidos en el sistema RPG/400. Sin embargo, el detalle con que pueden definirse los archivos varía:

- Un *archivo descrito externamente* se describe en el sistema OS/400 a nivel de campo. La descripción incluye información sobre la procedencia de los datos, como por ejemplo la base de datos o un dispositivo específico, así como una descripción de cada campo y sus atributos.
- Un *archivo descrito por programa* está descrito a nivel de campo en el programa en RPG/400 en las especificaciones de entrada/salida. La descripción de archivos para el sistema RPG/400 incluye información acerca de la procedencia de los datos y la longitud de los registros en el archivo.

Un archivo descrito externamente no tiene que redefinirse en un programa en RPG/400 en las especificaciones de entrada/salida. En un archivo descrito por programa, los campos y sus atributos deben estar descritos en las especificaciones de entrada/salida.

Los archivos descritos externamente ofrecen las siguientes ventajas:

- Menor codificación de los programas en RPG/400. Si muchos programas utilizan un mismo archivo puede definir los campos sólo una vez para el sistema RPG/400 para que todos los programas los utilicen. Esto elimina la necesidad de codificar especificaciones de entrada y salida para los programas en RPG/400 que utilizan archivos descritos externamente.
- Menor actividad de mantenimiento al cambiar el formato de registro del archivo. El usuario puede actualizar frecuentemente los programas cambiando el formato de registro del archivo así como los programas que utilizan los archivos sin cambiar ninguna codificación en el programa.
- Mejor documentación, ya que los programas que utilizan los mismos archivos utilizan formatos de registro y nombres de campos congruentes.

Si se especifica un archivo descrito externamente (identificado por una E en la posición 19) para los dispositivos SEQ o SPECIAL, el programa en RPG/400 utiliza las descripciones de campo para el archivo, pero para la interfaz del sistema OS/400 es como si se tratara de un archivo descrito por programa. Los archivos descritos externamente no pueden especificar funciones dependientes de dispositivo como por ejemplo control de formulario.

Puede optar por utilizar un archivo descrito externamente en el programa especificando el archivo como descrito por programa con una **F** en la posición 19 de las especificaciones de descripción de archivo. El compilador no realiza una copia en la descripción externa a nivel de campo del archivo en tiempo de compilación. Este enfoque se utiliza en conversiones en las que programas existentes utilizan archivos descritos por programa y nuevos programas utilizan archivos descritos externamente para hacer referencia al mismo archivo.

La Figura 35 muestra algunas relaciones típicas entre un programa en RPG/400 y archivos del sistema AS/400.

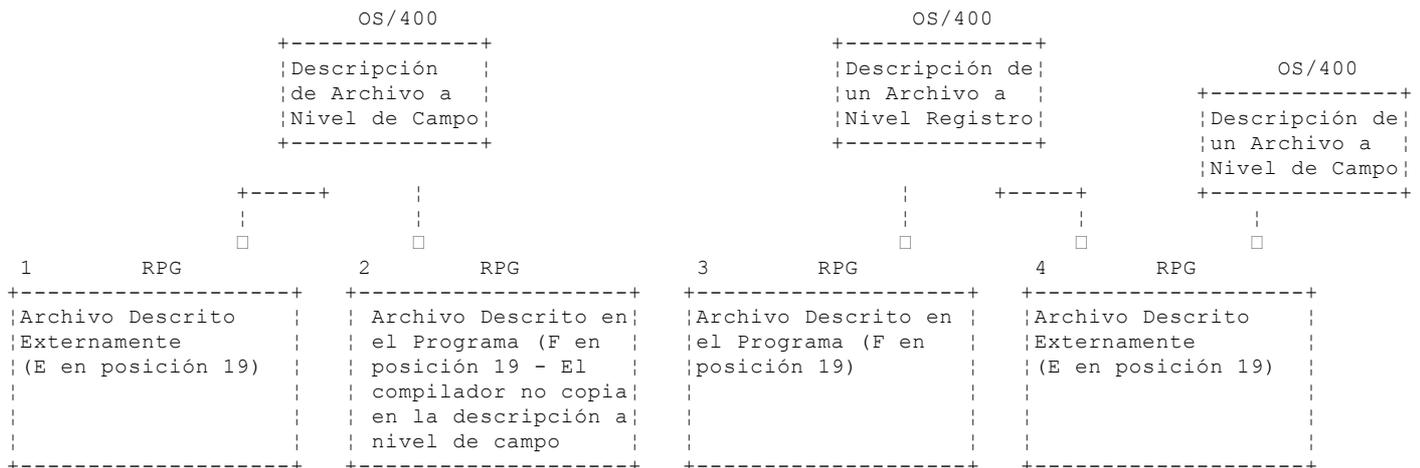


Figura 35. Relación típica entre un programa en RPG/400 y archivos del sistema AS/400

- 1 El programa en RPG/400 utiliza la descripción de un archivo a nivel de campo definida en el sistema OS/400. Un archivo descrito externamente se identifica mediante una **E** en la posición 19 de las especificaciones de descripción de archivo. En tiempo de compilación, el compilador realiza la copia en la descripción externa a nivel de campo.
- 2 Se utiliza un archivo descrito externamente como un archivo descrito por programa en el programa en RPG/400. Un archivo descrito en el programa se identifica mediante una **F** en la posición 19 de las especificaciones de descripción de archivo. Esta entrada indica al compilador que no realice la copia en las descripciones externas a nivel de campo. No es necesario que este archivo exista en tiempo de compilación.
- 3 Se describe un archivo al sistema RPG/400 sólo a nivel de registro. Los campos del registro se describen en el programa en RPG/400; por lo tanto, la posición 19 de las especificaciones de descripción de archivos tiene que contener una **F**. No es necesario que este archivo exista en tiempo de compilación.
- 4 Un nombre de archivo puede especificarse para el tiempo de compilación y un nombre de archivo distinto puede especificarse para el tiempo de ejecución. La **E** de la posición 19 de las especificaciones de descripción de archivo indica que la descripción externa del archivo ha de copiarse en tiempo de compilación. En tiempo de ejecución, puede utilizarse un mandato de alteración temporal de archivo para que el programa pueda acceder a un archivo distinto. Para alterar temporalmente un archivo en tiempo de ejecución debe asegurarse de que los nombres de los registros de ambos campos sean iguales. El programa en RPG/400 utiliza el nombre del formato de registro en las operaciones de entrada/salida, como por ejemplo una operación **READ** en la que especifica qué tipo de registro se espera.

A continuación se muestra un ejemplo de la utilización de alteración temporal de archivo en tiempo de compilación. Suponga que desea utilizar un archivo descrito externamente para un dispositivo TAPE al que el sistema no da soporte. Para ello debe:

- Definir un archivo físico denominado **FMT1** con el formato de registro que contenga la descripción de cada campo en el formato del registro. El formato de registro está definido en las especificaciones de descripción de datos (DDS). El archivo descrito externamente para un dispositivo de cinta deberá contener solamente un formato de registro.
- Crear el archivo denominado **FMT1** mediante el mandato de CL Crear archivo físico.
- Especificar el nombre de archivo **QTAPE** (el nombre del archivo de dispositivo suministrado por IBM para dispositivo de cinta magnética) en el programa en RPG/400. Este identifica el archivo como descrito externamente (como indica una **E** en la posición 19 de las especificaciones de descripción de archivo), y especifica el nombre de dispositivo **SEQ** entre las posiciones 40 y 46.
- Utilizar un mandato de alteración temporal **-OVRDBF FILE(QTAPE) TOFILE(FMT1)**-en tiempo de compilación para alterar temporalmente el nombre de archivo **QTAPE** y utilizar el nombre de archivo **FMT1**. Este mandato hace que el compilador realice una copia en la descripción externa del archivo **FMT1**, que describe el formato de registro al compilador del RPG/400.
- Crear el programa en RPG/400 utilizando el mandato **CRTRPGPGM**.
- Llamar al programa en tiempo de ejecución. La alteración temporal al archivo **FMT1** no debe entrar en efecto mientras se ejecuta el programa. Utilice el mandato CL **DLTOVR** (Suprimir alteración temporal).

Nota: Puede que necesite utilizar el mandato de CL **OVRTAPF** antes de llamar al programa para proporcionar la información necesaria para abrir el archivo de cintas.

5.4 Comprobación de nivel

Como los programas en RPG/400 dependen de la recepción de un archivo descrito externamente cuyo formato se corresponda con el que se ha copiado en el programa en tiempo de compilación, el sistema proporciona un función de comprobación de nivel que se asegura de que el formato sea el mismo.

El programa en RPG/400 siempre proporciona la información que la comprobación de nivel necesita cuando se utiliza un archivo descrito externamente DISK, WORKSTN o PRINTER. Puede solicitar la función de comprobación de nivel en los mandatos crear, cambiar y alterar temporalmente archivo. El valor por omisión del mandato crear archivo es solicitar la comprobación de nivel. La comprobación de nivel se produce en la base de formato de registro cada vez que se abre el archivo a menos que especifique **LVLCHK(*NO)** al emitir un mandato Alterar temporalmente archivo o Crear archivo. Si los valores de comprobación de nivel no coinciden, se comunica el error al programa. Entonces el programa en RPG/400 maneja el error **OPEN** tal como se describe en la "Manejo de excepciones/errores" en el tema 4.8.

El programa en RPG/400 no proporciona comprobación de nivel para archivos descritos por programa ni para archivos que utilicen los dispositivos SEQ o SPECIAL.

Para obtener más información sobre cómo especificar comprobaciones de nivel, consulte el manual *Guía para la Gestión de Datos*.

5.5 Bloqueo de archivo por un programa en RPG/400

El sistema RPG/400 permite un estado de bloqueo (exclusivo, permitida lectura exclusivamente, compartido para actualización, compartido sin actualización o compartido para lectura) que se situará en un archivo utilizado durante un trabajo. Los programas que están dentro de un trabajo no están afectados por los estados de bloqueo de los archivos. Un estado de bloqueo de archivo sólo se produce cuando un programa en otro trabajo intenta utilizar el archivo al mismo tiempo. Puede asignarse el estado de bloqueo de archivo mediante el mandato de CL ALCOBJ (Asignar objeto). Para obtener más información sobre la asignación de recursos y estados de bloqueo consulte la publicación *Guía para la Gestión de Datos*.

El sistema RPG/400 sitúa los estados de bloqueo siguientes en los archivos de la base de datos al abrir los archivos:

Tipo de archivo	Estado de bloqueo
Entrada	Compartido para lectura
Actualización	Compartido para actualización
Adición	Compartido para actualización
Salida	Compartido para actualización

El estado de bloqueo compartido para lectura, permite que otro usuario abra el archivo con un estado de bloqueo compartido para lectura, compartido para actualización, compartido sin actualización o permitida lectura exclusivamente, pero el usuario no puede especificar el uso exclusivo del archivo. El estado de bloqueo compartido para actualización, permite que otro usuario abra el archivo con un estado de bloqueo compartido para lectura o compartido para actualización.

El programa en RPG/400 sitúa un estado de permitida lectura exclusivamente, para los archivos de dispositivo. Otro usuario puede abrir el archivo con un estado de bloqueo de compartido para lectura.

El programa en RPG/400 puede cambiar el estado de bloqueo situado en el archivo utilizando el mandato Asignar objeto.

5.6 Bloqueo de registro por un programa en RPG/400

Cuando un programa lee un registro, lo lee utilizando una de estas dos modalidades: entrada o actualización. Si un programa lee un registro para actualización, se coloca un bloqueo en ese registro. Otro programa no puede leer el mismo registro para actualizarlo hasta que el primer programa libere ese bloqueo. Si un programa lee un registro para entrada, no se coloca el bloqueo en el registro. Un registro que está bloqueado por un programa puede ser leído para entrada por otro programa.

En los programas en RPG/400, se utiliza un archivo de actualización para leer registros con el fin de actualizarlos. Un registro que se lee desde un archivo con un tipo que no sea el de actualización puede leerse sólo para consulta. Por omisión, cualquier registro que se lea de un archivo de actualización se leerá para actualización. Para archivos de actualización, se puede especificar que se lea un registro para entrada utilizando una de las operaciones de entrada **CHAIN**, **READ**, **READE**, **READP** o **REDPE** y especificando una N en la columna 53 de la especificación de cálculo.

Cuando un programa en RPG/400 bloquea un registro, este registro permanece bloqueado hasta que se produzca uno de los siguientes casos:

- se actualice el registro.
- se suprima el registro.
- se lea otro registro del archivo (ya sea para consulta o actualización).
- se realice una operación **SETLL** o **SETGT** contra el archivo
- se realice una operación **UNLCK** contra el archivo.
- se realice una operación de salida contra el archivo definida por una especificación de salida sin nombres de campo incluidos.

Nota: Una operación de salida que añada un registro a un archivo no tiene como resultado que se libere un bloqueo de registro.

Si el programa lee un registro para actualización y este registro está bloqueado por otro programa del trabajo o por otro trabajo, la operación de lectura esperará hasta que se desbloquee el registro. Si el tiempo de espera sobrepasa el especificado en el parámetro **WAITRCD** del archivo, se produce una excepción. Si el gestor de errores por omisión tiene el control cuando se excede el tiempo de espera de bloqueo de registro, se emite un mensaje de error RPG1218. Una de las opciones listadas para este mensaje es volver a intentar la operación en la que se excedió el tiempo de espera. Para programas compilados para la versión 2 (o posterior) esto hará que la operación en la que se ha excedido el tiempo de espera vuelva a emitirse, lo cual permite al programa continuar esencialmente como si no se hubiera excedido el tiempo. Observe que si el archivo tiene una **INFSR** especificada en la que se produce una operación de E/S en el archivo antes de que se pase el control al gestor de errores por omisión, pueden producirse reasultados inesperados si la operación de entrada que se vuelve a intentar es un operación secuencial, ya que se puede haber modificado el cursor del archivo.

Con los programas compilados utilizando la versión 1 del compilador del RPG/400, la opción de reintento sigue visualizándose pero no es válida. Si se solicita un reintento para el programa en la versión 1, se emite un mensaje de error adicional (RPG1918) que indica que un reintento no es válido para programas que compilen utilizando la versión 1. En todos los casos, si se devuelve el control al programa especificando un retorno a ***GETIN**, el valor **RPG STATUS** se establece en 1218. Si se especifica un reintento y la operación de lectura posterior se realiza satisfactoriamente, el **STATUS** se devuelve como si el bloqueo de registro y el reintento posterior no se hubieran producido.

Si no se requieren cambios en un registro bloqueado, puede liberarlo de su estado de bloqueo utilizando la operación **UNLCK** o procesando las operaciones de salida definidas por las especificaciones de salida sin incluir los nombres de los campos. Estas operaciones de salida pueden procesarse por salida **EXCPT**, salida de detalle o salida de totales.

(Existen excepciones a estas reglas al operar bajo el control de compromiso. Consulte el Capítulo 6, "Control de compromiso" en el tema 6.0 para obtener más información.)

Desagrupación de bloques de registros de entrada y agrupación en bloques de registros de salida

5.7 Desagrupación de bloques de registros de entrada y agrupación en bloques de registros de salida

El compilador del RPG/400 desagrupa de bloques los registros de entrada y agrupa en bloques los registros de salida para mejorar el rendimiento en la ejecución de los archivos SEQ o DISK si:

- El archivo es sólo de salida (se ha especificado una **O** en la posición 15 de las especificaciones de descripción de archivos) y sólo contiene un formato de registro si el archivo está descrito externamente.
- El archivo es un archivo de tablas combinadas. (se ha especificado una **C** en la posición 15 y una **T** en la posición 16 de las especificaciones de descripción de archivos).
- Se trata de un archivo de sólo entrada. (se ha especificado una **I** en la posición 15 de las especificaciones de descripción de archivos). Sólo contiene un formato de registro si el archivo está descrito externamente y sólo utiliza los códigos de operación **OPEN**, **CLOSE**, **FEOD** y **READ**.

El compilador del RPG/400 genera un código de programa de objeto para agrupar y desagrupar en bloques los registros de todos los archivos SEQ o DISK que cumplan estas condiciones. Ciertas restricciones del sistema OS/400 puede que eviten la agrupación y la desagrupación en bloques. En estos casos, no se mejora el rendimiento y el área de realimentación de entrada/salida se actualiza por cada operación de entrada/salida.

El contenido de las posiciones 60 a 65 de la opción **RECNO** de la de continuación de las especificaciones de descripción de archivo puede que no sea válida cuando el compilador del RPG/400 agrupa y desagrupa los registros en bloques.

Las secciones de entrada/salida y de dependencia de dispositivo de la estructura de datos de información de archivo no se actualizan después de cada operación de lectura o grabación en archivos en los que los registros estén agrupados o desagrupados en bloques mediante el compilador del RPG/400. El área de realimentación se actualiza cada vez que se transfiere un bloque de registros. (Consulte el manual *RPG/400 Reference* para obtener más información.)

5.8 Compartimiento de una vía de datos abierta

Una vía de datos abierta es la vía a través de la cual se definen todas las operaciones de entrada y salida de un archivo. Normalmente se define una vía de datos abierta independiente cada vez que se abre un archivo. Si especifica **SHARE(*YES)** en la creación de archivo o en una alteración temporal la primera vía de datos abierta del program es compartida por los programas posteriores que abran el archivo al mismo tiempo. La posición del registro actual se guarda en la vía de datos abierta para todos los programas que utilicen el archivo. Si lee el registro de un programa y a continuación lee el registro de un programa al que llame, la lectura del registro que recupere estará en función de si la vía de datos abierta es compartida. Si la vía de datos abierta es compartida, la posición del registro actual en el programa que se llama viene determinada por la posición actual del programa que realiza la llamada. Si la vía de datos abierta no es compartida, cada programa tiene una posición independiente para el registro actual.

Si el programa retiene un bloqueo de registro en un archivo compartido y a continuación llama a un segundo programa que lee el archivo compartido para actualizarlo, puede liberar el bloqueo del primer programa realizando una operación **READ** en el archivo de actualización mediante el segundo programa o utilizando las operaciones **UNLCK** o lectura sin bloqueo.

Si comparte una vía de datos abierta mejorará el rendimiento ya que el sistema OS/400 no tiene que crear una nueva vía de datos abierta. Sin embargo, el compartir una vía de datos abierta puede causar problemas. Por ejemplo, se produce un error en los casos siguientes:

- Si un programa que comparte una vía de datos abierta intenta operaciones de archivo distintas de aquellas especificadas por la primera apertura (por ejemplo, intentar operaciones de entrada cuando en la primera apertura sólo se especificaron operaciones de salida).
- Si un programa que comparte una vía de datos abierta para un archivo descrito externamente intenta utilizar un formato de registro que el primer programa ha ignorado.
- Si un programa que comparte una vía de datos abierta para un archivo descrito en el programa especifica una longitud de registro que excede la longitud establecida por la primera apertura.

Cuando varios archivos de un programa en RPG/400 se alteran temporalmente para un archivo compartido en tiempo de ejecución, el orden de apertura del archivo es importante. Para controlar el orden de apertura del archivo, se debe utilizar una apertura controlada por el programador o utilizar un programa en CL para abrir los archivos antes de llamar al programa en RPG/400.

Si un programa comparte la vía de datos abierta de un archivo primario o secundario, el programa tiene que ejecutar los cálculos de detalle del registro que se procesa antes de llamar a otro programa que comparta la vía datos abierta. De otra forma, si se utiliza la consulta anticipada o si la llamada es en tiempo de totales, el compartir la vía de datos abierta para un archivo primario o secundario puede causar que el programa llamado lea datos de un registro equivocado del archivo.

Debe asegurarse de que cuando se abre por primera vez el archivo compartido en un trabajo, se especifican todas las opciones de apertura requeridas para las aperturas posteriores del archivo. Si las opciones de apertura especificadas para las aperturas posteriores de un archivo compartido no están incluidas en las especificadas para la primera apertura de un archivo compartido, se envía un mensaje de error al programa.

La Tabla 3 muestra en detalle las opciones de apertura del sistema para cada una de las opciones que puede especificar.

Tabla 3. Opciones de apertura del sistema permitidas con las opciones de apertura del usuario	
Opciones de apertura de usuario de RPG	Opciones de apertura del sistema
INPUT	INPUT
OUTPUT	OUTPUT (archivo creado por programa)
UPDATE	INPUT, UPDATE
DELETE	DELETE

Guía del usuario RPG/400
Compartimiento de una vía de datos abierta

ADD	OUTPUT (archivo existente)
------------	-----------------------------------

Para obtener información adicional sobre cómo compartir una vía de datos abierta, consulte la publicación *Guía para la Base de Datos*.

5.9 Utilización del mandato de lenguaje de control RCLSRC

El mandato de CL Reclamar Recursos (RCLSRC) está diseñado para utilizarlo en el programa de control de una aplicación. Libera el almacenamiento estático y cierra los archivos que hayan dejado abiertos otros programas de la aplicación que ya no están activos. Este mandato no libera siempre el almacenamiento estático del programa o cierra todos los archivos. Mediante la utilización de RCLSRC puede cerrar algunos archivos pero retener su almacenamiento estático. Cuando sucede esto, el almacenamiento estático indica que estos archivos están abiertos pero que su vía de datos abierta (OPD) no existe. Se producirá un error cuando se intente una operación de E/S con estos archivos. Para obtener información adicional, consulte el manual *CL Reference*.

5.10 Especificaciones para archivos descritos externamente

Puede utilizar las DDS para describir archivos al sistema OS/400. Cada tipo de registro en un archivo descrito externamente se identifica mediante un nombre de formato de registro exclusivo.

El siguiente texto describe las entrada especiales que puede utilizar en las especificaciones de descripción de archivo, de entrada y de salida para archivos descritos externamente. Recuerde que las especificaciones de entrada y de salida para archivos descritos externamente son optativas.

Subtemas

- 5.10.1 Especificaciones de descripción de archivo
- 5.10.2 Cambio de nombre de formatos de registro
- 5.10.3 Ignorado de formatos de registro
- 5.10.4 Campos de coma flotante
- 5.10.5 Alteración temporal o adición de funciones del RPG/400 a una descripción externa
- 5.10.6 Especificaciones de salida

5.10.1 Especificaciones de descripción de archivo

Una **E** en la posición 19 de las especificaciones de descripción de archivos identifica a un archivo descrito externamente. La entrada **E** indica al compilador que tiene que recuperar la descripción externa del archivo del sistema al compilar el programa.

La información de esta descripción externa incluye:

- Información sobre el archivo, como por ejemplo el tipo de archivo y los atributos del archivo tales como el método de acceso (por clave o por número relativo de registro)
- Descripción del formato de registro, que incluye el nombre del formato de registro y descripciones de campo (nombre, posiciones y atributos).

La información que el compilador del RPG/400 recupera desde la descripción externa se imprime en el listado del compilador cuando se compila el programa.

5.10.2 Cambio de nombre de formatos de registro

Muchas de las funciones que puede especificar para archivos descritos externamente (como por ejemplo la operación **CHAIN**) operan o en un nombre de archivo o un nombre de formato de registro. Por lo tanto, cada nombre de archivo y de formato de registro de un programa debe ser un nombre simbólico exclusivo.

Para cambiar un nombre de formato de registro, utilice la opción **RENAME** de la línea de continuación de especificaciones de descripción de archivos del archivo descrito externamente como aparece en la Figura 36. No puede especificar la opción **RENAME** en la línea principal de especificaciones de descripción de archivos. La opción **RENAME** suele utilizarse si el programa contiene dos nombre de formato de registro idénticos o si el nombre de formato de registro supera los ocho caracteres, que es la longitud máxima permitida en el programa RPG/400.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF...LregLCIAIDePcLaEDispos+.....KSalidaEntradA...U1.*
FMAERTL IP E K DISK
F FORMATARTI KRENAMEARTMAE
F*
```

Figura 36. Opción **RENAME** para nombres de formato de registro en un archivo descrito externamente

Para cambiar el nombre a un formato de registro en un archivo descrito externamente, tiene que utilizar una línea de continuación de especificaciones de descripción de archivos para especificar la opción **RENAME**. (La opción **RENAME** no puede especificarse en la línea principal de descripción de archivos debido a que las posiciones de nombre externo solapan algunas de las entradas de la línea principal de descripción de archivos). Entre el nombre externo del formato de registro, ajustado por la izquierda, en las posiciones 19 a 28 de la línea de continuación. Especifique una **K** en la posición 53, **RENAME** en las posiciones 54 a 59 y el nombre del programa para el formato de registro, ajustado por la izquierda, en las posiciones 60 a 67. Las restantes posiciones de la línea de continuación tienen que estar en blanco.

En este ejemplo, el nombre formato de registro **FORMATARTI** del archivo descrito externamente **MAERTL** pasa **ARTMAE** para su utilización en este programa.

5.10.3 Ignorado de formatos de registro

Si un formato de registro de un archivo descrito externamente no tiene que utilizarse en un programa, puede utilizar la opción **IGNORE** para hacer que el programa se ejecute como si el formato de registro no existiese en el archivo. Para archivos lógicos, esto quiere decir que todos los datos asociados con este formato serán inaccesibles desde el programa. Utilice la opción **IGNORE** en una línea de continuación de especificaciones de descripción de archivos para el archivo descrito externamente tal como se muestra en la Figura 37.

El archivo debe tener más de un formato de registro y no pueden ignorarse todos ellos, por lo menos uno tiene que permanecer. Con excepción de este requisito, puede ignorarse cualquier número de formatos de registro en un archivo.

Si se especifica el nombre de un formato de registro en una línea de continuación para la opción **IGNORE**, no puede especificarse en una línea de continuación para ninguna otra opción (**SFILE**, **RENAME**, o **PLIST**), o en una línea de continuación para otra opción **IGNORE**.

Los nombres de formatos de registro ignorados aparecen en el listado de referencia cruzadas, pero se indica que están ignorados.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF...LregLCIAIDePcLaEDispos+.....KSalidaEntrada...U1.*
FMAEARTL UF E K DISK
F NOUSADO K IGNORE
F*
```

Figura 37. Opción **IGNORE** para formatos de registro en un archivo descrito externamente

Para ignorar un formato de registro de un archivo descrito externamente, utilice una línea de continuación de especificaciones de descripción del archivo para especificar la opción **IGNORE**. (La opción **IGNORE** no puede especificarse en la línea principal de la descripción de archivos ya que las posiciones del nombre externo solapan algunas de las entradas de la línea principal de descripción de archivos). En la línea de continuación, entre el nombre externo del formato de registro, ajustado por la izquierda, en las posiciones 19 a 28, una **K** en la posición 53 e **IGNORE** en las posiciones 54 a 59. Las restantes posiciones de la línea de continuación tienen que estar en blanco.

En este ejemplo, se ignora el formato de registro **NOUSADO** del campo descrito externamente **MAEARTL**.

5.10.4 Campos de coma flotante

El programa en RPG/400 no da soporte a la utilización de campos de coma flotante. Si procesa un archivo descrito externamente , con campos de coma flotante, sucede lo siguiente:

- No puede acceder a los campos de coma flotante.
- Al crear un nuevo registro, los campos de coma flotante en el registro tienen el valor cero.
- Al actualizar los registros ya existentes, no se modifican los campos de coma flotante.
- Si intenta utilizar un campo de coma flotante como campo de clave, el programa recibe un error en tiempo de compilación.

5.10.5 Alteración temporal o adición de funciones del RPG/400 a una descripción externa

Puede utilizar las especificaciones de entrada para alterar temporalmente determinada información de la descripción externa de un archivo de entrada o para añadir funciones del RPG/400 a la descripción externa. En las especificaciones de entrada, puede:

- Asignar indicadores de identificación de registro a los formatos de registro tal como se muestra en la Figura 38.
- Asignar un nuevo nombre a un campo, tal como se muestra en la Figura 38.
- Asignar indicadores de nivel de control a los campos tal como se muestra en la figura Figura 38.
- Asignar valores de campo de comparación a campos para el proceso de registros coincidentes tal como se muestra en la Figura 39.
- Asignar indicadores de campo tal como se muestra en la Figura 39.

No puede utilizar las especificaciones de entrada para alterar temporalmente las ubicaciones de campo de un archivo descrito externamente. Los campos de un archivo descrito externamente se colocan en los registros en el orden en el que aparecen listados en las especificaciones de descripción de datos. Además, funciones dependientes de dispositivo como por ejemplo el control de formulario no son válidas en un programa en RPG/400 para archivos descritos externamente.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
INomreg++...In.....*
MAESTART 01 1
I.....Campoexter.....Campo+L1C1..PoNeCe...*
I          NUMARTIC 2          ARTI L1 3
I*
MAESTALM 02
I          NUMARTI          ARTI L1
I*
I*
```

Figura 38. Alteración temporal y adición de funciones del RPG/400 a una descripción externa

- 1 Para asignar un indicador de identificación de registro en un registro de un archivo descrito externamente, especifique el nombre del formato de registro en las posiciones 7 a 14 de las especificaciones de entrada y asigne un indicador válido de identificación de registro en las posiciones 19 a 20. Una utilización típica de las especificaciones de entrada con archivos descritos externamente es la de asignar indicadores de identificación de registros.

En este ejemplo, se asigna el indicador de identificación **01** al registro **MAESTART** y el indicador **02** al registro **MAESTALM**.

- 2 Para asignar un nuevo nombre a un campo en un registro descrito externamente, tiene que especificar el nombre externo del campo, ajustado por la izquierda, en las posiciones 21 a 30 de la línea de descripción del campo. En las posiciones 53 a 58, especifique el nombre que va a utilizarse en el programa.

En este ejemplo, se asigna el nuevo nombre **ARTI** al campo **NUMARTIC** en ambos registros para este programa, debido a que **NUMARTIC** excede la longitud máxima de 6 caracteres permitida para el nombre de un campo en **RPG/400**.

- 3 Para asignar un indicador de nivel de control a un campo en un registro descrito externamente, tiene que especificar el nombre del campo en las posiciones 53 a 58 y un indicador de nivel de control en las posiciones 59 y 60.

En este ejemplo, el campo **ARTI** de ambos registros **MAESTART** y **MAESTALM** se especifica para ser el campo de control **L1**.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
IREGMAE 01 1
I.....PDesdeA+++DNomcamL1M1FrP1NeCe...*
I          NUMCLI M1 1
```

```

I*
IREGSM      02
I
I           NUMCLI  M1
I           SALDEU      98  2
I*
    
```

Figura 39. Adición de funciones del RPG/400 a una descripción externa

- 1 Para asignar un valor de comparación a un campo en un registro descrito externamente, tiene que especificar el nombre del formato de registro en las posiciones 7 a 14 de la línea de identificación del registro. En la línea de descripción del campo, especifique el nombre del campo en las posiciones 53 a 58 y asigne un valor de nivel de comparación en las posiciones 61 y 62.

En este ejemplo, se ha asignado el valor de nivel de comparación **M1** al campo NUMCLI de ambos registros, REGMAE y REGSM.

- 2 Para asignar un indicador de campo a un campo en un registro descrito externamente, tiene que especificar el nombre del formato de registro en las posiciones 7 a 14 de la línea de identificación del registro. En la línea de descripción del campo, tiene que especificar el nombre del campo en las posiciones 53 a 58 y un indicador en las posiciones 65 a 70.

En este ejemplo, se prueba si el campo SALDEU, en el registro REGSM, es cero cuando se lee e introduce en el programa. Si el valor del campo es cero, se activa el indicador 98.

5.10.6 Especificaciones de salida

Las especificaciones de salida son optativas para los archivos descritos externamente. El programa en RPG/400 da soporte a códigos de operación tales como **WRITE** y **UPDAT**, que utilizan descripciones de registro externas para describir el registro de salida sin necesidad de las especificaciones de salida para el archivo descrito externamente.

Puede utilizar las especificaciones de salida para controlar cuándo deben grabarse los datos o para especificar los campos seleccionados que debe grabarse. Las entradas válidas para la línea de descripción de archivo de un archivo descrito externamente son los indicadores de salida (posiciones 23 a 31), el nombre de campo (posiciones 32 a 37), un espacio en blanco final (posición 39). Las palabras y los códigos de edición para los campos grabados en un archivo descrito externamente están especificados en las DDS del archivo. Las funciones dependientes de dispositivo como por ejemplo la búsqueda de desbordamiento (posición 16) o espaciado/salto de impresión (posiciones 17-22) no son válidas en el programa RPG/400 para archivos descritos externamente. El indicador de desbordamiento tampoco es válido para archivos descritos externamente. Para obtener una descripción sobre cómo especificar la edición en las DDS, consulte la publicación *DDS Reference*.

Si se utilizan las especificaciones de salida para un archivo descrito externamente, en las posiciones 7 a 17 se especifica el nombre de formato de registro en lugar del nombre de archivo.

Si todos los campos del archivo descrito externamente deben colocarse en el registro de salida, entre ***ALL** en las posiciones 32 a 37 de la línea de descripción del campo. Si se especifica ***ALL**, no puede especificar otras líneas de descripción de campo para este registro.

Si desea colocar sólo ciertos campos en el registro de salida, entre el nombre del campo en las posiciones 32 a 37. Los nombres de los campos especificados en estas posiciones tienen que ser los nombres de los campos definidos en la descripción externa del registro, a menos que se haya asignado un nuevo nombre al campo en las especificaciones de entrada. Consulte la Figura 40.

Debe conocer estas consideraciones para utilizar las especificaciones de salida para un archivo descrito externamente:

- En la salida de un registro actualizado, sólo se sitúan en el registro de salida que debe volverse a grabar aquellos campos especificados en las especificaciones del campo de salida que cumplan las condiciones especificadas por los indicadores de salida. Los campos no especificados en las especificaciones de salida se vuelven a grabar utilizando los valores que se hayan leído. Esta técnica ofrece un buen método de control en contraposición al código de operación **UPDAT** que actualiza todos los campos.
- En la creación de un nuevo registro, se sitúan en el registro los campos especificados en las especificaciones de campos de salida. Aquellos campos no especificados en las especificaciones de campos de salida o que no cumplen las condiciones especificadas por los indicadores de salida, se graban como ceros o blancos dependiendo del formato de datos especificado en la descripción externa.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OREGART D          20
O.....N01N02N03Campo+YBUpoSPConstante/palabraedición++...*
O
O*                  *ALL 1
O*
O*
OREGVEN D         30
O
O                  NOMVEN 2
O                  TARCOM
O                  15 BONOS
O*
O*
```

Figura 40. Especificaciones de salida para un archivo descrito externamente

- 1 Para un archivo de actualización, todos los campos del registro se graban en el registro descrito externamente **REGART** utilizando los valores actuales del programa para todos los campos del registro.

Para la creación de un nuevo registro, se graban todos los campos del registro en el registro descrito externamente **REGART**, utilizando

los valores actuales del programa para todos los campos del registro.

- 2 Para actualizar un registro, se graban los campos NOVME y TARCUM en el registro descrito externamente REGVEN cuando está activado el indicador 30. El campo BONOS se graba en el registro REGVEN cuando están activados los indicadores 30 y 15. Todos los demás campos del registro se graban con los valores que se han leído.

Para crear un nuevo registro, se graban los campos NOVME y TARCUM en el registro descrito externamente REGVEN cuando está activado el indicador 30. El campo BONOS se graba cuando están activados los indicadores 30 y 15. Todos los demás campos del registro se graban como ceros o blancos, en función de si el campo es numérico o de caracteres.

5.11 Archivos descritos por programa

Los archivos descritos por programa son aquellos cuyos registros y campos están descritos en las especificaciones de entrada/salida del programa que utiliza el archivo.

La Figura 41 muestra cómo especificar la comprobación de secuencia cuando los datos de entrada tienen que contener exactamente un registro del primer tipo (01 en las posiciones 15 y 16) seguido, al menos, por un registro de otro tipo (02 a 04 en las posiciones 15 y 16) en cada grupo de registros que se lea. Cuando se utilizan las especificaciones que se muestran en la Figura 41 y se leen dos registros consecutivos del primer tipo, se produce un error en tiempo de ejecución.

Si cada grupo de registros de entrada debe contener exactamente un registro de un tipo en concreto, pero el registro no necesita ir seguido por registros de otros tipos, no especifique ninguna comprobación de secuencia (entrada alfabética en las posiciones 15 y 16).

Las operaciones de grabación en un archivo descrito por programa requieren un nombre de estructura de datos en el campo del resultado.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
IINPUT 011 10 1 CA
I.....PDesdeA+++DNomcamL1M1FrP1NeCe...*
I 2 60 TIPO
I 02NO20 1 CB 2 11 CLAVE
I 03NO30 1 CC 2 21 NOMBRE
I 04NO40 1 CD 2 6 NUMERO
I*
```

Figura 41. Especificaciones de entrada para comprobación de secuencia

5.12 Archivos de impresora

El archivo **PRINTER** le permite imprimir el archivo de salida. Se permiten ocho archivos de impresora por programa como máximo. Debe asignar **PRINTER** como el nombre de dispositivo del archivo, y cada archivo debe tener un nombre exclusivo. Puede utilizar el mandato de CL CRTPRTF (Crear archivo de impresión) para crear un archivo de impresora (consulte la publicación *CL Reference* para obtener más información sobre el mandato CRTPRTF); o utilice los nombre de archivo suministrados por IBM. Consulte la publicación *Guía para la Gestión de Datos* para obtener más información acerca de estos nombres de archivo.

Los códigos de operación que son válidos para un archivo **PRINTER** son **WRITE, OPEN, CLOSE** y **FEOD**. Para obtener una descripción completa de estos códigos de operación, consulte el manual *RPG/400 Reference*.

Los archivos **PRINTER** pueden estar descritos externamente o estar descritos por el programa. En un archivo **PRINTER** descrito externamente no se permiten los indicadores de desbordamiento **OA-OG** y **OV**, la obtención de desbordamiento, las entradas de espacio/salto y la opción **PRTCTL**. Consulte el manual *RPG/400 Reference* para obtener las entradas de especificación de salida válidas para archivos descritos externamente. Consulte la publicación *DDS Reference* para obtener información sobre las DDS para archivos de impresora descritos externamente.

Para un archivo **PRINTER** descrito externamente puede especificar la palabra clave de DDS **INDARA**. Si intenta utilizar esta palabra clave para un archivo **PRINTER** descrito por programa, obtendrá un error en tiempo de ejecución.

Subtemas

- 5.12.1 Desbordamiento de página
- 5.12.2 Indicadores de desbordamiento
- 5.12.3 Lógica de obtención de desbordamiento
- 5.12.4 Opción PRTCTL (Control de impresión)

5.12.1 Desbordamiento de página

Una consideración importante al utilizar un archivo **PRINTER** es el desbordamiento de página. Para un archivo **PRINTER** descrito externamente, el manejo del desbordamiento de página es responsabilidad del usuario. Efectúe una de las siguientes acciones:

- Cuento el número de líneas de salida por página.
- Compruebe la condición de excepción/error especificando un indicador en las posiciones 56 y 57 de las especificaciones de cálculo que especifican la operación de salida, o especificando una INFSD que pueda manejar el error. La INFSD contiene información detallada sobre la condición de excepción/error del archivo. Consulte "Manejo de excepciones/errores" en el tema 4.8 para obtener más información sobre cómo manejar excepciones y errores.
- Especifique un indicador del **01** al **99** como el indicador de desbordamiento en las posiciones 33 y 34 de las especificaciones de descripción de archivo.
- Compruebe la INFSD para obtener el número de línea y el desbordamiento de página. Consulte el manual *RPG/400 Reference* para obtener más información.

Para archivos descritos por programa o descritos externamente puede especificar un indicador del **01** al **99** en las posiciones 33 a 34 de las especificaciones de descripción de archivo. Este indicador se activa cuando se imprime una línea en la línea de desbordamiento o se alcanza o sobrepasa la línea de desbordamiento durante una operación de espaciado o salto de impresión. Utilice el indicador para condicionar su respuesta a la condición de desbordamiento. El indicador no condiciona la lógica de desbordamiento del RPG/400 como lo haría un indicador de desbordamiento (**OA** a **OG**, **OV**). La activación del indicador es responsabilidad del usuario.

Tanto los archivos descritos pro programa como los descritos externamente, el número de línea y el número de página está disponible en la INFSD del archivo. Para acceder a esta información, especifique una INFSD para el archivo utilizando una especificación de continuación de archivo. En la especificación, defina el número de línea en las posiciones 367-368 y defina el número de página en las posiciones 369-372 de la estructura de datos. Tanto el campo de número de línea como el de número de página debe estar definido en binario sin posiciones decimales. Como la INFSD se actualiza después de cada operación de salida hacia el archivo de impresora, estos campos se pueden utilizar para determinar la línea actual y el número de página sin que se necesite una lógica de contador de líneas del programa.

Para un archivo **PRINTER** descrito por programa, las siguientes secciones se aplican a los indicadores de desbordamiento y la lógica de desbordamiento de búsqueda.

5.12.2 Indicadores de desbordamiento

Un indicador de desbordamiento (**OA** a **OG**, **OV**) se activa cuando se imprime o sobrepasa la última línea de una página. Puede utilizarse un indicador de desbordamiento para especificar las líneas que han de imprimirse en la página siguiente. Los indicadores de desbordamiento sólo pueden especificarse para archivos **PRINTER** descritos por programa y se utilizan principalmente para condicionar la impresión de líneas de cabecera. Un indicador de desbordamiento se define en las posiciones 33 y 34 de las especificaciones de descripción de archivo y puede utilizarse para condicionar operaciones en las especificaciones de cálculo (posiciones 9 a 17) y en las especificaciones de salida (posiciones 23 a 31). Si no se especifica ningún indicador de desbordamiento, el compilador del RPG/400 asigna el primer indicador de desbordamiento que no se utilice al archivo **PRINTER**. Los indicadores de desbordamiento también pueden especificarse como indicadores de resultado en las especificaciones de cálculo (posiciones 54 a 59).

El compilador del RPG/400 activa un indicador de desbordamiento sólo la primera vez que se produce una condición de desbordamiento en una página. Se produce una condición de desbordamiento cuando ocurre uno de los casos siguientes:

- Se imprime una línea pasada la línea de desbordamiento.
- Se sobrepasa la línea de desbordamiento durante una operación de espaciado.
- Se sobrepasa la línea de desbordamiento durante una operación de espaciado.

La Tabla 4 muestra el resultado de la presencia o ausencia de un indicador de desbordamiento en las especificaciones de salida y descripción de archivo.

Las siguientes consideraciones son aplicables a indicadores de desbordamiento que se utilicen en las especificaciones de salida:

- El espaciado que sobrepase la línea de desbordamiento activa el indicador de desbordamiento.
- El salto que sobrepase la línea de desbordamiento activa el indicador de desbordamiento.
- El salto, sobrepasada la línea de desbordamiento, a cualquier línea de una nueva página activa el indicador de desbordamiento a menos que se especifique un salto a un punto posterior a la línea de desbordamiento especificada.
- Un salto a una página nueva especificada en una línea no condicionada por un indicador de desbordamiento desactiva el indicador de desbordamiento después de que el formulario pase a la nueva página.
- Si especifica un salto a una nueva línea y la impresora se encuentra actualmente en esta línea, no se produce un salto. El indicador de desbordamiento se desactiva a menos que la línea sea posterior a la línea de desbordamiento.
- Cuando se especifica una línea **OR** para un registro de impresión de salida, se utilizan las entradas de espaciado y salto de la línea anterior. Si éstas difieren de la línea anterior, entre entradas de espaciado y salto en la línea **OR**.
- Los indicadores de nivel de control pueden utilizarse con un indicador de desbordamiento para que cada página contenga información sobre un grupo de control solamente. Consulte la Figura 42.
- Para condicionar una línea de desbordamiento, un indicador de desbordamiento puede aparecer en una relación **AND** o **OR**. Para una relación **AND**, el indicador de desbordamiento puede aparecer en la línea principal de especificación para que esta línea se considere una línea de desbordamiento. Para una relación **OR**, el indicador de desbordamiento puede especificarse en la línea principal de especificación o en la línea **OR**. Sólo puede asociarse un indicador de desbordamiento a un grupo de indicadores de salida. Para una relación **OR** sólo se utilizan para el condicionamiento de la línea de desbordamiento los indicadores de condicionamiento de la línea de especificación donde se ha especificado un indicador de desbordamiento.
- Si se utiliza un indicador de desbordamiento en una línea **AND**, la línea no es una línea de desbordamiento. En este caso, el indicador de desbordamiento se trata como cualquier otro indicador de salida.
- Cuando se utiliza el indicador de desbordamiento en una relación **AND**

con un indicador de identificación de registro, se obtienen frecuentemente resultados no usuales puesto que puede que el tipo de registro no se haya leído al ocurrir el desbordamiento. Por ello, no se activa el indicador de identificación de registro y no se imprimen todas las líneas condicionadas por el indicador de desbordamiento y el de identificación de registro.

- Un indicador de desbordamiento condiciona una línea de excepción (E en la posición 15), y condiciona los campos dentro del registro de excepción.

Tabla 4. Resultado de la presencia o ausencia de un indicador de desbordamiento		
Archivo Descripción Específic. Posiciones 33-34	Salida Específic. Posiciones 23-31	Acción
Sin entrada	Sin entrada	Se utiliza el primer indicador de desbordamiento no utilizado para condicionar el salto a la página siguiente al producirse el desbordamiento
Sin entrada	Entrada	Error en tiempo de compilación; se elimina el indicador de desbordamiento de las especificaciones de salida. Se utiliza el primer indicador de desbordamiento no utilizado para condicionar el salto a la página siguiente al producirse el desbordamiento.
Entrada	Sin entrada	Impresión continua; no se detecta el desbordamiento
Entrada	Entrada	Ejecuta el desbordamiento normal

La primera parte de la siguiente figura es un ejemplo de la codificación necesaria para imprimir cabeceras en cada página: primera página, cada página con desbordamiento, y cada nueva página que se empiece debido a un cambio en los campos de control (se activa **L2**). La primera línea permite que se impriman las cabeceras al principio de una nueva página (salta a 06) sólo cuando se produce un error de desbordamiento (se activa **OA** y no se activa **L2**).

La segunda línea permite la impresión de cabeceras en la nueva página sólo al principio de un nuevo grupo de control (se activa **L2**). De esta forma, no se producen las cabeceras duplicadas generadas tanto por **L2** como por **OA** que estén activadas al mismo tiempo. La segunda línea permite que se impriman las cabeceras en la primera página después de que se lea el primer registro, ya que el primer registro siempre provoca una interrupción del control (se activa **L2**) si en el registro hay especificados campos de control.

La segunda parte de la figura es la codificación necesaria para imprimir ciertos campos en cada página; se produce un salto a 06 cuando ocurre una condición de desbordamiento o un cambio del nivel de control (**L2**). El indicador **NL2** evita que la línea imprima y salte dos veces en el mismo ciclo.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OIMPR   H   306   OANL2
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O       OR       L2
O
O              8 'FECHA'
O              18 'CUENTA'
O              28 'NOMBRE'
O              46 'SALDO'
O*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OIMPR   D   306   OANL2
```

O	OR	L2	
O.....			N01N02N03Campo+YBUposPConstante/palabraedición++...*
O		CTA	8
O*			

Figura 42. Utilización de indicadores de nivel de control con un indicador de desbordamiento

5.12.3 Lógica de obtención de desbordamiento

Cuando no hay suficiente espacio en una página para imprimir el resto de las líneas de detalle, totales, excepción y cabecera condicionadas por el indicador de desbordamiento, puede llamarse a la rutina de obtención de desbordamiento. Esta rutina provoca un desbordamiento. Para determinar cuándo obtener la rutina de desbordamiento, estudie todas las situaciones de desbordamiento posibles. Al contar las líneas y los espacios puede calcular lo que sucede si se produce un desbordamiento en cada línea de detalle, totales y excepción.

La rutina de obtención de desbordamiento le permite alterar la lógica básica de desbordamiento del RPG/400 para evitar que se imprima sobre agujeros y le deja utilizar todo el espacio de la página que desee. Durante el ciclo regular del programa, el compilador del RPG/400 sólo comprueba una vez, inmediatamente después de la salida de totales, si está activado el indicador de desbordamiento. Cuando se especifica la función de obtención de desbordamiento, el compilador del RPG/400 comprueba en cada línea en la que se haya especificado la obtención de desbordamiento si existe desbordamiento. Consulte la Figura 43.

Especifique la obtención de desbordamiento con una **F** en la posición 16 de las especificaciones de salida de cualquier línea de detalle, totales o excepción de un archivo **PRINTER**. La rutina de obtención de desbordamiento no hace que el formulario avance a la siguiente página de forma automática.

Durante la salida, se comprueban los indicadores de desbordamiento de una línea de salida para determinar si se ha de escribir la línea. Si la línea debe escribirse, y se especifica una **F** en la posición 16, el compilador del RPG/400 realiza pruebas para determinar si está activado el indicador de desbordamiento. Si el indicador de desbordamiento está activado, la se obtiene la rutina de desbordamiento y se producen las siguientes operaciones:

1. Sólo se comprueba la salida de las líneas de desbordamiento del archivo en que se haya especificado la obtención.
2. Se escriben todas las líneas de totales condicionadas por el indicador de desbordamiento.
3. El formulario avanza a una nueva página cuando se especifica un salto a un número de línea inferior al número de línea en el que se encuentra actualmente la impresora en una línea condicionada por un indicador de desbordamiento.
4. Se escriben las líneas de cabecera, detalle y excepción condicionadas por el indicador de desbordamiento.
5. Se escribe la línea que ha obtenido la rutina de desbordamiento.
6. Se escriben las líneas de detalles o de totales que queden por escribirse en el ciclo del programa.

La posición 16 de cada línea **OR** debe contener una **OR** si va a utilizarse la rutina de desbordamiento para cada registro de la relación **OR**. No puede utilizarse la obtención de desbordamiento si se especifica un indicador de desbordamiento en las posiciones 23 a 31 de la misma línea de especificación. Si ocurre esto, no se obtiene la rutina de desbordamiento.

La Figura 44 muestra la utilización de la obtención de desbordamiento.

IMAGEN 17

Figura 43. Impresión de desbordamiento: Valor del indicador de desbordamiento

- A Cuando no se especifica la obtención del desbordamiento, las líneas de desbordamiento se imprimen después de la salida de totales. Con independencia de cuando se produzca el desbordamiento (**OA** está activo), el indicador de desbordamiento **OA** permanecerá activado durante el tiempo de salida de desbordamiento y se desactivará después del tiempo de salida de cabecera y detalle.
- B Cuando se especifica la obtención de desbordamiento, las líneas de desbordamiento se imprimen antes de la línea de salida para la que se ha especificado obtención de desbordamiento, siempre que el indicador de desbordamiento **OA** esté activado. Cuando se activa **OA**, permanece en dicho estado hasta después del tiempo de salida de cabecera y detalle. Las líneas de desbordamiento no se escriben por

segunda vez durante el tiempo de salida de desbordamiento, después de la última vez que se escribieron las líneas de desbordamiento.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OIMPRESO H 305 OA
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O
O          TF 1      L1
O
O          T 1      L1      TOTEMPL 25
O
O          T 1      L1      TOTEMPL 35
O
O          T 1      L1      TOTEMPL 45
O
O          TF 1      L1      TOTEMPL 55
O
O          T 1      L1      TOTEMPL 65
O
O          T 1      L1      TOTEMPL 75
O
O*

```

Figura 44. Utilización de la obtención de desbordamiento

Las líneas de totales con una **F** codificada en la posición 16 puede obtener la rutina de desbordamiento. Sólo lo hacen si se detecta una rutina de desbordamiento con anterioridad a la impresión de una de estas líneas. Antes de procesarse la obtención de desbordamiento, se efectúa una comprobación para determinar si está activado el indicador de desbordamiento. Si está activado, se obtiene la rutina de desbordamiento, se imprime la línea de cabecera condicionada por el indicador de desbordamiento y se procesan las operaciones de totales.

5.12.4 Opción PRTCTL (Control de impresión)

La opción **PRTCTL** (control de impresora) le permite modificar la información sobre el control de formularios y acceder al valor de la línea actual en el programa para un archivo **PRINTER** descrito por el programa.

Especifique la opción **PRTCTL** en una línea de continuación de las especificaciones de descripción de archivos para un archivo **PRINTER** con lo siguiente:

Nota: Si el archivo tiene una ODP compartida o una apertura controlada por el usuario, el valor de la cuenta de líneas puede ser incorrecto.

Posición	Entrada
6	F
7-52	En blanco si la información se especifica en una línea de continuación independiente.
53	K (indica una línea de continuación)
54-59	PRTCTL
60-65	Nombre de la estructura de datos utilizada para contener la información de control de la impresora y del contador de líneas
66-74	En blanco si la información se especifica en una línea de continuación independiente.

La estructura de datos especificada en las posiciones 60 a 65 de la línea de continuación de las especificaciones de descripción de archivos tiene que especificarse en las especificaciones de entrada y debe contener, por lo menos, los siguientes cinco subcampos, especificados en el orden siguiente:

Posiciones de la estructura de los datos	Contenido de los subcampos
1	Un campo de caracteres de una posición que contiene el valor de espaciar antes
2	Un campo de caracteres de una posición que contiene el valor de espaciar después
3-4	Un campo de caracteres de dos posiciones que contiene el valor de saltar antes
5-6	Un campo de caracteres de 2 posiciones que contiene el valor de saltar después
7-9	Un campo numérico de tres dígitos sin posiciones decimales que contiene el valor de la cuenta de líneas actual.

Los valores contenidos en los cuatro primeros subcampos de la estructura de datos son los mismos que los permitidos en las posiciones 17 a 22 (entradas de espaciado y salto) de las especificaciones de salida. Si las entradas de espaciado/salto (posiciones 17 a 22) de las especificaciones de salida están en blanco, y los subcampos 1 a 4 también están en blanco, el valor por omisión es espaciar 1 después. Si se especifica la opción **PRTCTL**, únicamente se utiliza para los registro de salida que tengan blancos en las posiciones 17 a 22. Puede controlar el valor de espaciado y de salto (subcampos 1 a 4) para el archivo **PRINTER** cambiando los valores en estos subcampos mientras el programa está ejecutando. Consulte la Figura 45.

El subcampo 5 contiene el valor de la cuenta de líneas actual. El compilador del RPG/400 no inicializa el subcampo 5 hasta después de que se imprima la primera línea de salida. El compilador del RPG/400 cambia entonces el subcampo 5 después de cada operación de salida al archivo.

La Figura 46 es un diagrama de proceso para archivos **PRINTER**.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF...LregLC1AIDePcLaEDispos+.....KSalidaEntrada...U1.*
FIMPRE  O  F  132  PRINTER  KPRTCTLLINEA
F*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
IDsnombre..NODsSal-arch++.....NvecLon+.....*
ILINEA      DS
I.....campoexter.....PDesdA+++DCampo+.....*
I              1  1 ESPANT
I              2  2 ESPDES
I              3  4 SALANT
I              5  6 SALDES
I              7  90LINACT
I*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C  01  LINACT  COMP 10  49
C  01 49  MOVE '3'  ESPDES
C*
C*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OIMPRE      01
O.....N01N02N03Campo+YBUposPConstante/palabraedición+++*
O              DATA  25
O*
```

Figura 45. Ejemplo de la opción PRTCTL

En las especificaciones de descripción de archivos se especifica la opción **PRTCTL** para el archivo **IMPRE**. El nombre de la estructura de datos asociada es **LINEA**.

La estructura de datos **LINEA** se define en las especificaciones de entrada con sólo aquellos subcampos predefinidos para la estructura de datos **PRTCTL**. Los cuatro primeros subcampos, posiciones 1 a 6, se utilizan para facilitar la información de espaciado y salto que normalmente se especifica en las columnas 17-22 de las especificaciones de salida. La opción **PRTCTL** permite al programador cambiar estas especificaciones dentro del programa.

En este ejemplo el valor del subcampo **ESPDES** se cambia a 3 cuando el valor del subcampo **LINACT** (valor de cuenta de líneas actual) es igual a 10. (Suponga que el indicador **01** se ha activado como un indicador de identificación de registros.)

IMAGEN 18

Figura 46. Diagrama de proceso para archivos PRINTER

Operaciones de archivo válidas:

1. **WRITE, OPEN, CLOSE, FEOD**

Nota: Las posiciones sombreadas deben estar en blanco. Las posiciones sin entradas son dependientes del programa.

5.13 Archivo secuencial

Una especificación de dispositivo secuencial (SEQ), en las posiciones 40 a 46 de la especificación de descripción de archivos, indica que la entrada o salida está asociada a un archivo organizado secuencialmente. Consulte la Figura 47. El dispositivo actual que ha de asociarse con el archivo mientras se ejecute el programa en RPG/400 puede especificarse mediante un mandato de alteración temporal de OS/400 (consulte el manual *CL Reference* para obtener más información), o mediante la descripción de archivos a la que señala el nombre del archivo.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF...LregLCIAIDePcLaEDispos+.....KSalidaEntradA....U1.*
FTARITPO IP E DISK
FHORAEXTRO F 132 SEQ
F*
F*
```

Figura 47. Dispositivo SEQ

Se ha especificado un dispositivo SEQ para el archivo **HORAEXTR**. Cuando se ejecute el programa, puede utilizar un mandato de alteración temporal de OS/400 para especificar el dispositivo actual (como por ejemplo una impresora, cinta o disquete) que se asociará al archivo mientras se ejecute el programa. Por ejemplo, se puede especificar disquete para algunas ejecuciones del programa e impresora para otras. La descripción de archivo, a la que señala el nombre del archivo, puede especificar el dispositivo actual, en cuyo caso no es necesario utilizar un mandato de alteración temporal.

Cualquier archivo organizado secuencialmente, como por ejemplo un disquete, cinta, base de datos, archivo para salvar o impresora, puede estar asociado a un dispositivo SEQ. Si se especifica SEQ en un programa en RPG/400, no puede especificarse ninguna función dependiente de dispositivo, como por ejemplo espaciar/saltar o **CHAIN**.

Utilice las especificaciones de dispositivo SEQ cuando grabe en un archivo de cinta. Para grabar registros de longitud variable en un archivo de cinta, utilice también el parámetro RCDBLKFMFMT del mandato de CL CRTAPF o OVRTAPF. (Consulte la publicación *CL Reference*.) Cuando utilice el parámetro RCDBLKFMFMT, la longitud de cada registro que graba el programa en el archivo de cinta viene determinada por la posición de fin de archivo más alta especificada en las posiciones 40 a 43 de las especificaciones de salida de este registro. Si no especifica ninguna posición de fin, el compilador del RPG/400 calcula la longitud del registro a partir de la longitud de los campos del registro.

Lea los registros de cinta de longitud variable como si fueran registros de cualquier archivo organizado secuencialmente. Asegúrese de que la longitud especificada en la especificación de descripción de archivo de cabida al registro más largo del archivo.

La siguiente figura muestra los códigos de operación permitidos para un archivo SEQ.

Especificaciones de descripción de archivos		Especificaciones de cálculo
Posiciones 15	Posiciones 16	Posiciones 28-32
I	P/S	CLOSE, FEOD
I	F	READ, OPEN, CLOSE, FEOD
O		WRITE, OPEN, CLOSE, FEOD

Nota: Para un archivo secuencial no están permitidas las especificaciones de control de impresión.

Figura 48. Códigos de operación de archivo válidos para un archivo secuencial

La Figura 49 es un diagrama de proceso para archivo SEQ.

Figura 49. Diagrama de proceso para archivos SEQ

Operaciones de archivo válidas:

1. **CLOSE, FEOD**
2. **READ, OPEN, CLOSE, FEOD**
3. **OPEN, CLOSE, FEOD**
4. **WRITE, OPEN, CLOSE, FEOD**

Nota: Las posiciones sombreadas deben estar en blanco. Las posiciones sin entradas son dependientes del programa.

5.14 Archivo especial

SPECIAL en las posiciones 40 a 46 de las especificaciones de descripción de archivos, le permite especificar un dispositivo de entrada y/o salida no soportado directamente por el lenguaje RPG/400. Las operaciones de entrada y salida para el archivo están controladas mediante una rutina escrita por el usuario. Las posiciones 54 a 59 de la línea de las especificaciones de descripción de archivos que contiene **SPECIAL** en las posiciones 40 a 46, han de contener el nombre de la rutina escrita por el usuario.

El RPG/400 llama a esta rutina escrita por el usuario para abrir el archivo, leer y grabar registros y cerrar el archivo. El RPG/400 crea una lista de parámetros que va a utilizar la rutina escrita por el usuario. La lista de parámetros contiene un parámetro de código de operación (opción), un parámetro de estado de retorno (estado), un parámetro de error encontrado (error) y un parámetro de área de registro (área). El programa en RPG/400 y la rutina escrita por el usuario acceden a esta lista de parámetros; el programa en RPG/400 que contiene el archivo **SPECIAL** no puede acceder a esta lista de parámetros.

A continuación se describen los parámetros incluidos en esta lista de parámetros creada por el RPG:

- Opción: El parámetro opción es un campo de caracteres de una posición que indica la acción que va a procesar la rutina escrita por el usuario. Según la operación que se está procesando en el archivo **SPECIAL (OPEN, CLOSE, READ, WRITE, DELET, UPDAT)**, desde el RPG/400 se pasa uno de los siguientes valores a la rutina escrita por el usuario:

Valor pasado	Descripción
O	Abrir el archivo.
C	Cerrar el archivo.
R	Read a record and place it in the area defined by the area parameter.
W	El programa en RPG/400 he colocado un registro en el área definida por el parámetro de área; el registro se grabará.
D	Suprimir el registro.
U	El registro es una actualización del último registro leído.

- Estado: El parámetro de estado es un campo de caracteres de una posición que indica el estado de la rutina escrita por el usuario cuando se devuelve el control al programa en RPG/400. El estado debe contener uno de los siguientes valores de retorno cuando la rutina escrita por el usuario devuelve el control al programa en RPG/400:

Valor devuelto	Descripción
0	Retorno normal. Se ha procesado la acción solicitada.
1	El archivo de entrada está al final del archivo y no se ha devuelto ningún registro. Si el archivo es de salida, este valor de retorno es un error.
2	No se ha procesado la acción solicitada; existe una condición de error.

- Error: El parámetro de error es un campo numérico con zona de cinco dígitos con cero posiciones decimales. Si la rutina escrita por el usuario detecta un error, el parámetro de error contiene una indicación o valor que representa el tipo de error. El valor se sitúa en las primeras cinco posiciones de la posición ***RECORD** en la INFDS cuando el parámetro de estado contiene 2.
- Area: El parámetro de área es un campo de caracteres cuya longitud es igual a la longitud del registro asociado al archivo **SPECIAL**. Este campo se utiliza para pasar el registro al o recibir el registro desde el programa en RPG/400.

Puede añadir parámetros adicionales a la lista de parámetros creada por RPG. Especifique **PLIST** en las posiciones 54 a 59 y el nombre de la **PLIST** en las posiciones 60 a 65 de las especificaciones de una línea de continuación de la descripción de archivo para el archivo SPECIAL. Consulte la Figura 50. Después utilice la operación **PLIST** en las especificaciones de cálculo para definir los parámetros adicionales.

La rutina escrita por el usuario, el nombre que está especificado en las posiciones 54 a 59 de las especificaciones de descripción de archivo para el archivo SPECIAL, debe contener una lista de parámetros de entrada que incluya los parámetros creados por el RPG y los parámetros especificados por el usuario.

Si se especifica el archivo SPECIAL como un archivo primario, los parámetros especificados por el usuario deben inicializarse antes de leer el primer archivo primario. Puede inicializar estos parámetros con una entrada de factor 2 en las sentencias **PARM** o especificando como parámetro una matriz en tiempo de compilación o un elemento de matriz.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF....LregLCIAIDePcLaEDispos+.....KSalidaEntrada....U1.*
FEXCEPT 1 F SPECIAL ENSAUS
F KPLIST ESPE
F*
F*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C ESPE PLIST
C PARM FLD1
C PARM FLD2
C PARM FLD3
C*
C*
```

Figura 50. Dispositivo SPECIAL

Se asigna el archivo **EXCEPT** al dispositivo SPECIAL. Las operaciones de E/S para el dispositivo SPECIAL las controla la rutina escrita por el usuario **ENSAUS**. Los parámetros especificados para **PLIST ESPE**, definidos por el usuario, se añaden al final de la lista de parámetros creada por el RPG para el dispositivo SPECIAL. Se puede acceder a los parámetros especificados por el programador mediante el programa en RPG/400 del usuario y la rutina escrita por el usuario; sin embargo, sólo se puede acceder a la lista de parámetros creada por el RPG mediante la lógica interna del RPG/400 y la rutina escrita por el usuario.

La Figura 51 muestra los códigos de operación que son válidos para un archivo SPECIAL.

Especificaciones de descripción de archivos			Especificaciones de cálculo
Posiciones			Posiciones
15	16		28-32
I	P/S		CLOSE, FEOD
C	P/S		WRITE, CLOSE, FEOD
U	P/S		UPDAT, DELET, CLOSE FEOD
O			WRITE, OPEN, CLOSE, FEOD
I	F		READ, WRITE, OPEN, CLOSE, FEOD
C	F		READ, WRITE, OPEN, CLOSE, FEOD
U	F		READ, UPDAT, DELET, OPEN, CLOSE, FEOD

Figura 51. Operaciones de archivo válidas para archivo SPECIAL

La Figura 52 es un diagrama de proceso para archivos SPECIAL.

IMAGEN 20

Figura 52. Diagrama de proceso para archivos SPECIAL

Operaciones de archivo válidas:

1. **CLOSE, FEOD**
2. **WRITE, CLOSE, FEOD**
3. **UPDAT, DELET, CLOSE, FEOD**
4. **READ, OPEN, CLOSE, FEOD**
5. **READ, WRITE, OPEN, CLOSE, FEOD**
6. **READ, DELET, UPDAT, OPEN, CLOSE, FEOD**
7. **WRITE, OPEN, CLOSE, FEOD**

Notas:

1. Las posiciones sombreadas deben estar en blanco. Las posiciones sin entradas son dependientes del programa.
2. Las posiciones 54 a 59 tienen que contener en nombre de la rutina escrita por el usuario que controla las operaciones de entrada/salida para el archivo.

6.0 Capítulo 6. Control de compromiso

Este capítulo describe cómo utilizar el control de compromiso para procesar las operaciones de archivos como un grupo. Mediante el control de compromiso se producirá uno de los dos resultados siguientes en las operaciones de archivo: o se ejecutarán con éxito todas las operaciones de archivo o ninguna de las operaciones de archivo tendrá efecto. De esta forma, puede procesar un grupo de operaciones como si se tratase de una unidad.

Subtemas

6.1 Utilización del control de compromiso

6.1 Utilización del control de compromiso

Efectúe los siguientes pasos para utilizar el control de compromiso:

- Utilice los mandatos de CL CRTJRN (Crear diario), CRTJRNRCV (Crear receptor de diario) y JRNPF (Registrar archivo físico por diario) para preparar la utilización del control de compromiso y los mandatos de CL STRCMTCTL (Arrancar control de compromiso) y ENDCMTCTL (Finalizar control de compromiso) para comunicar al sistema cuándo quiere arrancar y finalizar el control de compromiso. Consulte la publicación *CL Reference* para obtener más información sobre cómo utilizar estos mandatos.
- Especifique el control de compromiso en las especificaciones de descripción de archivo de los archivos que estarán bajo el control de compromiso.
- Utilice el código de operación **COMIT** (Comprometer) para aplicar un grupo de cambios a archivos bajo control de compromiso o utilice el código de operación **ROLBK** (Retrotraer) para eliminar el grupo de cambios pendientes en archivos bajo control de compromiso.

Subtemas

- 6.1.1 Comienzo y Fin del Control de Compromiso
- 6.1.2 Especificación de archivos para control de compromiso
- 6.1.3 Operaciones de control de compromiso
- 6.1.4 Bloqueo de control de compromiso
- 6.1.5 Control de compromiso en el ciclo del programa
- 6.1.6 Ejemplo de utilización del control de compromiso

6.1.1 Comienzo y Fin del Control de Compromiso

El mandato de CL STRCMTCTL comunica al sistema que se quieren procesar archivos bajo el control de compromiso.

El parámetro LCKLVL (Nivel de bloqueo) le permite seleccionar el nivel al que se bloquean los registros bajo el control de compromiso. Consulte "Bloqueo de control de compromiso" en el tema 6.1.4 y la publicación *CL Guía del Programador* para obtener más detalles sobre los niveles de bloqueo.

Cuando completa un grupo de modificaciones con una operación **COMIT** puede especificar una etiqueta que identifique el final del grupo. En caso de que el trabajo terminase de forma anormal, esta etiqueta identificativa se graba en un archivo, cola de mensajes o área de datos de manera que el usuario conozca qué grupo de modificaciones es el último grupo que se ha terminado satisfactoriamente. Especifique este archivo, cola de mensajes o área de datos en el mandato STRCMTCTL.

Antes de llamar a un programa que procese los archivos especificados para control de compromiso, emita el mandato STRCMTCTL. Si llama a un programa que abra un archivo especificado para control de compromiso antes de emitir el mandato STRCMTCTL, se producirá una anomalía al abrir el archivo.

El mandato de CL ENDCMTCTL comunica al sistema que el paso de direccionamiento ha finalizado para el proceso de archivos bajo control de compromiso. Consulte la publicación *CL Reference* para obtener más información sobre cómo utilizar los mandatos STRCMTCTL y ENDCMTCTL.

6.1.2 Especificación de archivos para control de compromiso

Entre una **K** en la posición 53 de las especificaciones de continuación de archivo y la palabra **COMIT** en las posiciones 54 a 59. En las especificaciones de descripción de archivo, indique que el archivo utiliza el dispositivo **DISK** en las posiciones 40 a 46.

Cuando un programa especifica control de compromiso para un archivo, la especificación sólo es aplicable a las operaciones de entrada y salida que este programa realice para este archivo. El control de compromiso no es aplicable a operaciones que no sean de entrada y salida. Tampoco es aplicable a archivos en los que el control de compromiso no esté especificado en el programa que efectúa la operación de entrada y salida.

Cuando más de un programa accede a un archivo como un archivo compartido, todos o ninguno de los programas deben especificar el archivo que estará bajo control de compromiso.

6.1.3 Operaciones de control de compromiso

La operación **COMIT** (Comprometer) indica al sistema que se ha completado un grupo de cambios en los archivos bajo el control de compromiso. La operación **ROLBK** (Retrotraer) elimina el grupo de cambios actual de los archivos bajo control de compromiso. Para información sobre cómo especificar estos códigos de operación y qué hace cada uno de los códigos de operación, consulte la publicación *RPG/400 Reference*.

Si se produce una anomalía en el sistema, éste emite implícitamente una operación **ROLBK**. Puede comprobar la identidad del último grupo de cambios completado con éxito utilizando la etiqueta que se especificó en el factor 1 del código de operación **COMIT** y el objeto de la comunicación que haya especificado en el mandato **STRCMTCTL**.

Al final de un paso de direccionamiento, o al emitir el mandato **ENDCMTCTL**, el sistema **RPG/400** emite una operación **ROLBK**, la cual elimina cualquier cambio posterior a la última operación **ROLBK** o **COMIT** que haya emitido. Para asegurarse de que todas las operaciones de archivo tengan efecto, emita una operación **COMIT** antes de finalizar un paso de direccionamiento que opere bajo el control de compromiso.

La operación **OPEN** permite realizar operaciones de entrada y salida sobre un archivo, y la operación **CLOSE** evita que puedan realizarse operaciones de entrada y salida en un archivo. Sin embargo, las operaciones **OPEN** y **CLOSE** no afectan a las operaciones **COMIT** y **ROLBK**. Una operación **COMIT** o **ROLBK** afecta a un archivo, incluso después de que éste se haya cerrado. Por ejemplo, el programa puede incluir los pasos siguientes:

1. Emitir **COMIT** (para archivos ya abiertos bajo el control de compromiso).
2. Abrir un archivo especificado para el control de compromiso.
3. Realizar algunas operaciones de entrada y salida en este archivo.
4. Cerrar el archivo.
5. Emitir **ROLBK**.

Los cambios realizados en el paso 3 se retrotraen mediante la operación **ROLBK** en el paso 5, aunque el archivo se haya cerrado en el paso 4. La operación **ROLBK** se podría emitir desde otro programa en el mismo paso de direccionamiento.

No es necesario que un programa opere todos los archivos bajo control de compromiso, y si lo hace el rendimiento puede verse afectado negativamente. Las operaciones **COMIT** y **ROLBK** no tienen ningún efecto sobre los archivos que no se encuentren bajo control de compromiso.

Nota: Cuando hay múltiples dispositivos conectados a un programa de aplicación, y el control de compromiso para los archivos que utiliza este programa está vigente, las operaciones **COMIT** o **ROLBK** continúan funcionando en base al archivo y no según el dispositivo. Puede que la base de datos se actualice con los bloques **COMIT** que se hayan completado parcialmente, o que se eliminen los cambios que hayan completado otros usuarios. Es responsabilidad del usuario asegurarse de que no ocurra esto.

6.1.4 Bloqueo de control de compromiso

En el mandato STRCMTCTL se especifica un nivel de bloqueo, ya sea **LCKLVL (*ALL)** o **LCKLVL (*CHG)**. Cuando el programa esté operando bajo el control de compromiso y haya procesado una operación de entrada o de salida en un registro de un archivo bajo el control de compromiso, el control de compromiso bloquea el registro de la siguiente forma:

- El programa puede acceder al registro.
- Otro programa del mismo paso de direccionamiento que tenga este archivo bajo el control de compromiso puede leer el registro. Si el archivo es un archivo compartido, el segundo programa también puede actualizar el registro.
- Otro programa del mismo paso de direccionamiento que no tenga este archivo bajo control de compromiso no puede leer ni actualizar el registro.
- Otro programa de un paso de direccionamiento distinto que tenga este archivo bajo el control de compromiso puede leer el registro si se ha especificado **LCKLVL (*CHG)**, pero no puede leerlo si se ha especificado **LCKLVL (*ALL)**. Con cualquiera de los niveles de bloqueo, el siguiente programa no puede actualizar el registro.
- Otro programa que no tenga este archivo bajo el control de compromiso y que no esté en el mismo paso de direccionamiento, puede leer pero no actualizar el registro.
- Los bloqueos del control de compromiso son diferentes de los bloqueos normales, dependen del **LCKLVL** especificado y sólo pueden liberarse mediante las operaciones **COMIT** y **ROLBK**.

Las operaciones **COMIT** y **ROLBK** liberan los bloqueos de los registros. La operación **UNLCK** no liberará los registros bloqueados utilizando el control de compromiso. Consulte la publicación *CL Reference* para obtener más detalles sobre los niveles de bloqueo.

Puede limitar el número de entradas que pueden bloquearse bajo el control de compromiso antes de que las operaciones **COMIT** o **ROLBK** sean necesarias. Para obtener más información, consulte el manual *Advanced Backup and Recovery Guide*

Nota: Las operaciones **SETLL** y **SETGT** bloquearán un registro en los mismos casos en los que una operación de lectura (no de actualización) hubiese bloqueado un registro para control de compromiso.

6.1.5 Control de compromiso en el ciclo del programa

El control de compromiso está pensado para archivos controlados en cálculo, en los que el usuario controla la entrada y la salida. No utilice control de compromiso con archivos primarios y secundarios, en los que el ciclo del programa en RPG/400 es el que controla la entrada y la salida. He aquí algunas de las razones para esta recomendación:

- No puede emitir una operación **COMIT** para la última salida de totales del programa.
- Es difícil la programación dentro del ciclo para la recuperación desde una condición de registro bloqueado.
- La operación **ROLBK** no restablece los indicadores de nivel.
- Después de una operación **ROLBK**, el proceso de registros coincidentes puede provocar un error de secuencia.

6.1.6 Ejemplo de utilización del control de compromiso

A continuación se facilita un ejemplo de las especificaciones y de los mandatos del CL para un programa que opere bajo el control de compromiso.

Para preparar la utilización del control de compromiso, emita los siguientes mandatos de CL:

CRTJRNRCV JRNRCV (RECEPTOR)

Este mandato crea un receptor de diario llamado **RECEPTOR**.

CRTJRN JRN (DIARIO) JRNRCV (RECEPTOR)

Este mandato crea un diario llamado **DIARIO** y lo conecta al receptor de diario llamado **RECEPTOR**.

STRJRNPF FILE (MAESTRO) JRN (DIARIO)

Este mandato dirige las entradas de diario del archivo **MAESTRO** al diario **DIARIO**.

En el programa, especifique **COMIT** para el archivo **MAESTRO**:

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF....LregLCIAIDePcLaEDispos+.....KSalidaEntradA....U1.*
FMAESTRO UF E K DISK KCOMIT
F*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* En las especificaciones de cálculo, utilice la operación COMIT para
C* completar un grupo de operaciones.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C CLA CHAINMAESTRO 50
C N50 UPDATEREGTRO 99
C N99 COMIT
C*
C* Si se produce una anomalía en una operación dentro de un grupo, utilice
C* la operación ROLBK para eliminar todo el grupo de operaciones.
C*
C 99 ROLBK
C*
```

Figura 53. Ejemplo de utilización del control de compromiso

Para operar con el programa (llamado **REVISAR**) bajo el control de compromiso, emita los mandatos:

STRCMTCTL LCKLVL (*ALL)

Este mandato inicia el control de compromiso con el nivel de bloqueo más alto.

CALL REVISE

Este mandato llama al programa (denominado **REVISAR**).

ENDCMTCTL

Este mandato finaliza el control de compromiso y hace que se ejecute una operación de retrotracción implícita.

7.0 Capítulo 7. Utilización de archivos DISK

Los archivos de base de datos, que estén asociados con el dispositivo DISK del RPG/400 en las posiciones 40 a 46 de las especificaciones de descripción de archivos, pueden ser:

- Archivos descritos externamente, cuyos campos se describan en el sistema OS/400 mediante las especificaciones de descripción de datos (DDS)
- Archivos descritos por el programa, cuyos campos se describen en las especificaciones de entrada/salida del programa que utiliza el archivo.

Todos los archivos de bases de datos se crean mediante mandatos de crear archivos del OS/400. Consulte la publicación *CL Reference* para obtener una descripción de los mandatos del OS/400 relacionados con archivos de base de datos.

Subtemas

- 7.1 Archivos DISK descritos externamente
- 7.2 Archivos DISK descritos por programa
- 7.3 Métodos para proceso de archivos DISK
- 7.4 Operaciones de archivo válidas

7.1 Archivos DISK descritos externamente

Los archivos DISK descritos externamente se identifican mediante una E en la posición 19 de las especificaciones de descripción de datos. La E indica que el compilador ha de recuperar la descripción externa del archivo desde el sistema cuando se compile el programa. Por lo tanto, debe crear el archivo antes de compilar el programa.

La descripción externa para un archivo DISK incluye:

- Las especificaciones de formato de registro que contienen una descripción de los campos de un registro.
- Las especificaciones de vía de acceso que describen cómo han de recuperarse los registros.

Estas especificaciones provienen de las DDS del archivo y del mandato de creación de archivos del OS/400 que se utilice para el archivo.

Subtemas

- 7.1.1 Especificaciones de formato de registro
- 7.1.2 Vía de acceso
- 7.1.3 Claves válidas para un registro o archivo
- 7.1.4 Métodos de proceso para archivos DISK descritos externamente

7.1.1 Especificaciones de formato de registro

Las especificaciones de formato de registro le permiten describir los campos de un registro así como la ubicación de los campos en un registro. Los campos se localizan en el registro según el orden en que están especificados en las DDS. La descripción del campo suele incluir el nombre del campo, el tipo de campo (caracteres, binario, decimal con zona o decimal empaquetado), y la longitud del campo (incluyendo el número de posiciones decimales de un campo numérico). En lugar de especificar los atributos de campo en el formato de registro de un archivo físico o lógico, puede definirlos en un archivo de referencias de campo.

Además, las palabras clave de las DDS pueden utilizarse para:

- Especificar que no se admiten valores de clave duplicados para el archivo (**UNIQUE**)
- Especificar una descripción de texto para un formato de registro o para un campo (**TEXT**).

Consulte en la publicación *Guía para la Base de Datos* para obtener una lista completa de las palabras clave de DDS válidas para un archivo de base de datos.

En la Figura 54 en el tema 7.1.2 se muestra un ejemplo de las DDS para un archivo de base de datos y en la Figura 55 en el tema 7.1.2 se muestra el ejemplo de un archivo de referencias de campos que define los atributos para los campos utilizados en el archivo de base de datos. Para obtener más información sobre el archivo de referencias de campos, consulte la publicación *DDS Reference*.

7.1.2 Vía de acceso

La descripción de un archivo descrito externamente contiene la vía de acceso que describe cómo han de recuperarse los registros desde el archivo. Los registros pueden recuperarse basándose en una vía de acceso en secuencia de llegada (sin claves) o en una vía de acceso en secuencia de clave.

La vía de acceso en secuencia de llegada se basa en el orden en el que los registros se almacenan en el archivo. Los registros se añaden al archivo uno después de otro.

Para la vía de acceso en secuencia de clave, la secuencia de registros del archivo se basa en el contenido del campo de clave que está definido en las DDS del archivo. Por ejemplo, en las DDS que se muestran en la Figura 54, **CLIE** se define como el campo de clave. La vía de acceso en secuencia de clave se actualiza siempre que se añadan o supriman registros, o cuando se modifica el contenido del campo de clave.

Para obtener una descripción completa de las vías de acceso de un archivo de base de datos descrito externamente, consulte la publicación *Guía para la Base de Datos*.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A.....T.Nombre++++Lon++TDpB.....Funciones+++++*****
A** ARCHIVO MAECLIL LOGICO MAESTRO CLIENTE
A                               UNIQUE
A           R REGCLI           PFILE(MAECLIP)
A                               TEXT('Registro maestro cliente')
A           CLIE
A           NOMB
A           DIRE
A           CDAD
A           PROV
A           CP
A           CODIGO
A           TIPCLI
A           SALDO
A           SALDEP
A           ULTIMP
A           ULTIMF
A           LIMCRE
A           VENTAÑ
A           VTAÑO
A           K CLIE
```

Figura 54. Ejemplo de las especificaciones de descripción de archivo para un archivo de base de datos

Las DDS de ejemplo son para el archivo lógico maestro de clientes CUSMSTL. El archivo contiene un formato de registro CUSREC (registro maestro de clientes). Los datos de este archivo están contenidos en el archivo físico MAECLIP, que se identifica mediante la palabra clave PFILE. La palabra clave UNIQUE se utiliza para indicar que no están permitidos valores de clave duplicados para este archivo. El campo **CLIE** se identifica mediante una **K** en la posición 17 de la última línea como el campo de clave de este formato de registro.

Los campos de este formato de registro se listan en el orden en que aparecerán en el registro. Los atributos de los campos se obtienen del archivo físico MAECLIP. El archivo físico, a su vez, para obtener los atributos de los campos hace referencia a un archivo de referencias de campo. El archivo de referencias de campos se muestra en la Figura 55.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A.....T.Nombre++++Lon++TDpB.....Funciones+++++*****
A**REFCPO REFDIS REFERENCIA DISTRIBUCION CAMPOS APLICACION
A           R DSTREF           TEXT('Distribución ref. campos')
A* CAMPOS COMUNES UTILIZADOS COMO REFERENCIA
A           BASDAD             6 0           EDTCDE(Y) 1
A                               TEXT('Campo base datos')
A* CAMPOS UTILIZADOS POR ARCHIVO MAESTRO CLIENTES
A           CLIE               5           CHECK(MF) 2
A                               COLHDG('Número' 'Cliente')
A           NOMB               20          COLHDG('Nombre Cliente')
A           DIRE               R           REFFLD(NOMB) 3
A                               COLHDG('Dirección Cliente')
A           CDAD               R           REFFLD(NOMB) 3
A                               COLHDG('Ciudad Cliente')
A           PROV               2           CHECK(MF) 2
```

```

A          CODIGO          6          COLHDG('Provincia')
A          CHECK(MF)      2
A          COLHDG('Código' 'Búsqueda')
A          TEXT('Código +
A          Búsqueda Núm. Cliente')
A          CP              5  0      CHECK(MF)      2
A          COLHDG('Código' 'Postal')
A          TIPCLI          1  0      RANGE(1 5)    4
A          COLHDG('Tipo' 'Cli')
A          TEXT('Tipo cliente 1=Gob 2=Fab 3=Dis
A          4=Bca 5=Otr')
A          SALDO           8  2      COLHDG('Ctas Cobr' 'Saldo')    5
A          EDTCDE(J)      6
A          SLPED           R          REFFLD(SALDO)
A          COLHDG('Imp Cta/Cob en' 'Arch +
A          Ped')

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A.....T.Nombre++++.Lon++TDpB.....Funciones+++++*****
A          ULTIMF          R          REFFLD(BASDAT)
A          COLHDG('Ultima' 'Fecha' 'Pago')
A          TEXT('Ultima fecha pago ex C/Cob')
A          LIMCRE          R          REFFLD(Saldo)
A          COLHDG('Limite' 'Credito')
A          TEXT('Limite credito cliente')
A          VENTAÑ          R          REFFLD(Saldo)
A          COLHDG('Ventas' 'Este' 'Año')
A          TEXT('Ventas Clientes este Año')
A          VTAÑO          R+  2      REFFLD(Saldo)
A          COLHDG('Ventas' 'Ultimo' 'Año')
A          TEXT('Ventas cliente ultimo año')    7

```

Figura 55. Ejemplo de un archivo de referencias de campos

Este ejemplo de un archivo de referencia de campos muestra las definiciones de los campos que utiliza el archivo MAECLIL (archivo lógico maestro de clientes) que se muestra en la Figura 54. El archivo de referencias de campos contiene normalmente definiciones de campos que otros archivos utilizan. En el texto que se incluye a continuación, se describen algunas de las entradas de este archivo de referencias de campos.

- 1 El campo **BASDAD** se edita mediante el código de edición **Y**, según se especifica mediante la palabra clave **EDTCDE(Y)**. Si un programa en RPG/400 utiliza este campo en un archivo de salida descrito externamente, el código de edición que se utilizará es el especificado en este archivo de referencia de campos; no puede alterarse temporalmente en el programa en RPG/400. Si un programa en RPG/400 utiliza este campo en un archivo de salida descrito por programa, debe especificarse un código de edición para el campo en las especificaciones de salida.
- 2 La entrada **CHECK(MF)** especifica que el campo es un campo de relleno obligatorio cuando se entra desde una estación de trabajo. Un campo de relleno obligatorio significa que tienen que entrarse todos los caracteres del campo desde la estación de trabajo.
- 3 Los campos **DIRE** y **CDAD** comparten los mismos atributos que se han especificado para el campo **NOMB**, según se indica mediante la palabra clave **REFFLD**.
- 4 La palabra clave **RANGE**, que se especifica para el campo **TIPCLI**, asegura que los únicos números válidos que pueden introducirse en este campo, desde una estación de trabajo, son de 1 a 5.
- 5 La palabra clave **COLHDG** proporciona una cabecera de columna para el campo cuando se utiliza con el Programa de Utilidad Interactivo para Bases de Datos (IDU).
- 6 El campo **SALDO** se edita mediante el código de edición **J**, tal como se indica mediante la palabra clave **EDTCDE(J)**.
- 7 Para algunos campos se facilita una descripción del texto (palabra clave **TEXT**). La palabra clave **TEXT** se utiliza a efectos de documentación y aparece en varios listados.

7.1.3 Claves válidas para un registro o archivo

Para una vía de acceso en secuencia de clave, puede definir uno o más campos en las especificaciones de descripción de datos (DDS) para utilizarlos como campos de clave para un formato de registro. (Estos campos no deben ser de coma flotante). No es necesario que todos los tipos de registro en un archivo tengan los mismos campos de clave. Por ejemplo, un registro de cabecera de pedido puede tener definido el campo **PEDIDO** como campo de clave y los registros de detalle del pedido pueden tener definidos los campos **PEDIDO** y **LINEA** como campos de clave.

La clave para un archivo viene determinada por las claves válidas para los tipos de registro de ese archivo. La clave del archivo se determina de la forma siguiente:

- Si todos los tipos de registro en un archivo tienen el mismo número de campos de clave definidos en las DDS que sean idénticas en atributos, la *clave para el archivo* está formado por todos los campos en la clave para los tipos de registro. (Los campos correspondientes no necesitan tener el mismo nombre). Por ejemplo, si el archivo tiene tres tipos de registro y la clave para cada tipo de registro está formado por los campos **A**, **B**, y **C**, la clave del archivo estará compuesta por los campos **A**, **B**, y **C**. Es decir, la clave del archivo es la misma que la clave de los registros.
- Si todos los tipos de registro en el archivo no tienen los mismos campos de clave, la clave para el archivo se compone de los campos de clave *comunes* a todos los tipos de registro. Por ejemplo, un archivo tiene tres tipos de registro y los campos de clave se han definido tal como sigue:
 - **REC1** contiene el campo de clave **A**.
 - **REC2** contiene los campos de clave **A** y **B**.
 - **REC3** contiene los campos de clave **A**, **B**, y **C**.

La clave del archivo es el campo de clave A- común para todos los tipos de registro.

- Si no hay ningún campo de clave común a todos los tipos de registro, no hay ninguna clave para el archivo.

En un programa en RPG/400 puede especificar un argumento de búsqueda en ciertos códigos de operación de archivo para identificar el registro que desea procesar. El programa en RPG/400 compara el argumento de búsqueda con la clave del archivo o del registro y efectúa la operación especificada en el registro cuya clave coincida con el argumento de búsqueda.

Subtemas

7.1.3.1 Argumentos de búsqueda válidos

7.1.3.2 Referencia a una clave parcial

7.1.3.1 Argumentos de búsqueda válidos

Puede especificar un argumento de búsqueda en las operaciones del RPG/400 **CHAIN**, **DELET**, **READE**, **REDPE**, **SETGT**, y **SETLL** que especifique un nombre de archivo o nombre de registro.

Para una operación con un nombre de archivo, el número máximo de campos que pueden especificarse en un argumento de búsqueda es igual al número total de campos de clave válidos para la clave del archivo. Por ejemplo, si todos los tipos de registro en un archivo no contienen los mismos campos de clave, puede usar una lista de claves (**KLIST**) para especificar un argumento de búsqueda que esté compuesto únicamente por el número de campos comunes a todos los tipos de registro en el archivo. Si un archivo contiene tres tipos de registro, los campos de clave se definen tal como sigue:

- **REC1** contiene el campo de clave **A**.
- **REC2** contiene los campos de clave **A** y **B**.
- **REC3** contiene los campos de clave **A**, **B** y **C**.

El argumento de búsqueda puede ser únicamente un sólo campo con atributos idénticos al campo **A** ya que dicho campo **A** es el único campo de clave común a todos los tipos de registro.

Para una operación con un nombre de registro, el número máximo de campos que pueden especificarse en un argumento de búsqueda es igual al número total de campos de clave válidos para este tipo de registro.

Si el argumento de búsqueda está formado por un campo, puede especificarse un literal, un nombre de campo o un nombre **KLIST** con **KFLD**. Si el argumento de búsqueda incluye más de un campo (una clave compuesta), debe especificar una **KLIST** con múltiples **KFLD**.

Los atributos de cada campo en el argumento de búsqueda deben ser idénticos a los atributos de clave de archivo o de registro del campo correspondiente. Los atributos incluyen la longitud, el tipo de datos (carácter o numérico) y el número de posiciones decimales. Los atributos se listan en la tabla de datos de información del campo de clave del listado del compilador. Observe un ejemplo en el Capítulo 2, "Entrada de especificaciones en RPG/400".

En todas estas operaciones de archivo (**CHAIN**, **DELET**, **READE**, **REDPE**, **SETGT**, y **SETLL**), también puede especificar un argumento de búsqueda que contenga un número inferior al total de campos válidos para el archivo o registro. Un argumento de este tipo hará referencia a una clave parcial.

7.1.3.2 Referencia a una clave parcial

Las reglas para la especificación de un argumento de búsqueda que haga referencia a una clave parcial son las siguientes:

- El argumento de búsqueda está compuesto por campos que corresponden a los campos más a la izquierda (orden superior) de la clave para el archivo o registro.
- Sólo pueden omitirse los campos más a la derecha de la lista de claves (**KLIST**) para un argumento de búsqueda que haga referencia a una clave parcial. Por ejemplo, si la clave total para un archivo o registro está compuesta por los campos de clave A, B y C, los argumentos de búsqueda válidos que hacen referencia a una clave parcial son el campo A y los campos A y B.
- Cada campo en el argumento de búsqueda debe ser idéntico en atributos al campo de clave correspondiente en el archivo o registro. Los atributos incluyen la longitud, tipo de datos (carácter o numérico), el número de posiciones decimales y el formato (por ejemplo, empaquetado o con zona).
- Un argumento de búsqueda no puede hacer referencia a parte de un campo de clave.

Si un argumento de búsqueda hace referencia a una clave parcial, el archivo se coloca en el primer registro que cumpla el argumento de búsqueda o el registro recuperado es el primer registro que cumple el argumento de búsqueda. Por ejemplo, las operaciones **SETGT** y **SETLL** colocan el archivo en el primer registro de la vía de acceso que cumple la operación y el argumento de búsqueda. La operación **CHAIN** recupera el primer registro de la vía de acceso que cumple el argumento de búsqueda. La operación **DELET** suprime el primer registro de la vía de acceso que cumple el argumento de búsqueda. La operación **READE** recupera el siguiente registro si la parte de la clave de este registro (o el el registro del tipo especificado) en la vía de acceso coincide con el argumento de búsqueda. La operación **REDPE** recupera el registro anterior si la parte de la clave de este registro (o el registro del tipo especificado) de la vía de acceso coincide con el argumento de búsqueda. Para obtener más información sobre estos códigos de operación consulte el manual *RPG/400 Reference*.

7.1.4 Métodos de proceso para archivos DISK descritos externamente

Puede procesar los archivos **DISK** descritos externamente de forma secuencial por clave, al azar por clave, al azar utilizando los números relativos de registros, secuencialmente entre límites o en orden consecutivo (sin clave o sin número relativo de registros). Una **K** en la posición 31 de las especificaciones de descripción de archivos de un archivo descrito externamente indica que el archivo ha de procesarse por clave. Si el proceso es secuencial, los registros se recuperan en secuencia de clave. Si el proceso es al azar, se emplean valores de clave para identificar los registros. Un blanco en la posición 31 indica que el archivo se procesa utilizando los números relativos de registros, secuencialmente (en secuencia de llegada) o al azar. El proceso al azar o secuencial viene determinado por las entradas en las posiciones 16 y 28 de las especificaciones de descripción de archivos y por el código de operación utilizado en las especificaciones de cálculo (por ejemplo **CHAIN**, **SETLL**, **READ**).

7.2 Archivos *DISK* descritos por programa

Los archivos descritos por el programa, que se identifican por una **F** en la posición 19 de las especificaciones de descripción de archivo, puede describirse como un archivo indexado, un archivo secuencial o un archivo de direcciones de registros.

Subtemas

7.2.1 Archivo indexado

7.2.2 Archivos secuenciales

7.2.3 Archivo de direcciones de registros

7.2.4 Archivo descrito externamente como archivo descrito por programa

7.2.1 Archivo indexado

Un archivo indexado es un archivo DISK descrito por el programa cuya vía de acceso se crea con valores de clave. Debe crear la vía de acceso para un archivo indexado utilizando las especificaciones de descripción de datos.

Un archivo indexado se identifica por una **I** en la posición 32 de las especificaciones de descripción de archivos.

Los campos de clave identifican los registros en un archivo indexado. Debe especificar la longitud del campo de clave en las posiciones 29 y 30, el formato del campo de clave en la posición 31 y la posición de comienzo del campo de clave en las posiciones 35 a 38 de las especificaciones de descripción de archivos.

Un archivo indexado puede procesarse secuencialmente por clave, secuencialmente entre unos límites o al azar por clave.

Subtemas

7.2.1.1 Argumentos de búsqueda válidos

7.2.1.1 Argumentos de búsqueda válidos

Para un archivo descrito por programa, un argumento de búsqueda debe ser un solo campo. Para las operaciones **CHAIN** y **DELET**, el argumento de búsqueda debe tener la misma longitud que el campo de clave que está definido en las especificaciones de descripción de archivos para el archivo indexado. Para las demás operaciones de archivo, el argumento de búsqueda puede ser un campo parcial.

Las DDS especifican los campos que han de utilizarse como campos de clave. Las posiciones 35 a 38 de las especificaciones de descripción de archivos especifican la posición de comienzo del primer campo de clave. La entrada en las posiciones 29 y 30 de las especificaciones de descripción de archivos deben especificar la longitud total de la clave tal como se ha definido en las DDS.

La Figura 56 y la Figura 57 muestran ejemplos de como usar las DDS para describir la vía de acceso para archivos indexados.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A.....T.Nombre++++Lon++TDpB.....Funciones+++++*****
A          R FORMATA                      PFILE(DETPED)
A          TEXT('Via de acc. para arch.
A          indexad')
A          CPOA                          14
A          PEDIDO                         5    0
A          CPOB                          101
A          K PEDIDO
A*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF....LregLCIAIDePcLaEDispos+.....KSalidaEntradA....U1.*
FDETPEDL IP F    118 3PI    15 DISK
F*
```

Figura 56. Utilización de especificaciones de descripción de datos para definir la vía de acceso de un archivo indexado

Debe utilizar las especificaciones de descripción de archivo para crear la vía de acceso de un archivo indexado descrito por el programa.

En las DDS del formato de registro FORMATA del archivo lógico DETPEDL, el campo **PEDIDO**, que tiene una longitud de cinco dígitos, se define como el campo de clave y está en formato empaquetado. La definición de **PEDIDO** como el campo de clave establece el acceso por clave a este archivo. Los otros dos campos, **CPOA** y **CPOB**, describen el resto de posiciones de este registro como campos de caracteres.

El archivo de entrada descrito por programa DETPEDL se describe en las especificaciones de descripción de archivo como un archivo indexado. Las posiciones 29 y 30 deben especificar el número de posiciones de registro necesarias para el campo de clave, tal como se definen en las DDS: tres posiciones. Las posiciones 35 a 38 especifican la posición 15 como la posición inicial del campo de clave del registro. Como el archivo está definido como descrito por el programa mediante la **F** en la posición 19, el compilador del RPG/400 no recupera la descripción externa a nivel de campo del archivo en tiempo de compilación. Por lo tanto, debe describir los campos del registro en las especificaciones de entrada.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A.....T.Nombre++++Lon++TDpB.....Funciones+++++*****
A          R FORMAT                      PFILE(DETPED)
A          TEXT('Via de acc. para arch.
A          indexad')
A          CPOA                          14
A          PEDIDO                         5
A          ARTIC                         5
A          CPOB                          96
A          K PEDIDO
A          K ARTIC
A*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF....LregLCIAIDePcLaEDispos+.....KSalidaEntradA....U1.*
FORDDTLL IP F    120 10AI    15 DISK
F*
```

Figura 57. (Parte 1 de 2). Utilización de las especificaciones de

descripción de datos para definir la vía de acceso (clave compuesta) para un archivo indexado

En este ejemplo, las especificaciones de descripción de datos definen dos campos de clave para el formato de registro FORMAT en el archivo lógico DETPEDL. Para utilizar los dos campos como una clave compuesta para un archivo indexado descrito por programa, los campos de clave deben ser contiguos en el registro.

En las especificaciones de descripción de archivos la longitud del campo de clave se define como 10 en las posiciones 29 y 30 (el número combinado de posiciones necesario para los campos **PEDIDO** y **ARTIC**). La posición de comienzo del campo de clave se describe como 15 en las posiciones 37 y 38. La posición de comienzo debe especificar la primera posición del primer campo de clave.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
IDsnombre..NODsSal-arch+.....NvecLon+.....*
ICLAV      DS
I.....campoexter.....PDesdA+++DCampo+.....*
I                                1  5 K1
I                                6 10 K2
I*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C                                MOVE PEDIDO  C1
C                                MOVE ARTIC   C2
C      CLAV      CHAINDETPEDL                                99
C*
```

Figura 58. (Parte 2 de 2). Utilización de las especificaciones de descripción de datos para definir la vía de acceso (clave compuesta) para un archivo indexado

Cuando las DDS especifican una clave compuesta, debe construir un argumento de búsqueda en el programa para realizar un **CHAIN** en el archivo. (No puede usarse una **KLIST** para un archivo descrito por el programa). Una forma de hacer esto es crear una estructura de datos con subcampos iguales a los campos de claves definidos en las DDS. Luego, en los cálculos, haga que los subcampos sean iguales al valor de los campos de claves y utilice el nombre de la estructura de datos como argumento de búsqueda en la operación **CHAIN**.

En este ejemplo, las operaciones **MOVE** definen los subcampos **K1** y **K2** con el mismo valor que los campos **PEDIDO** y **ARTIC**, respectivamente. Entonces el nombre de la estructura de datos (**KEY**) se utiliza como el argumento de búsqueda en la operación **CHAIN**.

7.2.2 Archivos secuenciales

Los archivos secuenciales son archivos en los que el orden de los registros del archivo se basa en el orden en que los registros están colocados en el archivo (es decir, en secuencia de llegada). Por ejemplo, el décimo registro colocado en el archivo ocupa la décima posición de registro.

Los archivos secuenciales pueden procesarse al azar utilizando los números relativos de registros, o consecutivamente, según un archivo de direcciones de registros. Puede utilizar los códigos de operación **SETLL** o **SETGT** para establecer los límites del archivo.

7.2.3 Archivo de direcciones de registros

Puede utilizar un archivo de direcciones de registros para procesar otro archivo. Un archivo de direcciones de registros puede contener (1) registros de límites que se utilizan para procesar un archivo secuencialmente entre límites o (2) números relativos de registros que se emplean para procesar un archivo mediante números relativos de registro. El propio archivo de direcciones de registros ha de procesarse secuencialmente.

Un archivo de direcciones de registros se identifica por una **R** en la posición 16 de las especificaciones de descripción de archivos. Si el archivo de direcciones de registros contiene números relativos de registros, debe haber una **T** en la posición 32. El nombre del archivo de direcciones de registros debe especificarse también en las posiciones 11 a 18 de las especificaciones adicionales y el nombre del archivo que ha de procesarse mediante el archivo de direcciones de registros debe especificarse en las posiciones 19 a 26 de las especificaciones adicionales.

Subtemas

7.2.3.1 Registros de límites

7.2.3.2 Números relativos de registro

7.2.3.1 Registros de límites

Para el proceso secuencial entre límites, el archivo de direcciones de registros contiene registros de límites. Un registro de límites contiene la clave de registros mas baja y la clave de registros más alta de los registros que han de leerse en el archivo.

El formato de los registros de límites en el archivo de direcciones de registros es el siguiente:

- La clave inferior empieza en la posición 1 del registro; la clave superior va inmediatamente después de la clave inferior. No puede aparecer ningún blanco entre las claves.
- Cada registro del archivo de direcciones de registros puede contener sólo un grupo de límites. Por lo tanto, la longitud del registro debe ser mayor o igual al doble de la longitud de la clave del registro.
- La clave inferior y la clave superior en el registro de límites deben tener la misma longitud. La longitud de las claves debe ser igual a la longitud del campo de clave del archivo que ha de procesarse.
- Una entrada en blanco igual en longitud al campo de clave del registro, hace que el compilador del RPG/400 lea el siguiente registro en el archivo de direcciones de registro.

7.2.3.2 Números relativos de registro

Para el proceso por número relativo de registro, el archivo de direcciones de registro contiene números relativos de registro. Cada registro recuperado desde el archivo que está procesándose se basa en un número relativo de registro del archivo de direcciones de registro. Un archivo de direcciones de registro que contenga números relativos de registro, no se puede utilizar para el proceso por límites. Cada número relativo de registro del archivo de direcciones de registro es un campo binario de múltiples bytes en el que cada campo contiene un número relativo de registro. Puede especificar la longitud del archivo de direcciones de registro como de 4, 3 o en blanco, dependiendo del fuente del archivo. Cuando utilice un archivo de direcciones de registro desde el entorno de sistema AS/400, especifique como 4 la longitud del archivo de direcciones de registro, dado que cada campo tiene 4 bytes de longitud. Cuando utilice un archivo de direcciones de registro desde el entorno de Sistema/36, especifique como 3 la longitud del archivo de direcciones de registro, dado que cada campo tiene 3 bytes de longitud. Si especifica en blanco la longitud del archivo de direcciones de registro, el compilador comprobará la longitud del registro primario en tiempo de ejecución para determinar si debe tratarlo como un archivo de direcciones de registro de 3 o de 4 bytes. Un valor menos 1 (<-1 o hexadecimal **FFFFFFF**) de número relativo de registro interrumpe la utilización de ese archivo de direcciones relativas de registro. Se produce un fin de archivo cuando se procesan todos los registros del archivo de direcciones de registros.

Archivo descrito externamente como archivo descrito por programa

7.2.4 *Archivo descrito externamente como archivo descrito por programa*

Un archivo que esté descrito externamente puede tratarse como un archivo descrito programa dentro de un programa en RPG/400. Especifique una F en la posición 19 de las especificaciones de descripción de archivo y describa los campos de los registros en las especificaciones de entrada y/o salida. Cuando se especifica una **F** en la posición 19 de las especificaciones de descripción de archivos de un archivo descrito externamente, el compilador no realiza una copia en la descripción externa.

7.3 Métodos para proceso de archivos DISK

Los métodos de proceso del archivo DISK incluyen:

- Proceso por número relativo de registro
- Proceso consecutivo
- Proceso secuencial por clave
- Proceso al azar por clave
- Proceso secuencial entre límites.

Subtemas

- 7.3.1 Proceso por número relativo de registro
- 7.3.2 Proceso consecutivo
- 7.3.3 Proceso secuencial por clave
- 7.3.4 Proceso secuencial entre límites
- 7.3.5 Ejemplos de proceso por clave

7.3.1 Proceso por número relativo de registro

El proceso de entrada o actualización al azar por número relativo de registro sólo se aplica a archivos controlados en cálculo. El código de operación **CHAIN** accede al registro deseado.

Los números relativos de registro identifican las posiciones de los registros con relación al principio del archivo. Por ejemplo, los números relativos de registro del primer, quinto y séptimo registro son 1, 5 y 7, respectivamente.

Para un archivo descrito externamente, el proceso de entrada o actualización por número relativo de registro se determina mediante un blanco en la posición 31 de las especificaciones de descripción de archivo y el código de operación **CHAIN**. El proceso de salida por número relativo de registro se determina mediante un blanco en la posición 31 y la utilización de la opción **RECNO** en la línea de continuación de las especificaciones de descripción de archivos para el archivo.

Puede utilizar la opción **RECNO** para la línea de continuación de las especificaciones de descripción de archivo para especificar un campo numérico que contenga el número relativo de registro que especifique en qué lugar se va a añadir un nuevo registro al archivo. El campo **RECNO** debe definirse como numérico con cero posiciones decimales. La longitud de campo debe ser suficiente para contener el número del registro más grande del archivo. Debe especificarse un campo **RECNO** si han de colocarse nuevos registros en el archivo utilizando las especificaciones de salida o una operación **WRITE**. Cuando actualice o añada un registro a un archivo mediante el número relativo de registro, el registro debe tener un lugar dentro del miembro. Para una actualización, este lugar puede ser un registro existente válido; para un registro nuevo, puede ser un registro suprimido. Puede utilizar el mandato de CL INZPFM para inicializar un registro y utilizarlos según el número relativo de registro. El número relativo de registro actual se coloca en el campo **RECNO** para todas las operaciones de recuperación u operaciones que vuelvan a situar el archivo (por ejemplo, **SETLL**, **CHAIN**, **READ**).

7.3.2 Proceso consecutivo

Durante el proceso consecutivo, los registros se leen en el orden en que aparecen en el archivo.

Para archivos de salida y para archivos de entrada que no usen funciones al azar (tal como **SETLL**, **SETGT**, **CHAIN**, or **ADD**), el compilador del RPG/400 toma el valor por omisión u opera como si se hubiese especificado **SEQONLY(*YES)** en el mandato de CL OVRDBF (Alterar temporalmente archivo de base de datos). (El compilador del RPG/400 no opera como si se hubiese especificado **SEQONLY(*YES)** para archivos de actualización). **SEQONLY(*YES)** permite que se coloquen múltiples registros en almacenamientos intermedios de la gestión de datos internos; a continuación los registros se transfieren al compilador del RPG/400 uno a uno durante la entrada. Si en un trabajo dos archivos lógicos utilizan el mismo archivo físico, y uno de ellos se procesa consecutivamente y el otro se procesa para actualización al azar, podría actualizarse un registro que ya haya sido colocado en el almacenamiento intermedio que se le ha dado al programa. En este caso, cuando se procese el registro desde el archivo consecutivo, el registro no reflejará los datos actualizados. Para evitar este problema, utilice el mandato de CL OVRDBF y especifique la opción **SEQONLY(*NO)**, la cual indica que no desea transferir múltiples registros para un archivo que se procese consecutivamente.

Para obtener más información acerca del proceso sólo secuencial, consulte la publicación *Guía para la Base de Datos*.

7.3.3 Proceso secuencial por clave

En el método de proceso secuencial por clave, los registros se leen desde el archivo en secuencia de clave.

El método de proceso secuencial por clave es válido para archivos por clave que se utilicen como archivo primario, secundario o controlado en cálculo.

Para archivos de salida o para archivos de entrada que no empleen funciones al azar (como por ejemplo **SETLL**, **SETGT**, **CHAIN**, o **ADD**) y que tengan sólo un formato de registro, el compilador del RPG/400 toma por omisión u opera como si se hubiese especificado **SEQONLY(*YES)** en el mandato de CL OVRDBF. (El compilador del RPG/400 no opera como si se hubiese especificado **SEQONLY(*YES)** para archivos de actualización). **SEQONLY(*YES)** permite que se coloquen múltiples registros en almacenamientos intermedios de la gestión de datos internos; a continuación los registros se transfieren al compilador del RPG/400 uno a uno durante la entrada. Si en un trabajo dos archivos utilizan el mismo archivo físico y uno de ellos se procesa secuencialmente y el otro se procesa al azar para actualización, podría actualizarse un registro que ya se hubiese colocado en el almacenamiento intermedio que se le ha presentado al programa. En este caso, cuando se procese el registro desde el archivo secuencial, el registro no reflejará los datos actualizados. Para evitar este problema, utilice el mandato de CL OVRDBF y especifique la opción **SEQONLY(*NO)**, que indica que no se desean transferir múltiples registros para un archivo que se procese secuencialmente.

Para obtener más información acerca del proceso sólo secuencial, consulte la publicación *Guía para la Base de Datos*.

La Figura 59 muestra distintas maneras en que pueden procesarse el registro de cabecera y los registros de detalle asociados con el registro de cabecera. La parte 1 muestra un ejemplo de un archivo que está leyéndose secuencialmente por claves; las partes 2 a 4 muestran ejemplos en los que se utiliza el código de operación **READ**; la parte 5 muestra el proceso de esos registros mediante la técnica de registros coincidentes.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*
F* En este ejemplo, el registro de cabecera de pedido (CABPEDL)
F* y el de detalle de pedido (DETPED) están contenidos en el mismo
F* archivo (ARCHPEL), definido como un archivo de entrada primario
F* con lectura secuencial por tecla. En las especificaciones de
F* descripción de datos del archivo, la clave del registro CABPED
F* se define como el campo PEDIDO y la clave para el registro DETPED
F* se define como el campo PEDIDO más el campo LINE (número de
F* línea), que es una clave compuesta.
F*
FNombarchIPEAF....LregLC1AIDePcLaEDispos+.....KSalidaEntradA....U1.*
FARCHPEL IP E K DISK

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I*
I* Se asigna un identificador identificativo a cada registro.
I* Estos identificadores identificativos de registro se utilizan
I* para controlar el proceso de los diferentes tipos de registro.
INomreg++....In.....*
ICABPED 01
I*
IDETPED 02
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C *IN01 IFEQ '1'
C " +-----+
C " |Proceso cabeceras |
C " +-----+
C END
C*
C *IN02 IFEQ '1'
C " +-----+
C " |Proceso de detalle|
C " +-----+
C END

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*
C* Este ejemplo es el mismo que el anterior, a excepción de que el
F* archivo ARCHPEL se define como archivo controlado en cálculo y la
F* lectura del archivo se efectúa mediante el código de operación READ.
F*
CFNombarchIPEAF....LregLC1AIDePcLaEDispos+.....KSalidaEntradA....U1.*
FARCHPEL IF E K DISK
```

F*

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...
I*
I* Los dos registros (CABPED y DETPED) se encuentran en el mismo
I* archivo y se asigna un indicador indentificativo de registro a
I* cada uno. Los indicadores identificativos de registro se utilizan
I* para controlar el proceso de los distintos tipos de registro.
I* Para archivos controlados en cálculo no se pueden identificar los
I* niveles de control de comparación.
I*

INomreg++....In.....*

ICABPED 01

I*

IDETPED 02

I*

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...

C*

C* El código de operación READ lee un registro del archivo ACHPEL. Se
C* especifica un indicador de fin de archivo en las posiciones 58 y 59.
C* Si la operación READ activa el indicador de fin de archivo, 99,
C* el programa bifurca al identificador FINARC y procesa la rutina
C* de fin de archivo. Los indicadores de identificación de registro
C* controlan el proceso de los distintos tipos de registro.
C*

CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*

C READ ARCHPEL 99

C 99 GOTO FINARC

C*

C *IN01 IFEQ '1'
C " +-----+
C " |Proceso cabeceras |
C " +-----+
C **END**

C*

C *IN02 IFEQ '1'
C " +-----+
C " |Proceso de detalle|
C " +-----+
C **END**

C*

C FINARC TAG
C " +-----+
C " |Rutina fin de archivo|
C " +-----+

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...

F*

F* Este ejemplo es similar al de la Parte 2 de esta figura. Sin
F* embargo, se utiliza el código de operación READ para leer cada
F* registro (CABPED y DETPED) en lugar de leer el archivo. La lógica
F* del programa controla cuándo ocurre cada READ. No se necesitan
F* indicadores identificativos de registro, ya que la lógica del
F* programa conoce con qué registro está trabajando, según sea el
F* nombre del formato de registro.
F*

FNombarchIPEAF....LregLCIAIDePcLaEDispos+.....KSalidaEntrada....U1.*

FARCHPEL IF E K DISK

F*

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...

CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*

C READ CABPED 99

C 99 GOTO FIN

C

C " +-----+
C " |Proceso cabeceras |
C " +-----+

C*

C READ DETPED 99

C 99 GOTO FIN

C

C " +-----+
C " |Proceso de detalle|
C " +-----+

C*

C FIN TAG

C

C

C

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...

F*

F* En este ejemplo, el archivo CABPEDL contiene los registros de

F* cabecera de pedido (CABPED) y el archivo DETPEDL contiene los
F* registros de detalle de pedido (DETPED). El archivo CABPEDL se
F* define como un archivo de entrada primario y el ciclo del
F* programa controla la lectura de los registros de este archivo.
F* El archivo DETPEDL se define como un archivo controlado en
F* cálculo y utiliza la operación READE para leer los registros del
F* archivo

F*
FNombarchIPEAF....LregLC1AIDePcLaEDispos+.....KSalidaEntradA....U1.*

FCABPEDL IF E K DISK
FDETPEDL IP E K DISK

F*
.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

C* El campo PEDIDO de la operación SETLL se utiliza para posicionar
C* el archivo DETPEDL en el primer registro DETPED que tenga una
C* clave igual o mayor que el contenido del campo PEDIDO. El campo
C* PEDIDO se utiliza como argumento de búsqueda de la operación
C* READE. Esta operación recupera el siguiente registro DETPED del
C* archivo si la clave del registro es igual al argumento de
C* búsqueda especificado en el factor 1. Si la clave y el argumento
C* de búsqueda no son iguales, se activa el indicador especificado
C* en las posiciones 58 y 59.

C*
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*

```

C          "          +-----+
C          "          |Proceso cabeceras |
C          "          +-----+
C          "
C          PEDIDO      SETLLDETPED          20
C N20              GOTO NO
C          BUCLE      TAG
C          PEDIDO      READEDETPED          21
C 21              GOTO FINGRU
C          "
C          "          +-----+
C          "          |Proceso de detalle|
C          "          +-----+
C          "
C          GOTO BUCLE
C          NO          TAG
C          "
C          "
C          FINGRU      TAG
    
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

F* En este ejemplo, el archivo CABPEDL contiene los registros de
F* cabecera de pedido (CABPED) y el archivo DETPEDL contiene los
F* registros de detalle de pedido (DETPED). El archivo CABPEDL se
F* define como un archivo de entrada primario y el archivo DETPEDL
F* como un archivo de entrada secundario. Los registros de cabecera
F* de pedido y de detalle de pedido se procesan utilizando la
F* técnica de registros coincidentes, con el campo PEDIDO de ambos
F* registros asignado al valor de nivel de comparación M1. Se han
F* asignado a los registros los indicadores identificativos 01 y
F* 02 para controlar el proceso de los diferentes tipos de registro.

F*
FNombarchIPEAF....LregLC1AIDePcLaEDispos+.....KSalidaEntradA....U1.*

FDETPEDL IP E K DISK
FDETPEDL IS E K DISK

F*
.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

```

INomreg++....In.....*
ICABPED      01
I                      PEDIDO M1
IDETPED      02
I                      PEDIDO M1
    
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*

```

C 01NMR          "          +-----+
C 01 MR          "          |Proceso cabeceras |
C 01 MR          "          +-----+
C*
C 02NMR          "          +-----+
C 02 MR          "          |Proceso de detalle|
C 02 MR          "          +-----+
    
```

Figura 59. Proceso de registros de cabecera de pedido y detalle de pedido

7.3.4 Proceso secuencial entre límites

El proceso secuencial entre límites utilizando un archivo de direcciones de registros se especifica por una **L** en la posición 28 de las especificaciones de descripción de archivos y es válido para un archivo con acceso por claves.

Puede especificar el proceso secuencial entre límites para un archivo de entrada o de actualización que se designe como archivo primario, secundario o controlado en cálculo. El archivo puede ser descrito externamente o estar descrito por el programa (indexado). El archivo debe tener las claves en secuencia ascendente.

Para procesar un archivo secuencialmente entre los límites de un archivo de direcciones de registros, el programa lee:

- Un registro de límites desde el archivo de direcciones de registros
- Registros del archivo que está procesándose entre límites, con claves mayores o iguales a la clave de registro superior en el registro entre límites. Si los dos límites suministrados por el archivo de direcciones de registros son iguales, sólo se recuperan los registros con la clave especificada.

El programa repite este procedimiento hasta que llega al final del archivo de direcciones de registros.

La Figura 61 en el tema 7.3.5 muestra un ejemplo de un archivo indexado que se procesa secuencialmente entre límites. La Figura 62 en el tema 7.3.5 muestra el mismo ejemplo con archivos descritos externamente en lugar de con archivos descritos por el programa.

7.3.5 Ejemplos de proceso por clave

La Figura 63 muestra un ejemplo del proceso de algunos registros de un grupo. La Figura 64 muestra ejemplos de cómo procesar el primer registro de un archivo y el último registro de un archivo.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF....LregLCIAIDePcLaEDispos+.....KSalidaEntradA....U1.*
FCAMBIO IP E K DISK
FMAESTRO UF E K DISK
F*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
INomreg++....In.....*
IREGMAE 01
IREGCAM 02
I*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C 02 CTA CHAINREGMAE 03
C 02N03 MOVE NUEVO NOMDIR
C 02N03 UPDATREGMAE
C*
```

Figura 60. Proceso al azar por clave de un archivo DISK descrito externamente

El archivo de actualización MAESTRO tiene que procesarse mediante claves. Las DDS de cada uno de los archivos descritos externamente (MAESTRO y CAMBIO) identifican el campo **CTA** como campo de clave. Como cada registro se lee del archivo de entrada primario, CAMBIO, el campo de número de cuenta (**CTA**) se utiliza como argumento de búsqueda para efectuar el encadenamiento con el registro correspondiente del archivo MAESTRO. Las especificaciones de entrada se emplean para asignar indicadores identificativos de registros en los archivos CAMBIO y MAESTRO. El archivo MAESTRO contiene un formato de registro REGMAE que contiene dos campos, **CTA** y **NOMDIR** (nombre y dirección). El archivo CAMBIO contiene un formato de registro REGCAM que contiene dos campos, **CTA** y **NUEVO**. Los datos del campo **NUEVO** deben trasladarse al campo **NOMDIR** antes de que se pueda actualizar REGMAE.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*
F* El archivo de entrada MAESTR, que es un archivo descrito por
F* programa (F en la posición 19) se describe como un archivo
F* indexado que se va a procesar por clave. (La vía de acceso para
F* el archivo indexado tiene que crearse mediante especificaciones
F* descripción de datos).
F*
F* El archivo de direcciones de registros LIMITE procesa el archivo
F* MAESTR secuencialmente entre límites (L en la posición 28). Cada
F* grupo de límites del archivo de direcciones de registros
F* correspondiente está formado por los registros del archivo MAESTR
F* que han de procesarse. Como el campo de clave de número de cuenta
F* (CTA) tiene una longitud de 8 posiciones, cada grupo de límites
F* contiene dos claves de 8 caracteres.
F*
FNombarchIPEAF....LregLCIAIDePcLaEDispos+.....KSalidaEntradA....U1.*
FLIMITE IR F 16 8 EDISK
FMAESTR IP F 64L 8AI 1 DISK
FIMPRE O F 96 OF PRINTER
F*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
E*
E* En las posiciones 11 a 18 de las especificaciones de extensión
E* debe especificarse el nombre del archivo de direcciones de
E* registros, LIMITE. En las posiciones 19 a 26 de las
E* especificaciones de extensión debe especificarse el nombre del
E* archivo que procesará el archivo de direcciones de registros.
E*
E...Arch(de)Arch(a)+NombreN/rN/tbLonPDSNommatLonPDSComentarios++++*
E LIMITE MAESTR
E*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I*
I* Tienen que utilizarse las especificaciones de entrada para
I* describir los registros del archivo MAESTR descrito por programa.
I*
INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
IMAESTR NS 01
I.....PDesdeA+++DNomcamL1M1FrP1NeCe...*
I 1 8 CTA
I 9 64 NOMDIR
I*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
O*
O* Como MESTR se procesa entre cada grupo de límites, se imprimen
O* los registros correspondientes. El proceso de MESTR termina cuando
O* se llega al fin de archivo de direcciones de de registro LIMITE.
O*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OIMPRE D 1 01
O.....N01N02N03Campo+YBUposPConstante/palabraedición+...*
O CTA 8
O NOMDIR 70
O*
```

Figura 61. Proceso secuencial entre límites de un archivo indexado

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF....LregLCIAIDePcLaEDispos+.....KSalidaEntrada....U1.*
FLIMITE IR F 16 8 EDISK
FMAESTR IP E L K DISK
FIMPRE O F 96 OF PRINTER
F*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
INomreg++....In.....*
IREGMAE 01
I*
```

Figura 62. Proceso secuencial entre límites de un archivo descrito externamente

Si el programa mostrado en la Figura 61 utilizase archivos descritos externamente, las especificaciones de descripción de archivos deberían codificarse como se muestra en este ejemplo. Las especificaciones de entrada se emplean para asignar un indicador identificativo de registro para el registro en el archivo descrito externamente. El archivo MAESTR contiene el formato de registro REGMAE. Las descripciones externas para el archivo identifican los campos de claves. Esas claves deben estar en secuencia ascendente.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* Este ejemplo muestra cómo recuperar el primer registro de un grupo.
C* La operación SETLL se utiliza para posicionar el archivo en el
C* primer registro de DETPED que tenga una clave igual o mayor que el
C* argumento de búsqueda contenido en el campo PEDIDO. La operación
C* READE lee el siguiente registro DETPED del archivo si la clave del
C* registro es igual al argumento de búsqueda (campo PEDIDO) especificado
C* en el factor 1. Si la clave no es igual al argumento de búsqueda, se
C* activa el indicador especificado en las posiciones 58 y 59.
C*
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C PEDIDO SETLLDETPED
C PEDIDO READEDETPED 22
C 22 GOTO FINARC
C "
C " +-----+
C " |Proceso de DETPED |
C " +-----+
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* Este ejemplo muestra cómo recuperar el último registro de un
C* grupo. La operación SETGT se utiliza para situar el archivo en el
C* siguiente registro DETPED que tenga una clave mayor que el
C* argumento de búsqueda contenido en el campo PEDIDO. La operación
C* REDPE lee el registro DETPED inmediatamente anterior al archivo
C* la clave del registro es igual al argumento de búsqueda (campo PEDIDO)
C* especificado en el factor 1. Si la clave no es igual al argumento
```

C* de búsqueda, se activa el indicador especificado en las posiciones
C* 58 y 59.

```
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          PEDIDO      SETGTDETPED
C          PEDIDO      REDPEDETPED          22
C  22          GOTO FINARC
C          "
C          " +-----+
C          " |Proceso de DETPED |
C          " +-----+
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

C* Este ejemplo muestra como recuperar el último registro del grupo
C* precedente. El campo PEDIDO que contiene la clave del grupo actual,
C* se utiliza en la operación SETLL para posicionar el archivo en el
C* primer registro de DETPED que tenga una clave igual o mayor que el
C* argumento de búsqueda contenido en el campo PEDIDO. La operación
C* READP lee, a continuación, el registro precedente. Si no existe
C* registro precedente en el archivo, el programa bifurca a la rutina
C* FINARC.

```
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          PEDIDO      SETLLDETPED
C          READPDETPED          22
C  22          GOTO FINARC
C          "
C          " +-----+
C          " |Proceso de DETPED |
C          " +-----+
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

C* Este ejemplo recupera el último registro de un grupo. El grupo
C* contiene uno o más registros en el archivo. La operación SETGT
C* posiciona el archivo en el siguiente registro que contiene un
C* valor de clave mayor que el argumento de búsqueda contenido en
C* el campo PEDIDO. Por ejemplo, si el campo PEDIDO contiene un
C* valor de 10, la operación SETGT posiciona el archivo en el
C* registro que contiene un valor de clave mayor que 10.

```
C*
C*          Claves
C*          9
C*          9
C*          10
C*          10
C* SETGT -----
C*          11
C*
```

C* La operación READP lee a continuación el registro precedente del
C* formato de registro DETPED, leyendo, por lo tanto, el último
C* registro del grupo precedente. La operación READP requiere un
C* indicador de fin de archivo en las posiciones 58 y 59; por lo
C* tanto, si se encuentra el comienzo del archivo, se activa el
C* indicador de parada H6 y el programa termina de forma anormal.

```
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          PEDIDO      SETGTDETPED
C          READPDETPED          H6
C  H6          RETRN
C          "
C          " +-----+
C          " |Proceso de DETPED |
C          " +-----+
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

C* La lectura del primer registro del grupo siguiente requiere que
C* la operación SETGT posiciones el archivo y la operación READ.
C* El campo pedido, que contiene la clave del grupo actual, se
C* especifica en el factor 1 de la operación SETGT. A continuación,
C* se utiliza la operación READ para leer el primer registro del
C* grupo siguiente. Tiene que especificarse un indicador en las
C* posiciones 58 y 59 de la operación READ para comprobar la
C* condición de fin de archivo. Esta técnica puede utilizarse cuando
C* el programa conoce el valor de clave de un grupo de registros o
C* registro específico y desea obtener el siguiente grupo. La
C* operación SETGT puede utilizarse para eliminar la lectura de
C* registros no deseados.

```
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          PEDIDO      SETGTDETPED
C          READ DETPED          22
C  22          GOTO FINARC
C          "
C          " +-----+
```

```
C          "      |Proceso de DETPED |
C          "      +-----+
```

Figura 63. Proceso de determinados registros en un grupo

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* Una vez abierto el archivo, se recupera el primer registro
C* mediante una operación READ subsiguiente. Para acceder al primer
C* registro en un archivo después de haber efectuado algún proceso,
C* se utiliza la constante figurativa *LOVAL (suponiendo secuencia
C* ascendente de clave) en la operación SETLL para establecer los
C* límites inferiores. La operación READ lee el siguiente registro
C* del formato de registro DETPED. Si no existe otro registro, se
C* produce una condición de fin de archivo y el programa bifurca a
C* la rutina NADA.
C*
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C          *LOVAL      SETLLDETPED
C          READ DETPED          22
C  22          GOTO NADA
C          "
C          "      +-----+
C          "      |Proceso de DETPED |
C          "      +-----+
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C* Utilice la constante figurativa *HIVAL (suponiendo secuencia
C* ascendente de clave) para acceder al último registro en un archivo.
C* Utilizando *HIVAL con la operación SETGT, se sitúa el archivo en
C* el siguiente registro que tenga un valor de campo de clave mayor
C* que el valor especificado en el factor 1.
C* La operación READP lee el siguiente registro anterior que, en este
C* ejemplo, es el último registro del archivo.
C*
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C          *HIVAL      SETGTDETPED
C          READPDETPED          22
C  22          GOTO NADA
C          "
C          "      +-----+
C          "      |Proceso de DETPED |
C          "      +-----+
```

Figura 64. Proceso de determinados registros en un archivo

7.4 Operaciones de archivo válidas

La Figura 65 muestra los códigos de operación de archivo válidos que se permiten en archivos DISK procesados mediante claves y la Figura 66 muestra los mismos códigos para archivos DISK que se procesen mediante métodos sin claves. Las operaciones que aparecen en estas figuras son válidas para archivos DISK descritos externamente y archivos DISK descritos por programa.

Antes de ejecutar el programa, puede alterar temporalmente un archivo y pasarlo a otro archivo. Especialmente puede alterar temporalmente un archivo secuencial del programa y convertirlo en un archivo con claves descrito externamente. (El archivo se procesa como un archivo secuencial). También puede alterar temporalmente un archivo con claves del programa a otro archivo con claves, siempre que los campos de clave sean compatibles. Por ejemplo, el archivo de alteración temporal no debe tener un campo de clave menor que el especificado en el programa.

Descripción de Archivos					Cálculo
Especificación de Posiciones					Especificación de Posiciones
15	16	28(1)	31(2)	66	28-32
I	P/S		K/A/P		CLOSE, FEOD, FORCE
I	P/S		K/A/P	A	WRITE, CLOSE, FEOD, FORCE
I	P/S	L	K/A/P		CLOSE, FEOD, FORCE
U	P/S		K/A/P		UPDAT, DELET, CLOSE, FEOD, FORCE
U	P/S		K/A/P	A	UPDAT, DELET, WRITE, CLOSE, FEOD, FORCE
U	P/S	L	K/A/P		UPDAT, CLOSE, FEOD, FORCE
I	F		K/A/P		READ, READE, REDPE, READP, SETLL, SETGT, CHAIN, OPEN, CLOSE, FEOD
I	F		K/A/P	A	WRITE, READ, REDPE, READE, READP, SETLL, SETGT, CHAIN, OPEN, CLOSE, FEOD
I	F	L	K/A/P		READ, OPEN, CLOSE, FEOD
U	F		K/A/P		READ, READE, REDPE, READP, SETLL, SETGT, CHAIN, UPDAT, DELET, OPEN, CLOSE, FEOD
U	F		K/A/P	A	WRITE, UPDAT, DELET, READ, READE, REDPE, READP, SETLL, SETGT, CHAIN, OPEN, CLOSE, FEOD
U	F	L	K/A/P		READ, UPDAT, OPEN, CLOSE, FEOD
O	Blanc's		K/A/P	A	WRITE (añadir nuevos registros a un archivo), OPEN, CLOSE, FEOD
O	Blanc's		K/A/P		WRITE (carga inicial de un nuevo archivo)(3), OPEN, CLOSE, FEOD
<p>Nota: (1) Debe especificarse una L en la posición 28 para especificar proceso secuencial entre límites de un archivo de entrada o actualización mediante un archivo de direcciones de registro.</p>					
<p>Nota: (2) Los archivos descritos externamente requieren una K en la posición 31; los archivos descritos por programa requieren una A o una P en la posición 31 y una I en la posición 32.</p>					
<p>Nota: (3) No se requiere una A en la posición 66 para la carga inicial de registros en un nuevo archivo. Si se especifica una A en la posición 66, debe especificarse ADD en las especificaciones de salida. El archivo debe haberse creado con el mandato CREATE FILE del OS/400.</p>					

Figura 65. Operaciones de archivo válidas para métodos de proceso por claves (al azar por clave, secuencial por clave, secuencias entre límites)

Descripción de Archivos Especificación de Posiciones				Cálculo Especificación de Posiciones	
15	16	31	54-59	66	28-32
I	P/S	Blanc			CLOSE, FEOD, FORCE
I	P/S	Blanc	RECNO		CLOSE, FEOD, FORCE
U	P/S	Blanc			UPDAT, DELET, CLOSE, FEOD, FORCE
U	P/S	Blanc	RECNO		UPDAT, DELET, CLOSE, FEOD, FORCE
I	F	Blanc			READ, READP, SETLL, SETGT, CHAIN, OPEN, CLOSE, FEOD
I	F	Blanc	RECNO		READ, READP, SETLL, SETGT,
U	F	Blanc			READ, READP, SETLL, SETGT, CHAIN, UPDAT, DELET, OPEN, CLOSE, FEOD
U	F	Blanc	RECNO		READ, READP, SETLL, SETGT, CHAIN, UPDAT, DELET, OPEN, CLOSE, FEOD
I	R	A/P/ Blanc(1)			OPEN, CLOSE, FEOD
I	R	Blanc(2)			OPEN, CLOSE, FEOD
O	Blanc	Blanc	RECNO	A	WRITE(3) (adición de registros a un archivo), OPEN, CLOSE, FEOD
O	Blanc	Blanc	RECNO		WRITE(4) (carga inicial de un nuevo archivo), OPEN, CLOSE, FEOD
O	Blanc	Blanc	Blanc		WRITE (carga secuencial o ampliación de un archivo), OPEN, CLOSE, FEOD
<p>Nota: (1) Si la posición 31 está en blanco para un archivo de direcciones entre límites, el formato de las claves en el archivo de direcciones es el mismo que el formato de las claves del archivo que se procesa.</p>					
<p>Nota: (2) Un archivo de direcciones de registros que contenga números relativos de registro requiere una T en la posición 32.</p>					
<p>Nota: (3) El campo RECNO que contiene el número relativo de registro tiene que establecerse con anterioridad a la operación WRITE o si se especifica ADD en las especificaciones de salida.</p>					
<p>Nota: (4) No se requiere una A en la posición 66 para la carga inicial de los registros en un nuevo archivo; sin embargo, si se especifica A en la posición 66, debe especificarse ADD en las especificaciones de salida. El archivo debe haberse creado con el mandato CREATE FILE del OS/400.</p>					

Figura 66. Operaciones de archivo válidas para métodos de proceso sin clave (secuencial, al azar por número relativo de registro y consecutivo)

8.0 Capítulo 8. Utilización de archivos WORKSTN

El archivo WORKSTN permite que un programa en RPG/400 se comunique interactivamente con un usuario de una estación de trabajo o que utilice la Función para las comunicaciones entre sistemas (ICF), referida a continuación como ICF, para comunicarse con otros programas. En este capítulo se describen:

- Función para las comunicaciones entre sistemas (ICF)
- Archivos WORKSTN descritos externamente
- Archivos WORKSTN descritos por programa
- Archivos de dispositivos múltiples.

Este capítulo incluye también algunos ejemplos sobre la utilización de archivos WORKSTN.

Subtemas

- 8.1 Función para las comunicaciones entre sistemas
- 8.2 Archivos WORKSTN descritos externamente
- 8.3 Proceso de archivos WORKSTN
- 8.4 Archivo WORKSTN descrito por programa
- 8.5 Archivos de múltiples dispositivos
- 8.6 Ejemplos de archivo WORKSTN

8.1 Función para las comunicaciones entre sistemas

Puede utilizar la ICF (ICF) para escribir programas que comunicarán con otros programas de aplicación en otros sistemas (enviar y recibir datos).

Para utilizar la ICF defina un archivo WORKSTN en su programa que se refiera a un archivo de dispositivo de ICF. Utilice el archivo proporcionado por el sistema **QICDMF** o un archivo creado utilizando el mandato CRTICCF de OS/400.

Codifique la ICF utilizando la ICF como un archivo del programa. La ICF es similar a un archivo de pantalla y contiene los formatos de comunicaciones requeridos para enviar o recibir datos entre sistemas.

Para obtener más información sobre la ICF, consulte la publicación *ICF Programmer's Guide*.

8.2 Archivos WORKSTN descritos externamente

Un archivo WORKSTN del RPG/400 puede utilizar un archivo de dispositivo de pantalla descrito externamente o un archivo de dispositivo ICF, el cual contiene información sobre el archivo y una descripción de los campos en los registros que han de grabarse.

Además de las descripciones de campo (como por ejemplo nombres de campo y atributos), las DDS de un archivo de dispositivo de pantalla se emplean para:

- Dar formato a la colocación del registro en la pantalla especificando las entradas de número de línea y de número de posición por cada campo y constante.
- Especificar funciones de atención tales como subrayado y campos con alta intensidad, imagen con contraste invertido o cursor parpadeante.
- Especificar comprobación de validez para los datos introducidos en la estación de trabajo de pantalla. Las funciones de comprobación de validez incluyen la detección de campos donde son necesarios datos, la detección de campos que se tienen que rellenar obligatoriamente, la detección de tipos incorrectos de datos, la detección de datos para un rango específico, la comprobación de datos para una entrada válida y el proceso de la verificación de 10 u 11 dígitos de comprobación de módulos.
- Controlar funciones de gestión de pantallas, como por ejemplo si los campos han de borrarse, recubrirse o retenerse cuando se visualizan datos nuevos.
- Asociar los indicadores 01 a 99 con las teclas de atención de mandatos o con las teclas de función de mandatos. Si una tecla de función se describe como una tecla de función de mandato (**CF**), se devuelven al programa tanto el registro de datos (con todas las modificaciones entradas en la pantalla) como el indicador de respuesta. Si una tecla de función se describe como una tecla de atención de mandato (**CA**), se devuelve al programa un indicador de respuesta, pero el registro de datos permanece inalterado. De ese modo, los campos de tipo carácter, únicamente de entrada, están en blanco y los campos numéricos, únicamente de entrada, se rellenan con ceros a no ser que esos campos se hayan inicializado de otra manera.
- Asignar un código de edición (**EDTCDE**) o una palabra clave de edición (**EDTWRD**) a un campo para especificar cómo han de visualizarse los valores de los campos.
- Especificar subarchivos.

Un formato de registro de dispositivo de pantalla contiene tres tipos de campos:

- Campos de entrada.* Los campos de entrada se transfieren desde el dispositivo al programa cuando el programa lee un registro. Los campos de entrada pueden inicializarse con un valor por omisión. Si no se ha modificado el valor por omisión, dicho valor se transfiere al programa. Los campos de entrada que no están inicializados se visualizan como blancos en los que el usuario de la estación de trabajo puede entrar datos.
- Campos de salida.* Los campos de salida se transfieren desde el programa al dispositivo cuando el programa graba un registro en una pantalla. Los campos de salida puede suministrarlos el programa o el formato de registro del archivo de dispositivo.
- Campos de entrada/salida (bivalente).* Un campo de entrada/salida es un campo de salida que puede modificarse. Se convierte en un campo de entrada si se modifica. Los campos de entrada/salida se transfieren desde el programa cuando éste graba un registro en una pantalla y se transfieren al programa cuando éste lee un registro desde la pantalla. Los campos de entrada/salida se emplean cuando el usuario ha de cambiar o actualizar los datos que el programa ha grabado en la pantalla.

Si especifica la palabra clave **INDARA** en las DDS para un archivo WORKSTN, el programa en rn RPG/400 transfiere indicadores al archivo WORKSTN en un área de indicador distinta y no en el almacenamiento intermedio de entrada/salida.

Para una descripción detallada de un archivo de dispositivo de datos descrito externamente y una lista de las palabras clave de las DDS que son válidas, consulte la publicación *DDS Reference*.

La Figura 67 en el tema 8.2.1 muestra un ejemplo de las DDS para un archivo de dispositivo de pantalla.

Subtemas

8.2.1 Proceso de un archivo WORKSTN descrito externamente

8.2.2 Indicadores de tecla de función en archivos de dispositivo de pantalla

8.2.3 Teclas de mandato en archivos de dispositivo de pantalla

8.2.1 Proceso de un archivo WORKSTN descrito externamente

Cuando se procesa un archivo WORKSTN descrito externamente, el sistema OS/400 transforma los datos del programa al formato especificado para el archivo y visualiza los datos. Cuando se transfieren los datos al programa, se transforman al formato empleado por el programa.

El sistema OS/400 proporciona información para el control del dispositivo para que éste efectúe operaciones de entrada/salida. Cuando se solicita un registro de entrada del dispositivo, el sistema OS/400 emite la solicitud y a continuación elimina de los datos la información de control del dispositivo antes de transferir los datos al programa. Además, el sistema OS/400 puede transferir indicadores al programa que reflejan qué campos, si hay alguno, se han alterado en el registro.

Cuando el programa solicita una operación de salida, transfiere el registro de salida al sistema OS/400. El sistema OS/400 proporciona la información de control del dispositivo necesaria para visualizar el registro. Cuando se visualiza el registro, se añade también cualquier información constante especificada para el formato de registro.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre++++RLon++TdpBLínPosFunciones+++++*****
A** CONSULTA DEL MAESTRO DE ARTICULOS
A          REF(DSTREF) 1
A          R PROMPT      TEXT('Formato Solicitud Artículo')
A 73N61      OVERLAY 2
A          CA01(98 'Fin del Programa') 3
A          1 2'Consulta de Artículo'
A          3 2'Número de Artículo'
A          ITEM        R      I 3 15PUTRETAIN 4
A 61          ERRMSG('Número de Artículo no Válido' 61) 5
A          R RESPONSE  TEXT('Formato de Respuesta')
A          OVERLAY 2
A          LOCK 6
A          5 2'Descripción'
A          DESCRP     R      5 15
A          5 37'Precio'
A          PRICE      R      5 44
A          7 2'Ubicación en Almacén' 7
A          WHSLOC     R      7 22
A          9 2'Existencias'
A          ONHAND     R      9 10
A          9 19'Asignado' 8
A          ALLOC      R      9 30
A          9 40'Disponible'
A          AVAIL      R      9 51
A*

```

Figura 67. Ejemplo de las especificaciones de descripción de datos para un archivo de dispositivo de pantalla

Este archivo de dispositivo de pantalla contiene dos formatos de registro: **PROMPT** (solicitud) y **RESPONSE** (respuesta).

- 1 Los atributos de los campos de este archivo de definen en el archivo de referencias de campos **DEFDES**.
- 2 Se utiliza la palabra clave **OVERLAY** de forma que ambos formatos de registro pueden emplearse en la misma pantalla.
- 3 La tecla de función 1 está asociada con el indicador 98, utilizado por el programador par finalizar el programa.
- 4 La palabra clave **PUTRETAIN** permite mantener en la pantalla el valor que se introduce en el campo **ITEM**. Además, se define el campo **ITEM** como un campo de entrada mediante la **I** en la posición 38. **ITEM** es el único campo de entrada en estos formatos de registro. Los demás campos de registro son campos de salida ya que la posición 38 para cada uno de ellos está en blanco.
- 5 La palabra clave **ERRMSG** identifica el mensaje de error que tiene que visualizarse si se activa el indicador 61 en el programa que utiliza este formato de registro.
- 6 La palabra clave **LOCK** impide al usuario de la estación de trabajo utilizar el teclado cuando visualiza inicialmente el formato de registro **RESPONSE**.
- 7 Las constantes tales como 'Descripción', 'Precio', y 'Situación en almacén' describen los campos que graba el programa.

- 8 Las entradas de línea y posición identifican donde tienen que grabarse en la pantalla los campos o constantes.

Cuando se transfiere un registro a un programa, los campos se disponen en el orden que se haya especificado en las DDS. El orden en el que se visualizan los campos se base en las posiciones de pantalla (números de línea y posición) asignadas al campo dentro de las DDS. El orden en el que se especifican los campos en las DDS y en orden en el que aparecen en la pantalla no tiene que ser el mismo necesariamente.

8.2.2 Indicadores de tecla de función en archivos de dispositivo de pantalla

Los indicadores de tecla de función **KA** a **KN** y **KP** a **KY** son válidos para un programa que contenga un archivo de dispositivo de pantalla WORKSTN si se especifica la tecla de función asociada en las especificaciones de descripción de datos (DDS).

Los indicadores de tecla de función están relacionados con las teclas de función tal como sigue: el indicador de tecla de función **KA** corresponde a la tecla de función 1, **KB** a la tecla de función 2 . . . **KX** a la tecla de función 23, y **KY** a la tecla de función 24.

Las teclas de función se especifican en las DDS con la palabra clave **CFxx** (función de mandato) o **CAxx** (atención de mandato). Por ejemplo, la palabra clave **CF01** autoriza la utilización de la tecla de función 1. Cuando pulse la tecla de función 1, se activa el indicador **KA** en el programa RPG/400. Si especifica la tecla de función como **CF01** (99), se activarán en el programa en RPG/400 el indicador **KA** de tecla de función y el indicador 99. Si el usuario de la estación de trabajo pulsa una tecla de función que no está especificada en las DDS, el sistema OS/400 informa al usuario que ha pulsado una tecla no válida.

Si el usuario de la estación de trabajo pulsa una tecla de función determinada, se activa el indicador de tecla de función asociado en el programa RPG/400 cuando se extrean los campos del registro (lógica de movimiento de campos) y se desactivan todos los otros indicadores de tecla de función. Si no se pulsa ninguna tecla de función, todos los indicadores de tecla de función se desactivan en tiempo de movimiento de campos. Los indicadores de tecla de función se desactivan si el usuario pulsa la tecla Intro.

8.2.3 Teclas de mandato en archivos de dispositivo de pantalla

Puede especificar las teclas de mandatos AYUDA, GIRAR HACIA ARRIBA, GIRAR HACIA ABAJO, IMPR, BORRA e INICIO en las especificaciones de descripción de datos (DDS) para un archivo de dispositivo de pantalla, con las palabras clave **HELP**, **ROLLUP**, **ROLLDOWN**, **PRINT**, **CLEAR**, y **HOME**.

Un programa en RPG/400 puede procesar las teclas de mandato siempre que el compilador del RPG/400 procese una operación **READ** o una operación **EXFMT** en un formato de registro para el que se hayan especificado en las DDS las palabras clave apropiadas. Cuando las teclas de mandato están en vigor y se pulsa una tecla de función, el sistema OS/400 devuelve el control al programa en RPG/400. Si se especifica un indicador de respuesta en las DDS para el mandato seleccionado, se activa dicho indicador y se desactivan todos los demás indicadores de respuesta que estén en vigor para el formato de registro y para el archivo.

Si no se ha especificado un indicador de respuesta en las DDS para una tecla de mandato, sucede lo siguiente:

- Para la tecla IMPR sin *PGM especificado, se efectúa la función de impresión.
- Para las teclas GIRAR HACIA ARRIBA y GIRAR HACIA ABAJO utilizadas con subarchivos, el subarchivo visualizado gira hacia arriba o hacia abajo, dentro del subarchivo. Si se intenta girar más allá del inicio o del final de un subarchivo, se produce un error en tiempo de ejecución.
- Para la tecla IMPR especificada con las teclas *PGM, GIRAR HACIA ARRIBA y GIRAR HACIA ABAJO utilizadas sin subarchivos y para las teclas BORRA, AYUDA e INICIO, se establece uno de los valores 1121-1126 de ***STATUS**, respectivamente, y el proceso continúa.

8.3 Proceso de archivos WORKSTN

Esta sección explica los códigos de operación de archivo válidos para un archivo WORKSTN.

Subtemas

8.3.1 Operación EXFMT

8.3.2 Operación READ

8.3.3 Operación WRITE

8.3.4 Archivo WORKSTN

8.3.5 Subarchivos

8.3.1 Operación EXFMT

La operación **EXFMT** es una combinación de **WRITE** seguida de **READ** para el mismo formato de registro. Si define un archivo WORKSTN en las especificaciones de descripción de archivos como un archivo combinado (**C** en la posición 15), controlado en cálculo (**F** en la posición 16), que utiliza datos descritos externamente en (**E** en la posición 19), puede usarse el código de operación **EXFMT** (ejecutar formato) para grabar y leer en la pantalla.

8.3.2 Operación READ

La operación **READ** es válida para un archivo controlado en cálculo combinado o un archivo controlado en cálculo de entrada que utilice los datos descritos externamente o los datos descritos por el programa. La operación **READ** recupera un registro de la pantalla. No obstante, debe haber un formato en el dispositivo antes que pueda producirse una operación de entrada. Esta necesidad puede satisfacerse en un dispositivo de pantalla condicionando un registro de salida con el indicador **1P**, grabando el primer formato en el dispositivo desde otro programa o, si la lectura se efectúa mediante el nombre de formato de registro, empleando en la descripción de registro en las DDS la palabra clave **INZRCD**.

8.3.3 Operación WRITE

La operación **WRITE** graba un nuevo registro en una pantalla y es válido para un archivo combinado o un archivo de salida. Las especificaciones de salida y la operación **EXCPT** también se pueden utilizar para grabar en un archivo WORKSTN. Consulte el manual *RPG/400 Reference* que incluye una descripción detallada de cada uno de estos códigos de operación.

La Figura 68 muestra los códigos de operación de archivo válidos para un archivo WORKSTN.

Especificaciones de descripción de archivos			Especificaciones de cálculo		
Posiciones			Posiciones		
15	16	28-32			
I	P/S		CLOSE, ACQ, REL, NEXT, POST, FORCE		
I	P/S		WRITE(1), CLOSE, ACQ, REL, NEXT, POST, FORCE		
I	F		READ, OPEN, CLOSE, ACQ, REL, NEXT, POST		
C	F		READ, WRITE(1), EXFMT(2), OPEN, CLOSE, ACQ, REL, NEXT, POST, UPDAT(3), CHAIN(3), READC(3)		
O	Blanco		WRITE(1), OPEN, CLOSE, ACQ, REL, POST		
<p>Nota: (1) La operación WRITE no es válida para un archivo descrito por programa que se utilice con un nombre de formato.</p>					
<p>Nota: (2) Si se utiliza la operación EXFMT, el archivo debe estar descrito externamente (una E en la posición 19 de las especificaciones de descripción de archivos).</p>					
<p>Nota: (3) Para formatos de registro de subarchivo, las operaciones UPDAT, CHAIN y READC también son válidas.</p>					

Figura 68. Códigos de operación de archivo válidos para un archivo WORKSTN

8.3.4 Archivo WORKSTN

La Figura 69 es un diagrama de proceso de los archivos WORKSTN.

IMAGEN 21

Figura 69. Diagrama de proceso para archivos WORKSTN

Operaciones de archivo válidas:

1. **CLOSE, FORCE**
2. **WRITE, CLOSE, FORCE**
3. **READ, OPEN, CLOSE**
4. **READ, WRITE, EXFMT, OPEN, CLOSE**
5. **WRITE, OPEN, CLOSE**
6. **READ, WRITE, OPEN, CLOSE**
7. **OPEN, CLOSE**
8. **READC, CHAIN, WRITE, UPDAT, (validas únicamente para registros definidos como subarchivos)**

Notas:

1. Las posiciones sombreadas deben estar en blanco y las posiciones sin entradas son dependientes del programa.
2. Las operaciones **WRITE** para un archivo descrito en el programa, requieren un nombre de estructura de datos en un campo del resultado; no son válidas las operaciones **WRITE** para un archivo descrito en el programa que use un nombre de formato en las especificaciones de salida.
3. El proceso de subarchivo es válido únicamente para un archivo descrito externamente.

8.3.5 Subarchivos

En las especificaciones de descripción de datos (DDS) pueden especificarse subarchivos de un archivo de dispositivo de pantalla para permitir al usuario manejar múltiples registros del mismo tipo en la pantalla . (Consulte la Figura 70.) Un subarchivo es un grupo de registros que se leen o se graban en un archivo de dispositivo de pantalla. Por ejemplo, un programa lee registros de un archivo de base de datos y crea un subarchivo de registros de salida. Cuando se haya grabado un subarchivo en su totalidad, el programa envía el subarchivo completo a la pantalla en una operación de grabación. El usuario de la estación de trabajo puede alterar los datos o entrar datos adicionales en el subarchivo. El programa lee a continuación todo el subarchivo desde el dispositivo de pantalla, lo transfiere al programa y procesa individualmente cada registro del subarchivo.

Los registros que desee que se incluyan en un subarchivo se especifican en las DDS del archivo. El número de registros que se puede incluir en un subarchivo también debe especificarse en las DDS. Un archivo puede contener más de un subarchivo y pueden estar activados al mismo tiempo hasta 12 subarchivos. Dos subarchivos pueden visualizarse a la vez.

La DDS de un subarchivo están formadas por dos formatos de registro: un formato de registro de subarchivo y un formato de registro de control de subarchivo. El formato de registro de subarchivo contiene la información del campo que se transfiere hacia o desde el archivo de pantalla bajo el control del formato de registro de control del subarchivo. El formato de registro de control del subarchivo hace que tengan lugar las operaciones físicas de lectura, grabación o de control del subarchivo. La Figura 71 muestra un ejemplo de las DDS para un formato de registro de subarchivo y la Figura 72 en el tema 8.3.5.1 muestra un ejemplo para un registro de control del subarchivo.

Para obtener una descripción de cómo utilizar palabras clave de subarchivo, consulte la publicación *DDS Reference*.

```

+-----+
|
|  Búsqueda Nombre Cliente
|
|  Cod. Búsqueda_____
|
|  Número  Nombre                      Calle                      Ciudad                      Prov.
|
|  XXXX    XXXXXXXXXXXXXXXXXXXXXXXX  XXXXXXXXXXXXXXXXXXXXXXXX  XXXXXXXXXXXXXXXXXXXXXXXX  XX
|
+-----+

```

Figura 70. Visualización de un subarchivo

Para utilizar un subarchivo de dispositivo de pantalla en un programa en RPG/400 debe especificar la palabra clave **SFILE** en las posiciones 54 a 59 en una línea de continuación de las especificaciones de descripción de archivos para el archivo WORKSTN. La palabra clave **SFILE** debe especificarse en una línea de continuación independiente. El archivo WORKSTN debe ser un archivo descrito externamente (**E** en posición 19).

Las posiciones 60 a 67 de la línea de continuación deben especificar el nombre del formato de registro del subarchivo (no el formato del registro de control). Las posiciones 47 a 52 deben especificar el nombre del campo que contenga el número relativo de registro que ha de usarse en el proceso del subarchivo.

En un programa en RPG/400, el proceso por número relativo de registro se define como parte de la definición **SFILE**. La definición **SFILE** implica un archivo de actualización controlado en cálculo que tenga **ADD** como subarchivo. Por eso, las operaciones de archivo que son válidas para el subarchivo no dependen de la definición del archivo **WORKSTN** principal. Es decir, el archivo **WORKSTN** puede definirse como un archivo primario o un archivo controlado en cálculo.

Utilice los códigos de operación **CHAIN**, **READC**, **UPDAT**, o **WRITE** con el formato de registro de subarchivo para transferir datos entre el programa y el subarchivo. Utilice los códigos de operación **READ**, **WRITE** o **EXFMT** con el formato de registro de control de subarchivo para transferir datos entre el programa y el dispositivo de pantalla o para procesar operaciones de control de subarchivo.

El proceso de subarchivos sigue las reglas para el proceso por número relativo de registro. El programa en RPG/400 coloca el número relativo de registro de cualquier registro recuperado mediante una operación **READC** en el campo denominado en las posiciones 47 a 52 de la línea de continuación de las especificaciones de descripción de archivos **SFILE**. Este campo se utiliza también para especificar el número de registro que el programa en RPG/400 utiliza en las operaciones **WRITE** del subarchivo o en las operaciones de salida que utilizan **ADD**. El número de campo especificado en las posiciones 47 a 52 debe estar definido como numérico con cero posiciones decimales. El campo debe tener suficientes posiciones para contener el número de registro más grande del archivo. (Consulte la palabra clave **SFLSIZ** en la publicación *DDS Reference*.) El código de operación **WRITE** y la especificación **ADD** de las especificaciones de salida requieren que se especifique un campo de número relativo de registro en las posiciones 47 a 52 de la línea de continuación de las especificaciones de descripción de archivos **SFILE**.

Si un archivo **WORKSTN** tiene asociado un subarchivo, todas las operaciones de entrada implícitas y todas las operaciones de cálculo explícitas que hagan referencia al nombre del archivo tienen lugar para el archivo **WORKSTN** principal. Las operaciones que especifiquen un nombre de formato de registro que no esté designado como un subarchivo se procesan en el archivo **WORKSTN** principal.

Si pulsa una tecla de función determinada durante la lectura de un registro que no es de subarchivo, las lecturas posteriores de un registro de subarchivo harán que el indicador de tecla de función correspondiente se vuelva a activar aunque el indicador de tecla de función se haya desactivado entre las lecturas. Esto continuará hasta que se lea un registro que no sea de un subarchivo del archivo **WORKSTN**.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++*****
A** BUSQUEDA DE NOMBRE DE CLIENTE
A
A          R SUBARC          REF(REFDES)  1
A          SFL                2
A          TEXT('Registro Subarchivo')
A          CLIE      R          7  3
A          NOMB      R          7 10
A          DIRE      R          7 32      3
A          CDAD      R          7 54
A          PROV      R          7 77
A*
```

Figura 71. Especificaciones de descripción de datos para un formato de registro de subarchivo

Las especificaciones de descripción de datos (DDS) para un formato de registro de subarchivo describen los registros del subarchivo:

- 1 Los atributos de los campos en el formato de registro están contenidos en el archivo de referencias de campos **REFDES** tal como se especifica mediante la palabra clave **REF**.
- 2 La palabra clave **SFL** identifica el formato de registro como un subarchivo.
- 3 Las entradas de línea y posición definen la situación de los campos en la pantalla.

Subtemas

8.3.5.1 Utilización de subarchivos

8.3.5.1 Utilización de subarchivos

Algunos usos clásicos de subarchivos incluyen:

- Sólo visualización. El usuario de la estación de trabajo revisa la visualización.
- Visualización con selección. El usuario solicita más información sobre uno de los elementos de la pantalla.
- Modificación. El usuario modifica uno o más de los registros.
- Sólo entrada, sin comprobación de validez. Se emplea un subarchivo para una función de entrada de datos.
- Sólo entrada, con comprobación de validez. Se emplea un subarchivo para una función de entrada de datos, pero se comprueban los registros.
- Combinación de tareas. Puede emplearse un subarchivo como una pantalla con modificación, más la entrada de nuevos registros.

La figura siguiente muestra un ejemplo de especificaciones de descripción de datos para un formato de registro de control de subarchivo. Para ver un ejemplo de la utilización de un subarchivo en un programa en RPG/400, consulte "Ejemplos de archivo WORKSTN" en el tema 8.6.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++*****
A          R ARCTRL                      SFLCTL (SUBARC)
A N70                      SFLCLR
A 70                      SFLDSPCTL
A 71                      SFLDSP
A                          SFLSIZ (15)
A                          SFLPAG (15)
A                          TEXT('Reg. Control Subarchivo')
A                          OVERLAY
A 71                      ROLLUP (97 'Continúa Búsqueda')
A                          CA01 (98 'Fin del Programa')
A                          HELP (99 'Tecla AYUDA')
A                          1 2'Búsqueda Nombre Cliente'
A                          3 2'Código Búsqueda'
A          CODBUS      R          I 3 14PUTRETAIN
A                          5 2'Numero'
A                          5 10'Nombre'
A                          5 32'Calle'
A                          5 54'Ciudad'
A                          5 76'Provincia'
A*

```

Figura 72. Especificaciones de descripción de datos (DDS) para un formato de registro de control de subarchivo

El formato de registro de control de subarchivo define los atributos del subarchivo, el campo de entrada para efectuar la búsqueda, las constantes y las teclas de función. Las palabras clave que puede utilizar indican lo siguiente:

- SFLCTL** indica el nombre del subarchivo asociado (**SUBARC**).
- SFLCLR** indica cuándo deberá borrarse el subarchivo (cuando el indicador 70 esté desactivado).
- SFLDSPCTL** indica cuándo tiene que visualizarse el registro de control del subarchivo (cuando el indicador 70 esté activado).
- SFLDSP** indica cuándo tiene que visualizarse el subarchivo (cuando el indicador 71 esté activado).
- SFLSIZ** indica el número total de registros a incluir en el subarchivo (15).
- SFLPAG** indica el número total de registros por página (15).
- ROLLUP** indica que el indicador 97 se activa en el programa cuando el usuario pulsa la tecla de Giro Hacia Arriba.
- HELP** permite al usuario pulsar la tecla AYUDA para que se visualice un mensaje que describe las teclas de mandato válidas.
- PUTRETAIN** permite mantener en la pantalla el valor que se ha entrado

en el campo **CODBUS**.

Además de la información de control, el formato de registro de control del subarchivo define también las constantes a utilizar como encabezamiento de columnas para el formato de registro del subarchivo.

8.4 Archivo WORKSTN descrito por programa

Un archivo WORKSTN descrito por el programa puede usarse con un nombre de formato especificado en las especificaciones de salida o sin ningún nombre de formato en las especificaciones de salida. El nombre del formato, si se especifica, hace referencia al nombre de un formato de registro de las especificaciones de descripción de datos. Este formato describe:

- Cómo se le da formato en la pantalla al flujo de datos enviado desde un programa en RPG/400.
- Cuáles son los datos enviados.
- Cuáles son las funciones ICF a realizar.

Si se utiliza un nombre de formato, deben emplearse las especificaciones de entrada y de salida para describir los registros de entrada y salida.

Para un archivo WORKSTN descrito por el programa puede especificar la opción **PASS** en la línea de continuación de las especificaciones de descripción de archivos. Las posiciones 60 a 65 deben contener ***NOIND**. **PASS *NOIND** indica que el programa en RPG/400 no transfiere adicionalmente indicadores a la gestión de datos durante la salida ni los recibirá durante la entrada. Es responsabilidad del usuario la transferencia de los indicadores describiéndolos como campos (en la forma ***INxx**, ***IN**, o ***IN,x**) en el registro de entrada o de salida. Deben especificarse en la secuencia requerida por las especificaciones de descripción de datos (DDS). Puede utilizar el listado de las DDS para determinar dicha secuencia.

Subtemas

- 8.4.1 Archivo WORKSTN descrito por programa con un nombre de formato
- 8.4.2 Archivo WORKSTN descrito por programa sin un nombre de formato

8.4.1 Archivo *WORKSTN* descrito por programa con un nombre de formato

Las siguientes especificaciones son aplicables a la utilización de un nombre de formato para un archivo *WORKSTN* descrito por programa.

Subtemas

- 8.4.1.1 Especificaciones de salida
- 8.4.1.2 Especificaciones de entrada
- 8.4.1.3 Especificaciones de cálculo
- 8.4.1.4 Consideraciones adicionales

8.4.1.1 Especificaciones de salida

En las especificaciones de salida debe especificar el nombre del archivo WORKSTN en las posiciones 7 a 14. El nombre del formato, que es el nombre del formato de registro en las DDS, se especifica como un literal en las posiciones 45 a 54 en la línea de descripción del campo siguiente. K1 a K8 deben especificarse (ajustados por la derecha) en las posiciones 40 a 43 de la línea que contenga el nombre del formato. La K que identifica la entrada es una longitud en lugar de una posición final y el número indica la longitud del nombre del formato. Por ejemplo, si el nombre del formato es **AYUCLI**, la entrada en las posiciones 40 a 43 será K6. (Se admiten ceros a la izquierda a continuación de la K). No puede condicionarse el nombre del formato (no son válidos los indicadores en las posiciones 23 a 31).

Los campos de salida deben encontrarse en el registro de salida en el mismo orden en que están definidos en las DDS. Sin embargo, los nombres de campo no tienen que ser los mismos. Las entradas de posición final de los campos hacen referencia a la posición del registro de salida que se transfiere del programa en RPG/400 a la gestión de datos, no a la ubicación de los campos en la pantalla.

Para transferir indicadores de la salida, efectúe una de las siguientes acciones:

- Especifique la palabra clave **INDARA** en las DDS para el archivo WORKSTN. No utilice la opción **PASS *NOIND** en la línea de continuación de las especificaciones de archivos y no especifique los indicadores en las especificaciones de salida. El programa y el archivo usan un área de indicadores separada para transferir los indicadores.
- Especifique la opción **PASS *NOIND** en la línea de continuación de las especificaciones de archivos. Especifique los indicadores en las especificaciones de salida como campos en la forma ***INxx**. Los campos de indicador deben preceder a otros campos en el registro de salida y deben aparecer en el orden especificado por las DDS del archivo WORKSTN. Este orden puede determinarse a partir del listado de las DDS.

8.4.1.2 Especificaciones de entrada

Las especificaciones de entrada describen el registro que el programa en RPG/400 recibe desde la pantalla o dispositivo ICF. El nombre del archivo WORKSTN debe especificarse en las posiciones 7 a 14. Los campos de entrada deben estar ubicados en el registro de entrada en la misma secuencia en que estén definidos en las DDS. Sin embargo, los nombres de campo no necesitan ser los mismos. Las entradas de ubicación de campo hacen referencia a los campos del registro de entrada.

Para recibir indicadores en la entrada, efectúe una de las siguientes acciones:

- Especifique la palabra clave **INDARA** en las DDS para el archivo WORKSTN. No utilice la opción **PASS *NOIND** en la línea de continuación de las especificaciones de archivos y no especifique los indicadores en las especificaciones de entrada. El programa y el archivo usan un área de indicadores separada para transferir los indicadores.
- Especifique la opción **PASS *NOIND** en la línea de continuación de las especificaciones de archivos. Especifique los indicadores en las especificaciones de entrada como campos en la forma ***INxx**. Los campos de indicador deben aparecer en el orden especificado por las DDS del archivo WORKSTN. Este orden puede determinarse a partir del listado de las DDS.

Debería asignarse un indicador de identificación de registros a cada registro del archivo para identificar el registro que se ha leído del archivo WORKSTN. Puede especificarse en las DDS un campo oculto con un valor por omisión para el código de identificación de registros.

8.4.1.3 Especificaciones de cálculo

El código de operación **READ** es válido para un archivo WORKSTN descrito por programa que esté definido como un archivo de control en cálculo combinado. Consulte la Figura 68 en el tema 8.3.3. El nombre de archivo debe especificarse en el factor 2 de esta operación. Debe existir un formato en el dispositivo antes que pueda producirse una operación de entrada. Esta necesidad puede satisfacerse en un dispositivo de pantalla condicionando un registro de salida con el indicador **1P**, grabando el primer formato en el dispositivo desde otro programa o, si la lectura se efectúa mediante el nombre de formato de registro, empleando en la descripción de registro en las DDS la palabra clave **INZRCD**. La operación **EXFMT** no es válida para un archivo WORKSTN descrito por programa. También puede utilizar la operación **EXCPT** para grabar en un archivo WORKSTN.

8.4.1.4 Consideraciones adicionales

Cuando se utiliza un nombre de un formato con un archivo WORKSTN descrito por el programa, debe considerarse además lo siguiente:

- El nombre especificado en las posiciones 45 a 54 de las especificaciones de salida se supone que es el nombre de un formato de registro en las DDS que se han empleado para crear el archivo.
- Si hay una especificación Kn para un registro de salida, ésta debe usarse también para todos los demás registros de salida para ese archivo. Si no se emplea una especificación Kn para todos los registros de salida a un archivo, se producirá un error en tiempo de ejecución.

Como ejemplo de utilización de un nombre de formato con un archivo WORKSTN de dispositivo de pantalla descrito en el programa, consulte "Programa de ejemplo 6-Archivo WORKSTN descrito por programa con nombre de formato FORMAT en las especificaciones de salida" en el tema 8.6.6.

8.4.2 Archivo WORKSTN descrito por programa sin un nombre de formato

Cuando no se utiliza un nombre de formato de registro, un archivo de dispositivo de pantalla descrito en el programa describe un archivo que contiene una descripción de formato de registro con un campo. Los campos en el registro deben describirse dentro del programa que utiliza el archivo.

Cuando crea el archivo de pantalla mediante el mandato Crear archivo de pantalla, el archivo tiene los atributos siguientes:

- Puede especificarse una longitud de registro variable; por lo tanto, la longitud real del registro debe especificarse en el programa que lo utiliza. (La longitud de registro máxima es el tamaño de la pantalla menos uno).
- No se transfieren indicadores ni desde ni hacia el programa.
- No se definen indicadores de teclas de función.
- El registro se graba en la pantalla comenzando en la posición 2 de la primera línea disponible.

Subtemas

8.4.2.1 Archivo de entrada

8.4.2.2 Archivo de salida

8.4.2.3 Archivo combinado

8.4.2.1 Archivo de entrada

En un archivo de entrada, el registro de entrada, que el soporte de dispositivo del RPG/400 lo trata como a un campo de entrada único, se inicializa a blancos cuando se abre el archivo. El cursor se sitúa al principio del campo, que es la posición 2 de la pantalla.

8.4.2.2 Archivo de salida

Para un archivo de salida, el soporte del dispositivo del OS/400 trata el registro de salida como una serie de caracteres que se ha de enviar a la pantalla. Cada registro de salida se graba como el siguiente registros secuencial del archivo; es decir, cada registro visualizado se sobrepone al registro visualizado anteriormente.

8.4.2.3 Archivo combinado

Para un archivo combinado, el registro, que el soporte de dispositivo del OS/400 trata un campo único, aparece en la pantalla y es tanto el registro de salida como el registro de entrada. El soporte del dispositivo inicializa el registro de entrada a blancos, y el cursor se coloca en la posición 2.

Para obtener más información sobre archivos de dispositivo de pantalla descritos por programa, consulte la publicación *Guía para la Gestión de Datos*.

8.5 Archivos de múltiples dispositivos

Cualquier archivo WORKSTN en RPG/400 que tenga especificada por lo menos una de las palabras clave **ID**, **IND**, **NUM**, o **SAVDS** (en la línea de continuación de las especificaciones de archivos) es un archivo de múltiples dispositivos. Mediante un archivo de múltiples dispositivos, el programa puede acceder a más de un dispositivo.

El programa en RPG/400 accede a los dispositivos a través de los dispositivos del programa, los cuales son un mecanismo simbólico para dirigir las operaciones a un dispositivo real. Cuando el usuario crea un archivo (utilizando las especificaciones de las DDS y mandatos tales como los de creación de archivo), debe considerar cosas tales como qué dispositivo está asociado con un dispositivo de programa, si un archivo tiene o no un dispositivo solicitante, qué formatos de registro habrán de utilizarse para pedir a los dispositivos que respondan a una operación **READ** mediante nombre de archivo y cuanto tiempo habrá de esperar una respuesta a esta operación de **READ**. Para una información detallada de las opciones y requisitos para crear un archivo de múltiples dispositivos, consulte los capítulos sobre archivos de pantalla en la publicación *Guía para la Gestión de Datos* y la información sobre archivos ICF en la publicación *ICF Programmer's Guide* además de los manuales a los que se hace referencia en estas dos publicaciones.

Con archivos de múltiples dispositivos se hace un uso particular de los siguientes códigos de operación:

- Además de abrir un archivo, la operación **OPEN** puede adquirir (como máximo) un dispositivo para un archivo de múltiples dispositivos. Al crear el archivo debe especificar el dispositivo.
- La operación **ACQ** (adquirir) adquiere cualquier otro dispositivo para el archivo.
- La operación **REL** (liberar) libera un dispositivo del archivo.
- La operación **WRITE**, cuando se utiliza con la palabra clave de las DDS **INVITE**, pide a un dispositivo de programa que responda a las siguientes operaciones de leer de los dispositivos de programa solicitados. Consulte la sección sobre la petición a un dispositivo de programa en las publicaciones *ICF Programmer's Guide* y *Guía para la Gestión de Datos*.
- La operación **READ** efectúa tanto una operación de leer desde los dispositivos de programa solicitados como una operación de leer desde un dispositivo de programa. Cuando no hay ninguna operación **NEXT** en vigor, una operación de lectura en ciclo de programa o una operación **READ** por nombre de archivo, esperará la entrada desde cualquiera de los dispositivos a los que se les haya pedido responder (leer desde dispositivo de programa solicitado). Otras operaciones de entrada y de salida, incluyendo una **READ** por nombre de archivo después de una operación **NEXT** y de una **READ** por nombre de formato, efectúa una operación de leer desde un dispositivo de programa utilizando el dispositivo de programa indicado en un campo especial. (El campo se nombra en la entrada **ID** de las líneas de continuación de las especificaciones de archivo).

Este dispositivo puede ser utilizado en la última operación de entrada, en un dispositivo que especifique el usuario o en el dispositivo solicitante. Consulte las secciones sobre lectura desde dispositivos de programa solicitados y sobre lectura desde un dispositivo de programa en las publicaciones *ICF Programmer's Guide* y *Guía para la Gestión de Datos*.

- La operación **NEXT** especifica el dispositivo que ha de utilizarse en la siguiente operación **READ** por nombre de archivo o en la siguiente operación de leer en ciclo de programa.
- La operación **POST** coloca información en la estructura de datos de información INFDS. La información puede ser sobre un dispositivo específico o sobre el archivo. (La operación **POST** no está limitada a ser utilizada con archivos de múltiples dispositivos).

En la publicación *RPG/400 Reference* hallará detalles sobre los códigos de operación del RPG/400.

En la línea de continuación de las especificaciones de descripción de archivos, puede especificar varias opciones para controlar el proceso de archivos de múltiples dispositivos.

- La entrada **NUM** indica el número máximo de dispositivos que pueden adquirirse para un archivo.

Utilizando un valor 1 para **NUM**, es posible obtener funciones correspondientes a un archivo de múltiples dispositivos, para un archivo que tenga sólo un dispositivo. Por ejemplo, la Figura 107 en el tema 8.6.8 ilustra la utilización de la función de tiempo de espera excedido de la operación **READ** para un archivo de múltiples dispositivos

- La entrada **ID** especifica el nombre de un campo. El campo puede contener el nombre de un dispositivo de programa al que dirigir algunas operaciones de entrada y salida.

Cuando se emite una operación de lectura desde dispositivo de programa o **WRITE**, el dispositivo utilizado para la operación es el dispositivo identificado por el campo especificado en la entrada **ID**. Este campo se inicializa a blancos y se actualiza con el nombre del dispositivo desde el cual se ha producido la última operación satisfactoria de entrada. También puede establecerse explícitamente trasladando un valor al mismo. El código de operación **ACQ** no afecta al valor de este campo. Si no hay ninguna entrada, la operación se realiza en el dispositivo desde el cual se ha producido la última operación satisfactoria de entrada. Se utiliza un nombre de dispositivo en blanco si aún no se ha producido con éxito ninguna operación de lectura desde un dispositivo.

Cuando se emite una operación de lectura desde un dispositivo de programa o **WRITE** con un nombre de dispositivo en blanco, el compilador RPG/400 utiliza implícitamente el nombre de dispositivo del dispositivo peticionario para el programa. Si el usuario llama un programa en RPG/400 interactivamente y adquiere un dispositivo ICF en el cual desea realizar una de estas operaciones, debe trasladar explícitamente el nombre del dispositivo ICF al nombre de campo especificado con la entrada **ID** antes de efectuar la operación. Si no se hace esto, el nombre de dispositivo utilizado estará en blanco (en cuyo caso se utiliza el nombre de dispositivo peticionario interactivo) o el nombre de dispositivo utilizado será el de la última operación satisfactoria de entrada. Una vez que el usuario haya realizado una operación de E/S en el dispositivo ICF, no necesita volver a modificar el valor a menos que se complete con éxito una operación de entrada con un dispositivo distinto.

- La entrada **SAVDS** indica una estructura de datos que se salva y se restaura por cada dispositivo que se adquiere para un archivo. La entrada **IND** indica un grupo de indicadores que han de salvarse y restaurarse por cada dispositivo adquirido para un archivo. Antes de una operación de entrada, se salvan el grupo en curso de indicadores y la estructura de los datos. Después de la operación de entrada, el compilador del RPG/400 restaura los indicadores y la estructura de datos para el dispositivo asociado con la operación. El grupo de indicadores o la estructura de datos pueden ser diferentes de los que estaban disponibles antes de la operación de entrada.
- La entrada **INFDS** especifica la estructura de datos de información de archivo para el archivo WORKSTN. Puede accederse al campo RPG/400 *STATUS y al código de retorno mayor/menor para la operación de E/S mediante esta estructura de datos. En particular cuando se utiliza ICF, los dos campos son útiles para la detección de los errores que se han producido durante las operaciones de E/S para archivos de múltiples dispositivos.

Nota: Cuando se especifiquen estas opciones de control, se debe codificar la opción **NUM** antes de las opciones **ID**, **IND** o **SAVDS**.

8.6 Ejemplos de archivo WORKSTN

Esta sección ilustra algunas aplicaciones de estaciones de trabajo y su codificación en RPG/400.

- "Programa de ejemplo 1-Consulta" en el tema 8.6.1 es un ejemplo de un programa básico de consulta que utiliza el archivo **WORKSTN** en el compilador del RPG/400.
- "Programa de ejemplo 2-Entrada de datos con actualización del maestro" en el tema 8.6.2 es un ejemplo de una entrada de datos con un programa de actualización del maestro.
- "Programa de ejemplo 3-Mantenimiento" en el tema 8.6.3 es un ejemplo de un programa de mantenimiento.
- "Programa de ejemplo 4-Proceso de subarchivo WORKSTN" en el tema 8.6.4 es un ejemplo de proceso de subarchivo **WORKSTN**.
- "Programa de ejemplo 5-Consulta por código postal y búsqueda por nombre" en el tema 8.6.5 es un ejemplo de un programa interactivo en el que se produce la búsqueda de un campo de nombre cuando el usuario de la estación de trabajo entra un distrito postal y una serie de búsqueda en respuesta a la primera pantalla mostrada por el programa. Este programa muestra una técnica para resolver los problemas típicos al intentar identificar un cliente y determinar el número de cliente correcto. En este ejemplo el usuario conoce el distrito postal y algún dato acerca del nombre del cliente, como por ejemplo el número de caracteres que constituyen el nombre.
- "Programa de ejemplo 6-Archivo WORKSTN descrito por programa con nombre de formato FORMAT en las especificaciones de salida" en el tema 8.6.6 es un ejemplo de un archivo **WORKSTN** descrito por el programa con un nombre de formato en las especificaciones de salida.
- "Programa de ejemplo 7-Línea de comienzo variable" en el tema 8.6.7 es un ejemplo de utilización de línea de comienzo variable para determinar dónde aparecerá en una pantalla un formato de registro.
- "Programa de ejemplo 8-Operación de lectura con límite de tiempo de espera excedido" en el tema 8.6.8 muestra como utilizar **READ** con un límite de tiempo excedido.

Subtemas

- 8.6.1 Programa de ejemplo 1-Consulta
- 8.6.2 Programa de ejemplo 2-Entrada de datos con actualización del maestro
- 8.6.3 Programa de ejemplo 3-Mantenimiento
- 8.6.4 Programa de ejemplo 4-Proceso de subarchivo WORKSTN
- 8.6.5 Programa de ejemplo 5-Consulta por código postal y búsqueda por nombre
- 8.6.6 Programa de ejemplo 6-Archivo WORKSTN descrito por programa con nombre de formato FORMAT en las especificaciones de salida
- 8.6.7 Programa de ejemplo 7-Línea de comienzo variable
- 8.6.8 Programa de ejemplo 8-Operación de lectura con límite de tiempo de espera excedido

8.6.1 Programa de ejemplo 1-Consulta

Las figuras siguientes ilustran un programa de consulta sencillo utilizando el archivo **WORKSTN** :

Figura	Contenido
Figura 73 a continuación y Figura 74	Especificaciones DDS para un archivo de base de datos y un archivo de dispositivo de pantalla
Figura 75	Especificaciones de descripción de archivo y de cálculo
Figura 76	Pantalla de solicitud
Figura 77	Pantalla de información al cliente

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A.....T.Nombre++++Lon++TDpB.....Funciones+++++*****
A* ARCHIVO MAESTRO CLIENTES -- CUSMSTP
A      R CUSREC
A      CUST          5          TEXT('NUMERO CLIENTE')
A      NAME          20          TEXT('NOMBRE CLIENTE')
A      ADDR          20          TEXT('CALLE CLIENTE')
A      CITY          20          TEXT('CIUDAD CLIENTE')
A      STATE         2          TEXT('ABREVIATURA PROVINCIA')
A      ZIP            5 0        TEXT('CODIGO POSTAL')
A      SRHCOD        3          TEXT('CODIGO BUSQ. NOMBRE
                                CLIENTE')
A      CUSTYP         1          TEXT('TIPO CLIENTE')
A      ARBAL         10 2        TEXT('SALDO C/D')
A*****
A* NOMB ARCHIV: CUSMSTL *
A* DESCRIPCION: VISTA LOGICA DEL ARCHIVO MAESTRO DE CLIENTES *
A*              (CUSMSTP) POR NUMERO DE CLIENTE (CUST) *
A*****
A.....T.Nombre++++Lon++TDpB.....Funciones+++++*****
A      R CUSREC          PFILE (CUSMSTP)
A      K CUST
```

Figura 73. DDS para el archivo CUSMSTP del programa de consulta WORKSTN

Las especificaciones de descripción de datos (DDS) para el archivo de base de datos utilizado por este programa describe un formato de registro **CUSREC**. El archivo lógico **CUSMSTL** con clave en función del número de cliente está basado en el archivo físico **CUSMSTP**, tal como indica la palabra clave **PFILE**. Cada campo del formato de registro está definido en el archivo físico **CUSMSTP**.

Nota: Normalmente los atributos de los campos, tales como número de posiciones decimales y tipo de datos, se definen en un archivo de referencias de campos en lugar de en las DDS para el formato de registro. Los atributos se muestran en las DDS para que pueda observar cuales son.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A* NOMB ARCHIV: CUSFMT *
A* DESCRIPCION: ARCHIVO DE PANTALLA PARA CONSULTA MAESTRO CLIENTES *
A*****
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++*****
A      DSPSIZ (24 80 *DS3)
A      CHGINPDFT (CS)
A      CA03 (15 'FIN DE TRABAJO')
A      PRINT
A      INDARA
A      R CUSHDG
A      OVERLAY
A      2 4TIME
A      DSPATR (HI)
A      2 29'Consulta Maestro Clientes'
A      DSPATR (HI)
A      DSPATR (UL)
A      2 70DATE
A      EDTCDE (Y)
A      DSPATR (HI)
A      R CUSFTG
```

```

A          23 20'INTRO - Continuar'
A          DSPATR(HI)
A          23 49'F3 - Fin de trabajo'
A          DSPATR(HI)
A          R CUSPMT
A          OVERLAY
A          CUST          5A I 10 50DSPATR(HI)
A          DSPATR(CS)
A 99          ERRMSG('No encontrado Número
A          de Cliente' 99)
A          10 26'Entrar Número de Cliente:'
A          DSPATR(HI)

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre++++RLon++TdPBLínPosFunciones+++++*****
A          R CUSFLDS
A          OVERLAY
A          8 25'Nombre'
A          NAME          20A O 8 35DSPATR(HI)
A          10 25'Calle'
A          ADDR          20A O 10 35DSPATR(HI)
A          12 25'Ciudad'
A          CITY          20A O 12 35DSPATR(HI)
A          14 25'Prov.'
A          STATE         2A O 14 35DSPATR(HI)
A          14 41'Distrito postal'
A          ZIP           5S 00 14 50DSPATR(HI)
A          16 25'Saldo C/D'
A          ARBAL         10Y 20 16 42DSPATR(HI)
A          EDTCDE(J)
A          6 25'Cliente'
A          CUST          5A O 6 35DSPATR(HI)

```

Figura 74. DDS para el archivo de dispositivo de pantalla CUSFMT del programa de consulta WORKSTN

Las especificaciones de descripción de datos (DDS) para el archivo del dispositivo de pantalla **CUSFMT** que va a utilizar este programa especifican las entradas a nivel de archivo y describen cuatro formatos de registro: **CUSHDG**, **CUSFTG**, **CUSPMT**, y **CUSFLDS**.

Las entradas a nivel de archivo definen el tamaño de la pantalla (**DSPSIZ**), los valores de entrada por omisión (**CHGINPDFT**), la tecla de atención de mandato utilizada para finalizar el programa, la tecla de impresión (**PRINT**) y un indicador de área separado (**INDARA**).

El formato de registro **CUSHDG** contiene la constante '**Consulta Maestro Cliente**', que identifica la pantalla. También contiene las palabras clave **TIME** y **DATE**, que visualizarán la fecha y la hora actuales en la pantalla.

El formato de registro **CUSFTG** contiene las constantes '**INTRO - Continuar**' y '**F3 - Fin Trabajo**', las cuales describen las opciones de proceso.

El formato de registro **CUSPMT** contiene la solicitud "Entrar Número de Cliente:" y el campo de entrada **CUST** en el que el usuario de la estación de trabajo entra el número de cliente. Los separadores de columna definen el campo de entrada en la pantalla en donde el usuario entrará el número de cliente. También se incluye el mensaje de error "No Encontrado Numero de Cliente" en este formato de registro. Este mensaje se visualiza si el programa activa el indicador 99.

El formato de registro **CUSFLDS** contiene las constantes '**Nombre**', '**Calle**', '**Ciudad**', '**Prov.**', '**Distrito Postal**', '**Saldo C/D**' y '**Cliente**' que identifican los campos a representarse en la pantalla desde el programa. Este formato de registro describe también los campos que corresponden a estas constantes. Todos estos campos se describen como campos de salida ya que los rellena el programa; el usuario no entra ningún dato en estos campos. Para entrar otro número de cliente el usuario pulsa Intro en respuesta a este registro.

Además de describir las constantes, los campos y los atributos para la pantalla, los formatos de registro definen también los atributos de pantalla para las constantes y los campos, los números de línea y las posiciones horizontales en donde se visualizarán las constantes y los campos.

Tenga en cuenta que al utilizar la palabra clave **OVERLAY**; los formatos de registro **CUSHDG**, **CUSPMT** y **CUSFLDS** recubrirán el formato de registro **CUSFTG**. El formato **CUSFTG** permanecerá en la pantalla cuando cualquiera de los demás formatos se represente en la pantalla.

Nota: Normalmente los atributos de campo se definen en el archivo de referencias de campo en lugar de en las DDS para un archivo. Sin embargo

se muestran aquí para que pueda ver los atributos de los campos.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*****
F*   ID PROGRAMA   - CUSTINQ                               *
F*   NOMB PROGRAMA - CONSULTA MAESTRO CLIENTES           *
F*****
FNombarchIPEAF....LregLC1AIDePcLaEDispos+.....KSalidaEntradA....U1.*
FCUSMSTL IF E           K           DISK
FCUSFMT CF E           WORKSTN

CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C           *IN15       DOWEQ'0'
C*
C* GRABAR CABECERA Y PIE EXCEPTO SI SE PRODUCE UN ERROR Y SOLICITAR
C* NUMERO DE CLIENTE
C           *IN99       CASEQ'0'       HEADNG
C           END
C           EXFMTCUSPMT
C* SI NO ES FIN DE TRABAJO Y ES UN NUMERO DE CLIENTE VALIDO
C* VISUALIZAR INFORMACION DEL CLIENTE
C           *IN15       IFEQ '0'
C           CUST       CHAINCUSREC           99
C           *IN99       IFEQ '0'
C           EXFMTCUSFLDS
C           END IF
C           END IF
C           END DO
C           MOVE '1'       *INLR
C*****
C*   SUBROUTINA   - HEADING                               *
C*   PROPOSITO   - VISUALIZAR CABECERA Y PIE             *
C*****
C           HEADNG       BEGSR
C           WRITECUSFTG
C           WRITECUSHDG
C           ENDSR

```

Figura 75. Especificaciones de descripción de archivo y cálculos para el programa de consulta WORKSTN

Para este programa sólo se requieren las especificaciones de descripción de archivo y las especificaciones de cálculo del RPG/400. No se necesitan especificaciones de entrada y salida ya que ambos archivos son archivos descritos externamente (según se indica mediante la entrada **E** en la posición 19). Ambos archivos se describen como archivos controlados en cálculo, según se indica mediante la **F** entrada en la posición 16 ya que las operaciones de entrada/salida se controlan mediante códigos de operación especificados por el programador. Además la **K** en la posición 31 de las especificaciones de descripción de archivos para el archivo **CUSMSTL** indica que el archivo se procesa mediante claves.

La operación **DOWEQ** realizará un bucle hasta que el usuario pulse la tecla F3 para finalizar el trabajo. La tecla F3 activa el indicador 15, tal como se define en las DDS. Si el indicador 15 está activado, finaliza el bucle, se activa el indicador **LR** y finaliza el programa.

La operación **CASEQ** realiza la subrutina **HEADNG** la cual representa la cabecera y los pies en la pantalla. Las cabeceras y los pies no se representan en la pantalla cuando se ha producido un error.

La operación **EXFMT** representa en la pantalla el registro **CUSPMT**. Este registro solicita del usuario que entre un número de cliente. Si el usuario entra un número de cliente y pulsa Intro, la misma operación **EXFMT** lee a continuación el registro entrándolo en el programa.

Si el usuario no finaliza el trabajo, la operación **CHAIN** recupera un registro del archivo **CUSMSTL**. Observe que el nombre del formato de registro **CUSREC** se especifica para la operación **CHAIN** en lugar del nombre del archivo. Si no se encuentra el registro, se activa el indicador 99 y el programa inicia el bucle de retorno para visualizar de nuevo el registro **CUSPMT**. Se visualiza el mensaje **No Encontrado Numero de Cliente**, la palabra clave **ERRMSG** en las DDS está condicionada por el indicador 99 y se bloquea el teclado. El usuario debe pulsar la tecla Restaurar en respuesta a este mensaje para restaurar el teclado. A continuación el usuario puede entrar otro número de cliente.

Si la operación **CHAIN** recupera un registro del archivo **CUSMSTL**, la operación **EXFMT** representa el registro **CUSFLDS** en la pantalla de la estación de trabajo. Este registro contiene el nombre del cliente, su dirección y el saldo de las cuentas a cobrar.

A continuación el usuario pulsa Intro y el programa inicia el bucle de retorno a la operación **EXFMT** y representa el registro **CUSPMT** en la pantalla de la estación de trabajo. El usuario puede entrar otro número de cliente o finalizar el programa.

La Figura 76 muestra la pantalla inicial representada en la pantalla **WORKSTN** por **EXFMT**.

```
+-----+
| 10:06:31                Consulta Maestro Clientes                01/25/94 |
|                                                                     |
|                               Entre número cliente:  ___A1        |
|                                                                     |
|                               INTRO - Continuar                    F3 - Fin de trabajo |
+-----+
```

Figura 76. Pantalla de Solicitud de Consulta de Cliente

La siguiente pantalla aparece si se encuentra un registro en el archivo **CUSTOMSTL** con el mismo número de cliente que el usuario haya entrado como respuesta a la primera pantalla:

```
+-----+
| 10:06:31                Consulta Maestro Clientes                01/25/94 |
|                                                                     |
|                               Cliente  ___A1                      |
|                               Nombre   CASAS                     |
|                               Dirección Nicaragua 12              |
|                               Ciudad   Zaragoza                   |
|                               Provincia Zar.  Cód. Pos.50079      |
|                               Saldo C/D      11.111.111,00        |
|                                                                     |
|                               INTRO - Continuar                    F3 - Fin de trabajo |
+-----+
```

Figura 77. Pantalla de Información de Consulta de Cliente

8.6.2 Programa de ejemplo 2-Entrada de datos con actualización del maestro

Las figuras siguientes ilustran un programa de Entrada de Datos que realiza solicitudes al usuario, actualiza un registro maestro y graba un archivo de transacción:

Figura	Contenido
Figura 78 a continuación y Figura 79	Especificaciones DDS para un archivo maestro, un archivo de transacción y un archivo de pantalla
Figura 80	Especificaciones de descripción de archivo y de cálculo
Figura 81	Pantalla de solicitud
Figura 82	Visualización de la información actual
Figura 83	Pantalla actualizada

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A.....T.Nombre++++.Lon++TdPB.....Funciones+++++*****
A* ARCHIVO MAESTRO PIEZAS -- PARTMST
A      R MSTREC
A      PART#          5S 0      TEXT('NUMERO PIEZA')
A      DESCRP        20          TEXT('DESCRIPCION')
A      ISSUE          7S 0      TEXT('CANTIDAD ENVIADA')
A      RECPT          7S 0      TEXT('CANTIDAD RECIBIDA')
A      ONHAND         9S 0      TEXT('CANTIDAD DISPONIBLE')
A      DTLUPD         6S 0      TEXT('FECHA ULTIMA
A                                  ACTUALIZACION')
A      K PART#
A*
A* ARCHIVO TRANSACCIONES PIEZAS -- TRNFIL
A      R TRNREC
A      PARTNO         5S 0      TEXT('NUMERO PIEZA')
A      QTYISS         7S 0      TEXT('CANTIDAD ENVIADA')
A      QTYREC         7S 0      TEXT('CANTIDAD RECIBIDA')
A      DATE           6S 0      TEXT('FECHA ACTUAL')

```

Figura 78. DDS de entrada de datos/Actualización del archivo maestro y del archivo de transacción

Las especificaciones de descripción de datos (DDS) para los archivos de base de datos utilizados por este programa describen dos formatos de registro: **MSTREC** y **TRNREC**. El archivo maestro **PARTMST** es un archivo físico con clave; el archivo de transacción **TRNFIL** es un archivo secuencial.

Nota: Normalmente los atributos de los campos, tales como número de posiciones decimales y tipo de datos, se definen en un archivo de referencias de campos en lugar de en las DDS para el formato de registro. Los atributos se muestran en las DDS para que pueda observar cuales son.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A* NOMB ARCHIV: PRTUPD
A* DESCRIPCION: TRANSACCION Y ACTUALIZACION ARCHIVO MAESTRO *
A*****
AAN01N02N03T.Nombre++++RLon++TdPBLínPosFunciones+++++*****
A      CHGINPDFT(CS)
A      PRINT(QSYSPT)
A      INDARA
A      CA03(03 'FIN DEL TRABAJO')
A      R PROMPT
A      2 4TIME DSPATR(HI)
A      2 28'ENTRADA TRANSACCION PIEZA'
A      DSPATR(HI UL)
A      2 70DATE EDTCDE(Y) DSPATR(HI)
A      6 4'Entrar Número de Pieza'
A      DSPATR(HI)
A      PART#          R      Y  I 6 23REFFLD(PART# PARTMST)
A      DSPATR(CS) CHECK(RB)
A 61      ERRMSG('PIEZA # NO SE ENCUENTRA' +
A      61)
A      23 6'INTRO - Continuar'

```

```

A          DSPATR(HI)
A          23 29'F3 - Fin de Trabajo'
A          DSPATR(HI)
A          R TRNFMT
A          CA12(12 'CANCELAR TRANSACCION')
A          2 4TIME DSPATR(HI)
A          2 28'ENTRADA TRANSACCION PIEZA'
A          DSPATR(HI UL)
A          2 70DATE EDTCDE(Y) DSPATR(HI)
A          6 10'Número de Pieza'
A          DSPATR(HI)

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++*****
A          PART#      R      Y  O  6 23REFFLD(PART# PARTMST)
A          DESCRP    R      O  7 23REFFLD(DESCRP PARTMST)
A          ONHAND    R      Y  O  9 10'Cantidad Disponible'
A          QTYISS    R      Y  B 11 25REFFLD(QTYISS TRNFIL)
A          QTYREC    R      Y  B 13 25REFFLD(QTYREC TRNFIL)
A          CHECK(RB)
A          DSPATR(HI CS)
A          13 10'Cantidad Enviada'
A          CHECK(RB) DSPATR(HI CS)
A          23 6'INTRO - Continuar'
A          DSPATR(HI)
A          23 29'F3 - Fin de Trabajo'
A          DSPATR(HI)
A          23 46'F12 - Cancelar Transacción'
A          DSPATR(HI)
    
```

Figura 79. DDS para entrada de datos/Actualización del archivo del dispositivo de pantalla PRTUPD

Las DDS para el archivo del dispositivo de pantalla **PRTUPD** contienen dos formatos de registro: **PROMPT** y **TRNFMT**. El registro **PROMPT** solicita el número de pieza que va a procesarse. Si no se encuentra la pieza, se visualiza un mensaje de error. El registro **TRNFMT** se utiliza para entrar cantidades enviadas y recibidas. Los campos se definen como entrada/salida (B en la posición 38) y salida (O en la posición 38).

La tecla F3 se ha definido a nivel de archivo y es válida para todos los formatos de registro. La tecla F12 está definida a nivel de registro para el formato de registro **TRNFMT** y no es válida para ningún otro formato.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*****
F*  ID PROGRAMA      - DTAENT                      *
F*  NOMB PROGRAMA   - MANTENIMIENTO DE TRANSACCION *
F*  ESTE PROGRAMA REALIZA LAS FUNCIONES SIGUIENTES: *
F*  - AÑADE NUEVOS REGISTROS DE TRANSACCION AL ARCHIVO TRNFIL *
F*  - ACTUALIZA ARCHIVO MAESTRO DE PIEZAS PARTMST *
F*****
FNombarchIPEAF....LregLC1AIDePcLaEDispos+.....KSalidaEntradA...U1.*
FPARTMST UF  E          K          DISK
FTRNFIL  O  E          K          DISK
FPRTUPD  CF  E          WORKSTN

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C*  LINEA PRINCIPAL *
C*****
CL0N01N02N03Factor1+++OpcdeFactor2+++ResultLenDHHiLoEqComments+++++*
C          EXFMTPROMPT
C*
C          *IN03      DOWEQ'0'
C*
C          PART#      CHAINMSTREC          61
C          *IN61      CASEQ'0'          NXTSCN
C          END
C*
C          *IN03      IFEQ '0'
C          EXFMTPROMPT
C          END
C*
C          END
C          MOVE '1'          *INLR

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
    
```

Programa de ejemplo 2-Entrada de datos con actualización del maestro

```

C*      SUBROUTINA - NXTSCN                                     *
C*      PROPOSITO - AÑADIR TRANSACCIONES DE PIEZAS           *
C*****
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgcComentarios+++*
C      NXTSCN      BEGSR
C      EXFMTRNFMT
C      *IN03      IFEQ '0'
C      *IN12      ANDEQ'0'
C      ADD QTYISS      ISSUE
C      ADD QTYREC      RECPT
C      ADD QTYREC      ONHAND
C      SUB QTYISS      ONHAND
C      Z-ADDUDATE      DTLUPD
C      Z-ADDUDATE      DATE
C      UPDATMSTREC
C      WRITETRNREC
C      Z-ADD*ZERO      QTYISS
C      Z-ADD*ZERO      QTYREC
C      ELSE
C      EXCPTRLS
C      END
C      ENDSR
OMSTREC E      RLS
    
```

Figura 80. Especificación de descripción de archivo y especificación de cálculo para el programa de entrada de datos/Actualización

Este programa (entrada de datos con actualización del maestro) solicita del usuario una transacción, actualiza un registro maestro y graba un registro de transacción.

El programa tiene dos archivos de disco (**PARTMST** y **TRNFIL**) y un archivo **WORKSTN** (**PRTUPD**). El programa comienza solicitando un número de pieza al usuario de la estación de trabajo. El usuario puede pulsar la tecla F3, la cual está asociada con el indicador 03 en las DDS, para finalizar el programa.

La operación **CHAIN** recupera el registro maestro. Si no se encuentra el registro maestro, se visualiza un mensaje de error; de lo contrario, se visualiza el formato de registro **TRNFMT**. El usuario puede pulsar la tecla F12 para cancelar la transacción; se libera el registro maestro y vuelve a visualizarse el formato de registro **PROMPT**. El usuario puede pulsar la tecla F3 para finalizar el programa o puede procesar la transacción. Cuando el usuario pulsa **INTRO** después de entrar las cantidades enviadas y/o recibidas, se actualiza el archivo maestro **PARTMST** con la fecha actual, la nueva cantidad disponible, envíos y recepciones y se añade la transacción al archivo de transacciones **TRNFIL**.

El usuario de la estación de trabajo responde a las solicitudes de la primera pantalla entrando un número de pieza, tal como se muestra en la Figura 81.

```

+-----+
| 10:12:14          ENTRADA TRANSACCION PIEZA          01/25/94 |
|                                                         |
| Entre Número de Pieza  ____1 |
|                                                         |
| INTRO - Continuar          F3 - Fin de trabajo |
+-----+
    
```

Figura 81. Pantalla de solicitud para el programa de entrada de datos/actualización

Como el número de pieza 1 está en el Archivo maestro de clientes, el programa visualiza el siguiente registro de la pieza.

```

10:12:20          ENTRADA TRANSACCION PIEZA          01/25/94

Pieza Número  00001
              ALPHABC

Cantidad Disponible  50
Cantidad Enviada    20
Cantidad Recibida   50

INTRO - Continuar  F3 - Fin del Trabajo  F12 - Cancelar Transacción
    
```

Figura 82. Pantalla TRNFMT

El usuario de la estación de trabajo puede pulsar Intro para continuar o F12 para cancelar la transacción.

```

10:12:38          ENTRADA TRANSACCION PIEZA          01/25/94

Pieza Número  00001
              ALPHABC

Cantidad Disponible  80
Cantidad Enviada    0000000
Cantidad Recibida   0000000

INTRO - Continuar  F3 - Fin del Trabajo  F12 - Cancelar Transacción
    
```

Figura 83. Pantalla TRNFMT

8.6.3 Programa de ejemplo 3-Mantenimiento

Las figuras siguientes ilustran un programa de consulta sencillo utilizando el archivo **WORKSTN**:

Figura	Contenido
Figura 84 a continuación Figura 85	Especificaciones DDS para el archivo maestro y para el archivo del dispositivo de pantalla
Figura 86	Especificaciones de descripción de archivo y de cálculo
Figura 87	Pantalla de solicitud de modalidad de visualización
Figura 88	Pantalla de solicitud de modalidad de adición
Figura 89	Pantalla de solicitud de modalidad de actualización
Figura 90	Pantalla de solicitud de modalidad de supresión

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A.....T.Nombre++++.Lon++TdpB.....Funciones+++++*****
A* ARCHIVO MAESTRO CLIENTES -- CUSTMSTR
A
R CSTMST
A CUST#          5S 0      TEXT('NUMERO DE CLIENTE')
A CSTNAM         20        TEXT('NOMBRE DE CLIENTE')
A CSTAD1         20        TEXT('CALLE')
A CSTAD2         20        TEXT('CALLE')
A CSTCTY         20        TEXT('CIUDAD')
A CSTSTE         2         TEXT('PROVINCIA')
A CSTZIP         5S 0      TEXT('DISTRITO POSTAL')
A
K CUST#
```

Figura 84. DDS para el programa de mantenimiento del archivo maestro

Las DDS para el archivo de base de datos utilizado por este programa describen un formato de registro: **CSTMST**. Se describe cada campo del formato de registro y el campo **CUST#** se identifica como el campo de clave para cada formato de registro.

Nota: Normalmente, los atributos de campo, tales como el número de posiciones decimales y el tipo de datos, están definidos en un archivo de referencia de campos en lugar de estar definidos en las DDS para el formato de registro. Los atributos se muestran en las DDS para que pueda observar cuáles son.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A* NOMB ARCHIV: CSTENT *
A* DESCRIPCION: ARCHIVO DE PANTALLA PARA CONSULTA MAESTRO CLIENTES *
A* LA OPCION DE CONSULTA AL MAESTRO DE CLIENTES *
A*****
AAN01N02N03T.Nombre++++RLon++TdpBLínPosFunciones+++++*****
A REF(CUSTMSTR)
A CHGINPDEF(CS)
A PRINT(QSYSPT)
A INDARA
A
R HDRSCN
A TEXT('SOLICITAR NUMERO DE
A CLIENTE')
A CA03(03 'FIN DE CONSULTA')
A CA05(05 'MODALIDAD DE ADICION')
A CA06(06 'MODALIDAD DE
A ACTUALIZACION')
A CA07(07 'MODALIDAD DE SUPRESION')
A CA07(07 'MODALIDAD DE SUPRESION')
A
A MODE          8A  O  1  4DSPATR(HI)
A                1  13'MODALIDAD'
A                DSPATR(HI)
A                2  4TIME
A                DSPATR(HI)
```

Guía del usuario RPG/400
Programa de ejemplo 3-Mantenimiento

```

A          2 28'MANTENIMIENTO ARCHIVO CLIENTES'
A          DSPATR(HI RI)
A          2 70DATE
A          EDTCDE(Y)
A          DSPATR(HI)
A          CUST#      R      Y  I 10 25DSPATR(CS)
A          CHECK(RZ)
A 51        ERRMSG('EL CLIENTE YA ESTA EN EL +
A          ARCHIVO' 51)
A 52        ERRMSG('EL CLIENTE NO ESTA EN EL +
A          ARCHIVO 52)

```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++*****
A          10 33'<--Entrar Número de Cliente'
A          DSPATR(HI)
A          23 4'F3 Fin de Trabajo'
A          23 21'F5 Añadir'
A          23 34'F6 Actualizar'
A          23 50'F7 Suprimir'
A          23 66'F8 Visualizar'
A          R CSTINQ      TEXT('VISUALIZAR INFO CLI')
A          CA12(12 'PANTALLA ANTERIOR')
A          MODE          8  O  1 4DSPATR(HI)
A          1 13'MODALIDAD'
A          DSPATR(HI)
A          2 4TIME
A          DSPATR(HI)
A          2 28'MANTENIMIENTO ARCHIVO CLIENTES'
A          DSPATR(HI RI)
A          2 70DATE
A          EDTCDE(Y)
A          DSPATR(HI)

```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++*****
A          4 14'Cliente:'
A          DSPATR(HI UL)
A          CUST#      R      O  4 25DSPATR(HI)
A          CSTNAM     R      B  6 25DSPATR(CS)
A 04          DSPATR(PR)
A          CSTAD1     R      B  7 25DSPATR(CS)
A 04          DSPATR(PR)
A          CSTAD2     R      B  8 25DSPATR(CS)
A 04          DSPATR(PR)
A          CSTCTY     R      B  9 25DSPATR(CS)
A 04          DSPATR(PR)
A          CSTSTE     R      B 10 25DSPATR(CS)
A 04          DSPATR(PR)
A          CSTZIP     R      B 10 40DSPATR(CS)
A          EDTCDE(Z)
A 04          DSPATR(PR)
A          23 2'F12 Cancelar'
A          MODE1      8  O 23 20
A          R CSTBLD      TEXT('AÑADIR REG CLI')
A          CA12(12 'PANTALLA ANTERIOR')
A          MODE          8  O  1 4DSPATR(HI)
A          1 13'MODALIDAD'      DSPATR(HI)
A          2 4TIME
A          DSPATR(HI)
A          2 28'MANTENIMIENTO ARCHIVO CLIENTES'
A          DSPATR(HI RI)
A          2 70DATE
A          EDTCDE(Y)
A          DSPATR(HI)
A          4 14'Cliente:' DSPATR(HI UL)

```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++*****
A          CUST#      R      O  4 25DSPATR(HI)
A          6 20'Nombre'      DSPATR(HI)
A          CSTNAM     R      I  6 25
A          7 17'Calle'      DSPATR(HI)
A          CSTAD1     R      I  7 25
A          8 17'Calle'      DSPATR(HI)
A          CSTAD2     R      I  8 25
A          9 20'Ciudad'     DSPATR(HI)
A          CSTCTY     R      I  9 25
A          10 19'Provin'    DSPATR(HI)
A          CSTSTE     R      I 10 25
A          10 36'Dist Post'DSPATR(HI)
A          CSTZIP     R      Y  I 10 40
A          23 2'F12 Cancelar Adición'

```

Figura 85. DDS del archivo del dispositivo de pantalla para la consulta al maestro de clientes

Las DDS del archivo del dispositivo de pantalla **CSTENT** contienen tres formatos de registro: **HDRSCN**, **CSTINQ** y **CSTBLD**. El registro **HDRSCN** solicita el número de cliente y la modalidad de proceso. El registro **CSTINQ** se utiliza para las modalidades de Actualización, Supresión y Visualización. Los campos se definen como entrada/salida (B en la posición 38). Los campos están protegidos cuando se selecciona la modalidad de Visualización o Supresión (**DSPATR(PR)**). El registro **CSTBLD** sólo proporciona campos de entrada para un nuevo registro (**I** en la posición 38).

El formato de registro **CUSHDG** contiene la constante **Consulta Maestro Clientes**; la palabra clave **ERRMSG** define los mensajes que se visualizarán si se produce un error. Las palabras clave CA definen las teclas de función que pueden utilizarse y las teclas de función asociadas con los indicadores de un programa en RPG.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*****
F*   ID PROGRAMA      - CUSTMNT                               *
F*   NOMB PROGRAMA   - MANTENIMIENTO MAESTRO CLIENTES       *
F*   ESTE PROGRAMA  AÑADE, ACTUALIZA, SUPRIME Y VISUALIZA  *
F*   DE CLIENTES EN EL ARCHIVO MAESTRO DE CLIENTES.        *
F*****
FNombarchIPEAF....LregLC1AIdPcLaEDispos+.....KSalidaEntradA....U1.*
FCUSTMSTRUF  E           K           DISK           A
FCSTENT  CF  E           WORKSTN

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C           CSTKEY      KLIST
C           KFLD         CUST#
C*****
C*   LINEA PRINCIPAL                                     *
C*****
C           MOVE 'VISUALIZAR'MODE
C           EXFMTHDRSCN
C*
C           *IN03      DOWEQ'0'
C           EXSR SETMOD
C*
C           CUST#      IFNE *ZERO
C           MODE       CASEQ'AÑADIR'  ADDSUB
C           MODE       CASEQ'ACTUALIZAR'UPDSUB
C           MODE       CASEQ'SUPRIMIR'DELSUB
C           MODE       CASEQ'VISUALIZAR'INQSUB
C           END
C           END
C*
C           EXFMTHDRSCN
C           END
C           MOVE '1'      *INLR

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C*   SUBROUTINA - ADDSUB                                     *
C*   PROPOSITO - AÑADIR NUEVO CLIENTE AL ARCHIVO          *
C*****
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C           ADDSUB     BEGSR
C           CSTKEY     CHAINCSTMST           50
C           *IN50      IFEQ '0'
C           MOVE '1'      *IN51
C           ELSE
C           MOVE '0'      *IN51
C           MOVE *BLANK  CSTNAM
C           MOVE *BLANK  CSTAD1
C           MOVE *BLANK  CSTAD2
C           MOVE *BLANK  CSTCTY
C           MOVE *BLANK  CSTSTE
C           Z-ADD*ZERO  CSTZIP
C           EXFMTCSTBLD
C           *IN12      IFEQ '0'
C           WRITECSTMST
C           END
C           END
C           ENDSR

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****

```

```

C*      SUBROUTINA - UPDSUB                                     *
C*      PROPOSITO - ACTUALIZACION REGISTRO MAESTRO CLIENTES   *
C*****
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C      UPDSUB      BEGSR
C      MOVE '0'          *IN04
C      CSTKEY      CHAINCSTMST          52
C      *IN52      IFEQ '0'
C      EXFMTCSTINQ
C      *IN12      IFEQ '0'
C      UPDATCSTMST
C      ELSE
C      EXCPTRLS
C      END
C      ENDSR
C*****
C*      SUBROUTINA - DELSUB                                     *
C*      PROPOSITO - SUPRESION REGISTRO MAESTRO CLIENTES       *
C*****
C      DELSUB      BEGSR
C      MOVE '1'          *IN04
C      CSTKEY      CHAINCSTMST          52
C      *IN52      IFEQ '0'
C      EXFMTCSTINQ
C      *IN12      IFEQ '0'
C      DELETCSTMST
C      ELSE
C      EXCPTRLS
C      END
C      ENDSR
C*****
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C*      SUBROUTINA - INQSUB                                     *
C*      PROPOSITO - VISUALIZACION REGISTRO MAESTRO CLIENTES   *
C*****
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C      INQSUB      BEGSR
C      MOVE '1'          *IN04
C      CSTKEY      CHAINCSTMST          52
C      *IN52      IFEQ '0'
C      EXFMTCSTINQ
C      EXCPTRLS
C      END
C      ENDSR
C*****
C*      SUBROUTINA - SETMOD                                     *
C*      PROPOSITO - DEFINICION DE MODALIDAD DE MANTENIMIENTO   *
C*****
C      SETMOD      BEGSR
C      *IN05      IFEQ '1'
C      MOVE 'AÑADIR 'MODE
C      MOVE MODE      MODE1
C      ELSE
C      *IN06      IFEQ '1'
C      MOVE 'ACTUALIZAR 'MODE
C      MOVE MODE      MODE1
C      ELSE
C      *IN07      IFEQ '1'
C      MOVE 'SUPRIMIR 'MODE
C      MOVE MODE      MODE1
C      ELSE
C*****
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C*
C      *IN08      IFEQ '1'
C      MOVE 'VISUALIZAR 'MODE
C      MOVE MODE      MODE1
C      ELSE
C      END
C      END
C      END
C      END
C      ENDSR
OCSTMST E          RLS

```

Figura 86. Especificaciones de descripción de archivo y de cálculo para el programa de mantenimiento

Este programa mantiene un archivo maestro de clientes para adiciones,

F12 Cancelar VISUALIZACION

Figura 87. Pantalla de modalidad de visualización para programa de mantenimiento

El usuario de la estación de trabajo responde a la solicitud de adición entrando un nuevo número de cliente como se muestra en la pantalla siguiente.

MODALIDAD DE ADICION
10:09:20 MANTENIMIENTO ARCHIVO CLIENTES 01/25/94

00009 <--Entre Número de Cliente

F3 Fin de Trabajo F5 Añadir F6 Actualizar F7 Suprimir F8 Visualizar

En la pantalla siguiente se añade un nuevo número de cliente al Archivo Maestro de Clientes.

MODALIDAD DE ADICION
10:09:36 MANTENIMIENTO ARCHIVO CLIENTES 01/25/94

Cliente: 00009

Nombre LANE, ROBERTO
Calle Bellavista
Calle Jaime II
Ciudad Inca
Provin P.Mallorca Dist Post 15679

F12 Cancelar Adición

Figura 88. Pantalla de modalidad de adición para programa de mantenimiento

El usuario de la estación de trabajo responde a la solicitud de actualización entrando un número de cliente, tal como se muestra en la pantalla siguiente.

```
-----+-----  
MODALIDAD DE ACTUALIZACION  
10:10:43                MANTENIMIENTO ARCHIVO CLIENTES                01/25/94  
  
00006  <--Entre Número de Cliente  
  
F3 Fin de Trabajo  F5 Añadir  F6 Actualizar  F7 Suprimir  F8 Visualizar  
-----+-----
```

Figura 89. Pantalla de modalidad de actualización para programa de mantenimiento

El usuario de la estación de trabajo responde a la solicitud de supresión entrando un nuevo número de cliente en la siguiente pantalla.

```
-----+-----  
MODALIDAD DE SUPRESION  
10:10:52                ARCHIVO MANTENIMIENTO CLIENTES                01/25/94  
  
00009  <--Entre Número de Cliente  
  
F3 Fin de Trabajo  F5 Añadir  F6 Actualizar  F7 Suprimir  F8 Visualizar  
-----+-----
```

Figura 90. Pantalla de modalidad de supresión para programa de mantenimiento

8.6.4 Programa de ejemplo 4-Proceso de subarchivo WORKSTN

Las figuras siguientes muestran un archivo **WORKSTN** :

```

+-----+
| Tabla 8. Lista de figuras para el proceso del subarchivo WORKSTN |
+-----+
| Figura | Contenido |
+-----+
| Figura 91 a continuación y Figura 92 | Especificaciones DDS para el archivo maestro y para el archivo del dispositivo de pantalla |
+-----+
| Figura 93 | Especificaciones de descripción de archivo y de cálculo |
+-----+
| Figura 94 | Pantalla de solicitud |
+-----+
| Figura 95 | Pantalla de visualización |
+-----+
    
```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A.....T.Nombre++++Lon++TdpB.....Funciones+++++++*
A* ARCHIVO MAESTRO CLIENTES -- CUSMSTP
A
  R CUSREC
A   CUST      5      TEXT('NUMERO CLIENTE')
A   NAME     20      TEXT('NOMBRE CLIENTE')
A   ADDR     20      TEXT('CALLE CLIENTE')
A   CITY     20      TEXT('CIUDAD')
A   STATE    2       TEXT('ABREVIATURA PROVINCIA')
A   ZIP      5  0    TEXT('CODIGO POSTAL')
A   SRHCOD   3       TEXT('CODIGO BUSQ. NOMBRE CLIENTE')
A   CUSTYP   1       TEXT('TIPO CLIENTE')
A   ARBAL    10  2   TEXT('SALDO C/D')
A*****
A* NOMB ARCHIV: CUSZIPL *
A* DESCRIPCION: VISTA LOGICA DEL ARCHIVO MAESTRO DE CLIENTES *
A*              (CUSMSTO) POR DISTRITO POSTAL (ZIP) *
A*****
A.....T.Nombre++++.Lon++TdpB.....Funciones+++++++*
A
  R CUSREC              PFILE (CUSMSTP)
A   ZIP                 R
A   NAME                 R
A   ARBAL                R
A
  K ZIP
    
```

Figura 91. DDS para el archivo maestro del programa de proceso del subarchivo WORKSTN

Las especificaciones de descripción de datos (DDS) para el archivo de base de datos utilizado por este programa describen un formato de registro: **CUSREC**. El archivo lógico **CUSZIPL** teclado por distrito postal se basa en el archivo físico **CUSMSTP** tal como indica la palabra clave **PFILE**. El formato de registro generado por el archivo lógico incluirá solamente los campos especificados en el archivo lógico DDS. Se excluirán todos los demás campos.

Nota: Normalmente, los atributos de campo, tales como el número de posiciones decimales y el tipo de datos, están definidos en un archivo de referencia de campos en lugar de estar definidos en las DDS para el formato de registro. Los atributos se muestran en las DDS para que pueda observar cuáles son.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A* NOMB ARCHIV: CUSSRC *
A* DESCRIPCION: VISUALIZACION MAESTRO CLIENTES POR DISTRITO POSTAL *
A*****
AAN01N02N03T.Nombre++++RLon++TdpBLínPosFunciones+++++++*
A
  REF (CUSMSTP)
A   CHGINPDFT (CS)
A   PRINT (QSYSVRT)
A   INDARA
A   CA03(03 'FIN DEL TRABAJO')
A
  R HEAD
A
  OVERLAY
A   2 4TIME
A   DSPATR (HI)
A   2 28'BUSQUEDA ARCHIVO CLIENTES'
A   DSPATR (HI RI)
A   2 70DATE
    
```

```

A          EDTCDE (Y)
A          DSPATR (HI)
A          R FOOT1
A          23 6'INTRO - Continuar'
A          DSPATR (HI)
A          23 29'F3 - Fin Trabajo'
A          DSPATR (HI)
A          R FOOT2
A          23 6'INTRO - Continuar'
A          DSPATR (HI)
A          23 29'F3 - Fin Trabajo'
A          DSPATR (HI)
A          23 47'F4 - REARRANCAR DISTRITO
A          POSTAL'
A          DSPATR (HI)

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++*****
A          R PROMPT
A          OVERLAY
A          4 4'Entrar Distrito Postal'
A          DSPATR (HI)
A          ZIP          R          Y I 4 19DSPATR (CS)
A          CHECK (RZ)
A 61          ERRMSG ('NO ENCONTRADO DISTRITO
A          POSTAL + 61)
A          R SUBFILE
A          NAME          R          9 4
A          ARBAL          R          9 27EDTCDE (J)
A          R SUBCTL
A          SFLCTL (SUBFILE)
A 55          SFLCLR
A N55          SFLDSPCTL
A N55          SFLDSP
A          SFLSIZ (13)
A          SFLPAG (13)
A          ROLLUP (95 'GIRO ARRIBA')
A          OVERLAY
A          CA04 (04 'REARRANCAR DISTRITO
A          POSTAL')
A          4 4'Distrito Postal'
A          ZIP          R          O 4 14DSPATR (HI)
A          7 4'Nombre Cliente'
A          DSPATR (HI UL)
A          7 27'Saldo C/D'
A          DSPATR (HI UL)

```

Figura 92. DDS para el archivo del dispositivo de pantalla del programa de proceso del subarchivo WORKSTN

Las DDS para el archivo del dispositivo de pantalla **CUSSRC** contienen seis formatos de registro: **HEAD**, **FOOT1**, **FOOT2**, **PROMPT**, **SUBFILE** y **SUBCTL**.

El formato de registro **PROMPT** solicita al usuario que entre un distrito postal. Si el distrito postal no se encuentra en el archivo, se visualiza un mensaje de error. El usuario puede pulsar la tecla F3, activando el indicador 03, para finalizar el programa.

El formato del registro **SUBFILE** debe estar definido inmediatamente antes del formato de registro de subarchivo **SUBCTL**. El formato de registro de subarchivo, que está definido con la palabra clave **SFL**, describe cada campo del registro y especifica la ubicación donde va a aparecer el primer registro en la pantalla (aquí, en la línea 9).

El formato del registro de control del subarchivo contiene las siguientes palabras clave únicas:

- SFLCTL** identifica este formato como formato de registro de control y nombra el formato de registro de subarchivo asociado.
- SFLCLR** describe cuando van a borrarse los registros existentes en el subarchivo (cuando el indicador 55 esté activado). Se necesita esta palabra clave para pantallas adicionales.
- SFLDSPCTL** indica cuándo tiene que visualizarse el formato de registro de control del subarchivo (cuando esté desactivado el indicador 55).
- SFLDSP** indica cuándo tiene que visualizarse el subarchivo (cuando el indicador 71 esté activado).
- SFLSIZ** especifica el tamaño total del subarchivo. En este ejemplo el tamaño del subarchivo es de 13 registros que se visualizan en las líneas 9 a 21.

- **SFLPAG** define el número de registros de una página. En este ejemplo el tamaño de la página es el mismo que el tamaño del subarchivo.
- **ROLLUP** indica que está activado en el programa el indicador 95 cuando se utiliza la función de giro hacia arriba.

La palabra clave **OVERLAY** define este formato de registro de control de subarchivo como un formato de recubrimiento. El formato de registro puede grabarse sin que el sistema OS/400 borre primero la pantalla. La tecla F4 es válida para repetir la búsqueda con el mismo distrito postal. (Esta utilización de la tecla F4 permite una forma de giro hacia abajo).

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*****
F*   ID PROGRAMA   - CUSTSFL                               *
F*   NOMB PROGRAMA - BUSQUEDA EN MAESTRO DE CLIENTES      *
F*   ESTE PROGRAMA VISUALIZA EL ARCHIVO MAESTRO DE CLIENTES POR *
F*   DISTRITO POSTAL                                     *
F*****
FNombarchIPEAF...LregLCIAIDePcLaEDispos+.....KSalidaEntradA...U1.*
FCUSZIPL IF E           K           DISK
FCUSSRC CF E           WORKSTN
F
                                RECNUMKSFIL  SUBFILE

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C
      CSTKEY           KLIST
C
      KFLD             ZIP
C*****
C*   LINEA PRINCIPAL                                     *
C*****
C
      WRITEFOOT1
C
      WRITEHEAD
C
      EXFMTPROMPT
C*
C
      *IN03           DOWEQ '0'
C
      CSTKEY         SETLLCUSREC           20
C
      *IN20           IFEQ *ZERO
C
      MOVE '1'       *IN61
C
      ELSE
C
      EXSR SFLPRC
C
      END
C
      *IN03           IFEQ '0'
C
      *IN04           IFEQ '0'
C
      *IN61           IFEQ '0'
C
      WRITEFOOT1
C
      WRITEHEAD
C
      END

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C
      EXFMTPROMPT
C
      END
C
      END
C
      END
C*
C
      SETON           LR
C*****
C*   SUBROUTINA   - SFLPRC                               *
C*   PROPOSITO   - PROCESAR SUBARCHIVO Y VISUALIZAR      *
C*****
C
      SFLPRC         BEGSR
C
      NXTPAG         TAG
C
      EXSR SFLCLR
C
      EXSR SFLFIL
C
      SAMPAG         TAG
C
      WRITEFOOT2
C
      WRITEHEAD
C
      EXFMTSUBCTL
C
      *IN95         IFEQ '1'
C
      *IN71         IFEQ '0'
C
      GOTO NXTPAG
C
      ELSE
C
      GOTO SAMPAG
C
      END
C
      END
C
      ENDSR

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C*   SUBROUTINA   - SFLFIL                               *
C*   PROPOSITO   - RELLENAR SUBARCHIVO                   *

```

```

C*****
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C      SFLFIL      BEGSR
C      *IN21      DOWEQ'0'
C      ZIP        READECUSREC          71
C      *IN71      IFEQ '1'
C              MOVE '1'          *IN21
C              ELSE
C              ADD 1          RECNUM
C              WRITESUBFILE          21
C              END
C              END
C              ENDSR
C*****
C*      SUBROUTINA - SFLCLR          *
C*      PROPOSITO - BORRAR REGISTROS SUBARCHIVO          *
C*****
C      SFLCLR      BEGSR
C              MOVE '1'          *IN55
C              WRITESUBCTL
C              MOVE '0'          *IN55
C              MOVE '0'          *IN21
C              Z-ADD*ZERO          RECNUM 50
C              ENDSR

```

Figura 93. Especificación de descripción de archivo y especificación de cálculo para el programa de proceso del subarchivo WORKSTN

Las especificaciones de descripción de archivo identifican el archivo de disco que va buscarse y el archivo del dispositivo de pantalla que va a utilizarse (**CUSSRC**). La línea de continuación para el archivo **WORKSTN** identifica el formato de registro (**SUBFILE**) que se utilizará como subarchivo. El campo de número relativo de registro (**RECNUM**) especificado en las posiciones 47 a 52 de la línea de continuación controla el registro al que se accede dentro del subarchivo.

El programa visualiza el formato de registro **PROMPT** y espera la respuesta del usuario de la estación de trabajo. La tecla F3 activa el indicador 03 el cual controla el final del programa. El distrito postal (**ZIP**) se utiliza para situar el archivo **CUSZIPL** mediante la operación **SETLL**. Tenga en cuenta que el nombre del formato de registro **CUSREC** se utiliza en la operación **SETLL** en lugar del nombre del archivo **CUSZIPL**. Si no se encuentra ningún registro, se visualiza un mensaje de error.

La subrutina **SFLPRC** maneja el proceso para el subarchivo: lo borra, rellena y visualiza. El subarchivo está preparado para solicitudes adicionales en la subrutina **SFCLR**. Si está activado el indicador 55, no se produce ninguna acción en la pantalla, pero se borra el área del almacenamiento principal para los registros del subarchivo. La rutina **SFLFIL** rellena el subarchivo con registros. Se lee un registro del archivo **CUSZIPL**. Si el distrito postal es el mismo, se incrementa la cuenta de registros (**RECNUM**) y se graba el registro en el subarchivo. Esta subrutina se repite hasta que esté lleno el subarchivo (indicador 21 en la operación **WRITE**) o se produzca un final de archivo en el archivo **CUSZIPL** (indicador 71 en la operación **READ**). Cuando el subarchivo se llena o se produce un fin de archivo, se graba el subarchivo en la pantalla mediante la operación **EXFMT** con el formato de control del subarchivo. El usuario revisa la pantalla y decide:

- Finalizar el programa pulsando la tecla F3.
- Volver a arrancar el distrito postal pulsando la tecla F4. No se visualiza el formato de registro **PROMPT** y se visualiza el subarchivo comenzando por el mismo distrito postal.
- Rellenar otra página pulsando las teclas de **GIRO ARRIBA**. Si se produce un final de archivo en el archivo **CUSZIPL**, vuelve a visualizarse la página actual; de lo contrario, se borra el subarchivo y se visualiza la página siguiente.
- Continuar con otro distrito postal, pulsando la tecla **INTRO**. Se visualiza el formato de registro **PROMPT**. El usuario puede entrar un distrito postal o finalizar el programa.

En la siguiente pantalla, el usuario ha de entrar un distrito postal en respuesta a la solicitud.

```

-----
| 10:11:08          BUSQUEDA ARCHIVO CLIENTES          01/25/94          |
|                                                         |
| Entre distrito postal 72901                             |
|                                                         |
-----

```

INTRO - Continuar F3 - Fin de Trabajo

Figura 94. Pantalla de Solicitud para el Programa de Proceso del Subarchivo WORKSTN

El subarchivo se graba en la pantalla tal como se muestra:

```
10:11:23                    BUSQUEDA ARCHIVO CLIENTES                    01/25/94
Distrito Postal 72901

Nombre Cliente            Saldo C/D
FLORES                    11.111.111,00
LANG                      22.222.222,00
ARGEMI                    33.333.333,00
BELL                      44.444.444,00
ROMAN                     55.555.555,00
FRASER                    66.666.666,00
DOMINGO                   77.777.777,00
HEREDIA                   88.888.888,00
CASANOVAS                11.111.111,00
RICO                      11.111.222,00
MORALES                   33.333.333,00
SUAREZ                    44.444.444,00
KENT                      55.555.555,00

INTRO - Continuar    F3 - Fin Trabajo    F4 - REARRANCAR DISTRITO POSTAL
```

Figura 95. Pantalla de visualización para el programa de proceso del subarchivo WORKSTN

8.6.5 Programa de ejemplo 5-Consulta por código postal y búsqueda por nombre

Las figuras siguientes ilustran un programa de consulta sencillo utilizando el archivo **WORKSTN**:

Figura	Contenido
Figura 96 a continuación y Figura 97	Especificaciones DDS para el archivo maestro y para el archivo del dispositivo de pantalla
Figura 98	Especificaciones de descripción de archivo y de cálculo
Figura 99	Pantalla de solicitud
Figura 100	Pantalla de información
Figura 101	Pantalla de información detallada

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A.....T.Nombre++++Lon++TDpB.....Funciones+++++*****
A* ARCHIVO MAESTRO CLIENTES -- CUSMSTP
A      R CUSREC
A      CUST          5          TEXT('NUMERO CLIENTE')
A      NAME          20          TEXT('NOMBRE CLIENTE')
A      ADDR          20          TEXT('CALLE CLIENTE')
A      CITY          20          TEXT('CIUDAD')
A      STATE         2          TEXT('ABREVIATURA PROVINCIA')
A      ZIP            5  0       TEXT('CODIGO POSTAL')
A      SRHCOD        3          TEXT('CODIGO BUSQ. NOMBRE CLIENTE')
A      CUSTYP         1          TEXT('TIPO CLIENTE')
A      ARBAL         10  2       TEXT('SALDO C/D')
A*****
A* NOMB ARCH      : MLGMSTL1 *
A* DESCRIPCION: VISTA LOGICA DEL ARCHIVO MAESTRO DE CLIENTES *
A*      (CUSMSTP) POR DISTRITO POSTAL (ZIP) Y NOMBRE (NAME) *
A*****
A.....T.Nombre++++Lon++TDpB.....Funciones+++++*****
A      R CUSREC          PFILE(CUSMSTP)
A      K ZIP
A      K NAME
    
```

Figura 96. Especificaciones DDS para archivo maestro para consulta por distrito postal

Las DDS para el archivo de base de datos utilizado en este programa definen un formato de registro de nombre **CUSREC** e identifican los campos **ZIP** y **NAME** como campos de clave.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A* NOMB ARCH      : MLG265D *
A* DESCRIPTION: VISUALIZAR MAESTRO CLIENTES POR DISTRITO POSTAL Y *
A*      NOMBRE *
A*****
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++*****
A      DSPSIZ(24 80 *DS3)
A      REF(CUSMSTP)
A      CHGINPDFT(CS)
A      PRINT(QSYSVRT)
A      INDARA
A      CA03(03 'FIN DEL TRABAJO')
A      R HEAD
A      OVERLAY
A      2 4TIME
A      DSPATR(HI)
A      2 29'Consulta Maestro Clientes'
A      DSPATR(HI UL)
A      2 70DATE
A      EDTCDE(Y)
A      DSPATR(HI)
A      R FOOT1
A      23 6'INTRO - Continuar'
A      DSPATR(HI)
A      23 29'F3 - Fin del Trabajo'
A      DSPATR(HI)
    
```

```

A      R FOOT2
A      23 6'INTRO - Continuar'
A      DSPATR(HI)
A      23 29'F3 - Fin del Trabajo'
A      DSPATR(HI)
A      23 47'F4 - Rearrancar Distrito Postal'
A      DSPATR(HI)
A      R PROMPT
A      OVERLAY
A      4 4'Entrar Distrito Postal'
A      DSPATR(HI)
A      ZIPCD      R      Y  I 4 19REFFLD (ZIP CUSMSTP)
A      CHECK(RZ) DSPATR(CS)
A      5 7'Nombre de Búsqueda'
A      DSPATR(HI)
A      SRCNAM     R      I 5 19REFFLD (NAME CUSMSTP)
A      DSPATR(CS)
A      *.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A      AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++*****
A      R SUBFILE
A      CHANGE(99 'CAMPO CAMBIADO')
A      SFL
A      SEL      1  B  9 8DSPATR(CS)
A      VALUES(' ' 'X')
A      ZIP      R      O  9 17
A      CUST     R      O  9 30
A      NAME     R      O  9 43
A      R SUBCTL
A      SFLCTL(SUBFILE)
A      SFLSIZ(0013)
A      SFLPAG(0013)
A  55
A N55
A N55
A      SFLCLR
A      SFLDSPCTL
A      SFLDSP
A      ROLLUP(95 'GIRO ARRIBA')
A      OVERLAY
A      CA04(04 'REARRANCAR DISTRITO
A      POSTAL')
A      4 4'Distrito Postal'
A      ZIPCD     R      O  4 17REFFLD (ZIP CUSMSTP)
A      DSPATR(HI)
A      5 4'Nombre de Búsqueda'
A      SRCNAM     R      O  5 17REFFLD (NAME CUSMSTP)
A      DSPATR(HI)
A      7 6'Seleccionar'
A      DSPATR(HI)
A      8 6' "X" Distrito Postal Numero -
A      Nombre cliente '
A      DSPATR(HI)
A      DSPATR(UL)
A      *.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A      AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++*****
A      R CUSDSP
A      OVERLAY
A      CA04(04 'REARRANCAR DISTRITO
A      POSTAL')
A      6 25'Cliente'
A      CUST      5A  O  6 35DSPATR(HI)
A      8 25'Nombre'
A      NAME      20A O  8 35DSPATR(HI)
A      10 25'Calle'
A      ADDR      20A O 10 35DSPATR(HI)
A      12 25'Ciudad'
A      CITY      20A O 12 35DSPATR(HI)
A      14 25'Prov.'
A      STATE     2A  O 14 35DSPATR(HI)
A      14 41'Distrito postal'
A      ZIP      5S 00 14 50DSPATR(HI)
A      16 25'Saldo C/D'
A      ARBAL     10Y 20 16 42DSPATR(HI)
A      EDTCDE(J)

```

Figura 97. Especificaciones DDS para archivos de dispositivo de pantalla para consulta por distrito postal

Las DDS para el archivo del dispositivo de pantalla **CUSSRC** contienen siete formatos de registro: **HEAD**, **FOOT1**, **FOOT2**, **PROMPT**, **SUBFILE**, **SUBCTL** y **CUSDSP**.

El formato de registro **PROMPT** solicita al usuario que entre un distrito postal y un nombre de búsqueda. Si no se realiza ninguna entrada, la visualización comienza al principio del archivo. El usuario puede pulsar la tecla F3, activando el indicador 03, para finalizar el programa.

El formato del registro **SUBFILE** debe estar definido inmediatamente antes del formato de registro de subarchivo **SUBCTL**. El formato de registro de subarchivo definido con la palabra clave **SFL** describe cada campo del registro y especifica la ubicación en donde aparecerá en la pantalla el primer registro (aquí en la línea 9).

El formato del registro de control del subarchivo **SUBCTL** contiene las siguientes palabras clave únicas:

- **SFLCTL** identifica este formato como formato de registro de control y denomina el formato de registro de subarchivo asociado.
- **SFLCLR** describe cuándo van a borrarse los registros existentes del subarchivo (cuando está activado el indicador 55). Se necesita esta palabra clave para pantallas adicionales.
- **SFLDSPCTL** indica cuándo va a visualizarse el formato del registro de control del subarchivo (cuando está desactivado el indicador 55).
- **SFLDSP** indica cuándo va a visualizarse el subarchivo (cuando está desactivado el indicador 55).
- **SFLSIZ** especifica el tamaño total del subarchivo. En este ejemplo, el tamaño del subarchivo es de 15 registros que se visualizan en las líneas 9 a 23.
- **SFLPAG** define el número de registros de una página. En este ejemplo el tamaño de la página es el mismo que el tamaño del subarchivo.
- **ROLLUP** indica que el indicador 95 se activa en el programa cuando se utiliza la función de giro hacia arriba.

La palabra clave **OVERLAY** define este formato de registro de control de subarchivo como un formato de recubrimiento. El formato de registro puede grabarse sin que el sistema OS/400 borre antes la pantalla. La tecla F3 es válida para repetir la búsqueda con el mismo distrito postal. (Esta utilización de la tecla F4 permite una forma de giro hacia abajo).

El formato de registro **CUSDSP** visualiza la información de los clientes seleccionados.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*****
F*   ID PROGRAMA   - MLG265                               *
F*   NOMBR Progr  - BUSQUEDA EN LISTA DISTRIBUCION POR DISTRITO *
F*                   POSTAL/NOMBRE
F*****
FNombarchIPEAF....LregLCIAIDePcLaEDispos+.....KSalidaEntrada....U1.*
FMLGMSTL1IF E           K           DISK
FMLG265D CF E           WORKSTN
F                   RECNUMKSFIL SUBFILE
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C           CSTKEY      KLIST
C           KFLD        ZIPCD
C           KFLD        SRCNAM
C           ZIPKEY      KLIST
C           KFLD        ZIP
C           KFLD        NAME
C*****
C*   LINEA PRINCIPAL                                     *
C*****
C           WRITEFOOT1
C           WRITEHEAD
C           EXFMTPROMPT
C           *IN03      DOWEQ'0'
C           CSTKEY    SETLLCUSREC
C           EXSR SFLPRC
C           EXSR SFLCHG
C           *IN03      IFEQ '0'
C           *IN04      ANDEQ'0'
C           WRITEFOOT1
C           WRITEHEAD
C           EXFMTPROMPT
C           END
C           END
C*
C           SETON           LR

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C*   SUBROUTINA   - SFLPRC                               *
C*   PROPOSITO    - PROCESAR SUBARCHIVO Y VISUALIZAR     *

```

```

C*****
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          SFLPRC      BEGSR
C          NXTPAG      TAG
C                      EXSR SFLCLR
C                      EXSR SFLFIL
C          SAMPAG      TAG
C                      WRITEFOOT2
C                      WRITEHEAD
C                      EXFMTSUBCTL
C          *IN95       IFEQ '1'
C          *IN71       IFEQ '0'
C                      GOTO NXTPAG
C                      ELSE
C                      GOTO SAMPAG
C                      END
C                      END
C                      ENDSR

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C*      SUBROUTINA - SFLFIL *
C*      PROPOSITO - RELLENAR SUBARCHIVO *
C*****
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          SFLFIL      BEGSR
C          *IN21       DOWEQ'0'
C                      READ CUSREC          71
C          *IN71       IFEQ '1'
C                      MOVE '1'           *IN21
C                      ELSE
C                      ADD 1              RECNUM
C                      MOVE *BLANK       SEL
C                      WRITESUBFILE      21
C                      END
C                      END
C                      ENDSR
C*****
C*      SUBROUTINA - SFLCLR *
C*      PROPOSITO - BORRAR REGISTROS SUBARCHIVO *
C*****
C          SFLCLR      BEGSR
C                      MOVE '1'           *IN55
C                      WRITESUBCTL
C                      MOVE '0'           *IN55
C                      MOVE '0'           *IN21
C                      Z-ADD*ZERO       RECNUM 50
C                      ENDSR

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C*      SUBROUTINA - SFLCHG *
C*      PROPOSITO - REGISTRO CLIENTE SELECCIONADO *
C*****
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          SFLCHG      BEGSR
C                      READCSUBFILE      98
C          *IN98       IFEQ '0'
C          ZIPKEY      CHAINCUSREC        71
C                      EXFMTCUSDSP
C                      END
C                      ENDSR

```

Figura 98. Especificación de descripción de archivo y especificación de cálculo para el programa de consulta por distrito postal y búsqueda por nombre

Las especificaciones de descripción de archivo identifican el archivo en disco que va a buscarse y el archivo del dispositivo de pantalla que va a utilizarse (MLG265D). La línea de continuación para el archivo WORKSTN identifica el formato de registro que se va a utilizar como subarchivo (SUBFILE). El campo del número relativo de registro (RECNUM) especificado en las posiciones 47 a 52 de la línea de continuación controla el registro al que se accede dentro del subarchivo.

El programa visualiza el formato de registro PROMPT y espera la respuesta del usuario de la estación de trabajo. La tecla F3 activa el indicador 03, el cual controla el final del programa. El distrito postal (ZIP) y el nombre (NAME) se utilizan como clave para situar el archivo MLGMSTL1 mediante la operación SETLL. Observe que se utiliza el nombre del formato de registro CUSREC en la operación SETLL en lugar del nombre del archivo MLGMSTL1.

Programa de ejemplo 5-Consulta por código postal y búsqueda por nombre

La subrutina **SFLPRC** maneja el proceso para el subarchivo: lo borra, rellena y visualiza. El subarchivo está preparado para solicitudes adicionales en la subrutina **SFCLR**. Si está activado el indicador 55, no se produce ninguna acción en la pantalla, pero se borra el área del almacenamiento principal para los registros del subarchivo. La rutina **SFLFIL** rellena el subarchivo con registros. Se lee un registro del archivo **MLGMSLT1**, se incrementa la cuenta de registros (**RECNUM**) y se graba el registro en el subarchivo. Esta subrutina se repite hasta que está lleno el subarchivo (indicador 21 en la operación **WRITE**) o se produce un final de archivo en el archivo **MLGMSTL1** (indicador 71 en la operación **READ**). Cuando el subarchivo se llena o se produce un fin de archivo, se graba el subarchivo en la pantalla mediante la operación **EXFMT** con el formato de control del subarchivo. El usuario revisa la pantalla y decide:

- Finalizar el programa pulsando la tecla F3.
- Rearrancar el subarchivo pulsando la tecla F4. No se visualiza el formato de registro **PROMPT** y se visualiza el subarchivo comenzando por el mismo distrito postal.
- Rellenar otra página pulsando las teclas de **GIRO HACIA ARRIBA**. Si se ha producido un final de archivo en el archivo **MLGMST1**, vuelve a visualizarse la página actual; de lo contrario, se borra el subarchivo y se visualiza la página siguiente.
- Visualizar los detalles del cliente entrando **X** y pulsando **INTRO**. El usuario puede volver a continuación a la pantalla **PROMPT** pulsando **INTRO**, volver a visualizar el subarchivo pulsando la tecla F4 o finalizar el programa pulsando la tecla F3.

En la pantalla siguiente, el usuario responde a la solicitud inicial entrando un distrito postal y un nombre.

```

-----
11:07:56                Consulta Maestro Clientes                01/25/94

Entre Distrito Postal 26903
Nombre de Búsqueda ALVAREZ

INTRO - Continuar       F3 - Fin de Trabajo
-----

```

Figura 99. Pantalla de solicitud para búsqueda por distrito postal

El usuario solicita más información entrando una X en la pantalla siguiente.

```

-----
11:09:20                Consulta Maestro Clientes                01/25/94

Distrito Postal 26903
Nombre de Búsqueda ALVAREZ

Seleccione
"X"      Dist Post  Número  Nombre Cliente
        26903     00011  ALVAREZ
        26903     00012  HEREDIA
        26903     00013  ROMERO
-----

```

Programa de ejemplo 5-Consulta por código postal y búsqueda por nombre

	26903	00014	MORALES
	26903	00015	CARRERAS
X	26903	00016	DELGADO
	26903	00017	ALARCON
	26903	00018	DOMINGO
	26903	00019	TORRENCE
	26903	00019	ARAGON
	26903	00020	VILLALBA
	27810	00021	VALCARCEL
	27810	00022	VENTURA
	27810	00023	BOSCH

INTRO - Continuar F3 - Fin del Trabajo F4 - Rearrancar Distrito Postal

Figura 100. Pantalla de información para búsqueda por distrito postal

En la siguiente pantalla, el usuario selecciona la tecla de función adecuada para continuar o finalizar la consulta.

11:09:20	Consulta Maestro Clientes	01/25/94
Cliente	00016	
Nombre	DELGADO	
Calle	Europa, 150	
Ciudad	Tarrasa	
Prov	Barcelona	Distrito Postal 26903
Saldo A/D	100,00	

INTRO - Continuar F3 - Fin del Trabajo F4 - Rearrancar Distrito Postal

Figura 101. Pantalla de información detallada para búsqueda por distrito postal

Las figuras siguientes ilustran la utilización de un **WORKSTN** con nombre **FORMAT** en las especificaciones de salida.

Figura	Contenido
Figura 102 a continuación	Especificaciones DDS para el archivo de dispositivo de pantalla
Figura 103	Especificaciones de descripción de archivo, entrada, cálculo y salida

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A* NOMB ARCHIV: CUSINQ *
A* DESCRIPCION: ARCHIVO DE PANTALLA PARA NOMBRE FORMATO EN SALIDA *
A*****
A.....T.Nombre++++.Lon++TDpB.....Funciones+++++
A
A      R ITMPMT
A
A      TEXT('SOLICITUD CONSULTA
A      INVENTARIO')
A      CF01(15 'FIN DEL PROGRAMA')
A      1 2'Solicitud Consulta Inventario'
A      2 2'Entrar Número Artículo'
A      RECID      1  I 2 23DFT('A') DSPATR(ND PR)
A      ITEM      5  I 2 25
A 99
A      R ITMDTL
A      TEXT('DETALLE INVENTARIO')
A      OVERLAY
A      5 2'Núm. Artículo'
A      5 14'Descripción'
A      5 41'Precio'
A      5 53'Vendido'
A      5 62'Disponible'
A      ITEM      5      7 2
A      DESCRP   20      7 14
A      PRICE     8      7 41
A      PENDNG    5      7 53
A      ONHAND    5      7 62
    
```

Figura 102. Especificaciones DDS para el archivo WORKSTN descrito por programa con nombre FORMAT en las especificaciones de salida

Las especificaciones de descripción de datos para el archivo del dispositivo de pantalla **CUSINQ** describen cómo se visualizan los datos enviados desde el programa en RPG/400.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF....LregLC1AIDePcLaEDispos+.....KSalidaEntradA....U1.*
FCUSINQ CP F      50      WORKSTN      KPASS *NOIND
FINVMSTL IF E      K      DISK

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
INomreg++....In.....*
ICUSINQ NS 03 2 CA
I.....Campoexter.....Campo+L1C1..PoNeCe...*
I      1 1 *IN15
I      2 2 RECID
I      3 7 ITEM

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgcComentarios+++
C      *IN03      IFEQ '1'
C      ITEM      CHAININVDTL      99
C      END
C      *IN15      IFEQ '1'
C      MOVE '1'      *INLR
C      RETRN
C      END

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OCUSINQ D      1P
O      OR      03
O      OR      99
    
```

Programa de ejemplo 6-Archivo WORKSTN descrito por programa con nombre de formato FORMAT en las especificaciones de salida

```

O.....N01N02N03Campo+YBUposPConstante/palabraedición+...*
O                                     K6 'ITMPMT'
O                                     1
O           D           03N99
O                                     K6 'ITMDTL'
O                                     5
O                                     ITEM
O                                     25
O                                     DESCRP
O                                     33
O                                     PRICE
O                                     38
O                                     PENDNG
O                                     43
O                                     ONHAND

```

Figura 103. Especificaciones de descripción de archivo, entrada, cálculo y salida para el archivo WORKSTN descrito por programa con nombre FORMAT en las especificaciones de salida

Dado que el nombre de formato **ITMPMT** está condicionado por **1P** en las especificaciones de salida, el archivo se graba antes de que se produzca ninguna operación de entrada. Este formato también se graba al archivo cuando está activado el indicador 03 o el indicador 99. Si está activado el indicador 99, se visualiza el mensaje de error que está definido en las DDS. Para transferir el indicador 99 a la salida, defina el campo ***IN99** en el registro de salida. Se graba el formato **ITMDTL** en el archivo cuando está activado el indicador 03 y no lo está el indicador 99. Las posiciones finales para los campos deben ser las mismas que las posiciones finales definidas en el listado de las DDS.

8.6.7 Programa de ejemplo 7-Línea de comienzo variable

Las figuras siguientes muestran ejemplos de programa para una línea de comienzo variable

```

+-----+
| Tabla 11. Lista de figuras para una línea de comienzo variable |
+-----+
| Figura | Contenido |
+-----+
| Figura 104 a | Especificaciones DDS para el archivo de |
| continuación | dispositivo de pantalla |
+-----+
| Figura 105 | Especificaciones de descripción de archivo, |
| | adicionales y de cálculo |
+-----+

```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A*  NOMBR ARCH : INQUIRY *
A*  DESCRIPCION: ARCHIVO DE PANTALLA PARA LINEA DE COMIENZO *
A*****
AAN01N02N03T.Nombre++++RLon++TdPBLínPosFunciones+++++*****
A
A          R PROMPT          PRINT
A          MONTH          9A O 6 15DSPATR (HI)
A          DAY            2 O 6 26DSPATR (HI)
A          YR             2 O 6 30DSPATR (HI)
A          6 45TIME      DSPATR (HI)

```

Figura 104. DDS para línea de comienzo variable

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*****
F*  ID PROGRAMA - VARLINE *
F*  NOMBRE PROGRAMA - PANTALLA DE LINEA DE COMIENZO VARIABLE *
F*****
FNombarchIPEAF....LregLCIAIDePcLaEDispos+.....KSalidaEntrada....U1.*
FINQUIRY CF E          WORKSTN
F          KSLN      SLNFLD
E....Arch (de) Arch (a) +NombreN/rN/tbLonPDSNommatLonPDSComentarios+++++
E          TABM      1 12 2 0 TABD      9      TABLA DE MESES
CL0N01N02N03Factor1+++OpcdeFactor2+++ResultLenDHHiLoEqComments+++++
C          MOVE UDAY      DAY
C          MOVE UYEAR      YR
C          UMONTH      LOKUPTABM      TABD      66
C          *IN66      IFEQ '1'
C          MOVE TABD      MONTH
C          END
C          Z-ADD6      SLNFLD      20
C*
C          EXFMT PROMPT
C*
C          MOVE '1'      *INLR
**
01ENERO
02FEBRERO
03MARZO
04ABRIL
05MAYO
06JUNIO
07JULIO
08AGOSTO
09SEPIEMBRE
10OCTUBRE
11NOVIEMBRE
12DICIEMBRE

```

Figura 105. Especificaciones de descripción de archivo, adicionales y de cálculo para línea de comienzo variable

Un campo de número de línea de comienzo (**SLN**) determina el número de la línea en la que se graba un formato de registro en un archivo de pantalla. Puede especificarse **SLN** tanto para los archivos descritos por programa como para los archivos descritos externamente. Para utilizar una línea de comienzo variable para un formato de registro de archivo de pantalla, especifique la opción **SLN** en las especificaciones de continuación de archivo. Las DDS para el archivo deben especificar **SLNO (*VAR)** para uno o más formatos de registro. Sólo estos formatos de registro están afectados

por el valor del campo **SLN**.

En las operaciones de salida para el archivo, el valor del campo **SLN** determina el número de línea en la que se graban realmente los formatos de registro. Si el campo **SLN** tiene un valor de 1 a 24, se le resta 1 al valor y el resultado se suma a los números de líneas especificados en las DDS. Los valores resultantes se emplean como números reales de línea para grabar los campos y las constantes especificadas en las DDS. Sin embargo, la línea de comienzo para el formato de registro es el valor del campo **SLN**. Esto significa que el formato de registro grabado ocupa todas las líneas entre el comienzo del formato y el número real más alto de línea grabado en la pantalla. Si el campo **SLN** tiene un valor de 0, aparece un formato en la pantalla como si se hubiese especificado un valor en el campo **SLN** de 1. Si el valor del campo **SLN** es negativo o mayor que 24, se emite un mensaje de error 1299 de RPG/400. Para obtener más información, consulte las publicaciones *Guía para la Gestión de Datos* y *DDS Reference*.

En este ejemplo, la operación **EXFMT** utiliza un campo de número de línea de comienzo (**SLNFLD**) con un valor de 6. Esto hace que se visualice el formato de registro comenzando por la línea 06, los campos de salida se muestran en la línea 11:

$(6(\text{SLNFLD}) - 1 + 6(\text{número línea comienzo DDS}))$.

La Figura 106 muestra un formato de pantalla especificado con una línea de comienzo variable.



Figura 106. Pantalla de Solicitud para Línea de Comienzo Variable

8.6.8 Programa de ejemplo 8-Operación de lectura con límite de tiempo de espera excedido

Las figuras siguientes muestran ejemplos de programas para la operación **READ** con límite de tiempo de espera excedido

Figura	Contenido
Figura 107 a continuación	Especificaciones DDS para el archivo de dispositivo de pantalla
Figura 109	Especificaciones de descripción de archivo, entrada y cálculo

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A*  NOMBR ARCH : HOTELDSP *
A*  DESCRIPCION: ARCHIVO DE PANTALLA PARA EJEMPLO DE LIMITE DE *
A      TIEMPO DE ESPERA EXCEDIDO *
A*****
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++*****
A                               INVITE
A      R REQUEST
A                               OVERLAY
A      ROOM          5A  I 10 46DSPATR(HI)
A                               10 26'Entrar Número Habitación'
A                               DSPATR(HI)
```

Figura 107. Operación de lectura de las DDS con límite de tiempo de espera excedido

```
-----
Entre Número de Habitación: 10025
-----
```

Figura 108. Pantalla de Ejemplo para Límite de Tiempo de Espera Excedido

La Figura 109 muestra un ejemplo de especificaciones de descripción de archivo, entrada y cálculo para una operación **READ** con límite de tiempo de espera excedido.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*****
F*  ID PROGRAMA      - TIEMPO ESPERA EXCEDIDO *
F*  NOMBRE PROGRAMA - TIEMPO EXCEDIDO EN LECTURA *
F*****
FNombarchIPEAF....LregLC1AIDePcLaEDispos+.....KSalidaEntrada....U1.*
FHOTELDSPCF  E                               WORKSTN
F                               KNUM          1
F                               KINFDS FEEDBK

IDsnombre..NODsSal-arch++++.....NvecLon+.....*
```

```

IFEEDBK      DS
I.....campoexter.....PDesdA+++DCampo+.....*
I                                *STATUS STATUS

CL0N01N02N03Factor1+++OpcdeFactor2+++ResultLenDHHiLoEqComments+++++*
C                                WRITEREQUEST
C                                READ HOTELDSP                                9950
C                                EXSR ERRCHK
C                                MOVE '1'                                *INLR
C*****
C*      SUBROUTINA - ERRCHK                                          *
C*      PROPOSITO - COMPROBAR ESTADO PARA ESPERA MAXIMA            *
C*****
C                                ERRCHK BEGSR
C                                STATUS IFEQ 1331
C                                MOVE 'SIGNOFF' CMD                                7
C                                Z-ADD7                                LEN                                155
C                                CALL 'QCMDEXC'
C                                PARM                                CMD
C                                PARM                                LEN
C                                END
C                                ENDSR

```

Figura 109. Especificaciones de descripción de archivo, entrada y cálculo para una operación de lectura con límite de tiempo de espera excedido

Este programa hace que la estación de trabajo termine la sesión cuando no se lleva a cabo ninguna actividad en la estación de trabajo en el transcurso de tiempo especificado.

- En las DDS para el archivo de pantalla **HOTELDSP**, está especificada la palabra clave **INVITE** para todos los formatos. El usuario especifica el período de tiempo de espera mediante el parámetro **WAITRCD** en el mandato **CRTDSPF** (o **CHGDSPP**) para crear (o cambiar) este archivo.
- En las especificaciones de archivo, el archivo **HOTELDSP** está especificado como un archivo **WORKSTN** con la opción **NUM**. El RPG trata el archivo como un archivo de múltiples dispositivos.
- En las especificaciones de entrada, el subcampo ***STATUS** de la estructura de datos de información de archivo se llama **STATUS**.
- La operación **WRITE** sitúa al formato **REQUEST** en la estación de trabajo y, a causa de la palabra clave **INVITE**, convierte a la estación de trabajo en un dispositivo invitado.
- La operación **READ por nombre de archivo** al archivo **HOTELDSP** esperará una respuesta desde el dispositivo invitado durante el período de tiempo especificado en el parámetro **WAITRCD**.
- Si no se recibe respuesta durante este tiempo, se activa el indicador de error 99 y el programa continúa con la siguiente operación.
- La siguiente operación ejecuta la subrutina **ERRCHK**. Esta subrutina comprueba el subcampo **STATUS** de la estructura de datos de información de archivo. El código de estado 1331 indica que se ha excedido el tiempo de espera de la operación **READ** y que la subrutina **ERRCHK** desconecta la estación de trabajo. Otros códigos de estado producen resultados distintos.

Nota: Este ejemplo no es un programa completo.

9.0 Capítulo 9. *Formatos de campos de datos y estructuras de datos*

Este capítulo describe cómo el programa en RPG/400 trabaja con datos almacenados en campos de archivos de datos. Dentro de los archivos, los campos pueden agruparse en estructuras de datos.

Subtemas

- 9.1 Formato de campos en archivos
- 9.2 Estructuras de datos
- 9.3 Estructuras de datos especiales
- 9.4 Especificaciones de subcampos de estructuras de datos
- 9.5 Ejemplos de estructura de datos

9.1 Formato de campos en archivos

Los campos de entrada y salida de un programa en RPG/400 pueden presentar un formato tipo carácter, decimal con zona, decimal empaquetados o binario. Sólo puede especificarse un signo a la derecha o a la izquierda con el formato decimal con zona. El compilador convierte todos los campos de entrada numéricos (a menos que se encuentren en una estructura de datos) a formato decimal empaquetado para el proceso interno. El programa se ejecuta de la misma forma tanto si los datos numéricos están en formato decimal empaquetado, decimal con zona o binario. Sin embargo, el sistema procesa los cálculos aritméticos de forma más eficaz si los datos están en formato decimal empaquetado. Los subcampos dentro de una estructura de datos se tratan siempre en el formato especificado por la especificación del subcampo.

Subtemas

- 9.1.1 Formato decimal empaquetado
- 9.1.2 Formato decimal con zona
- 9.1.3 Formato binario
- 9.1.4 Signos

9.1.1 Formato decimal empaquetado

Formato decimal empaquetado significa que cada byte de almacenamiento (excepto para el byte de orden inferior) puede contener dos números decimales. Cada byte (excepto para el byte de orden inferior) se divide en dos partes de dígitos de 4 bits. El byte de orden inferior contiene un dígito en la parte más a la izquierda y el signo (+ o -) en la parte más a la derecha. Se emplean signos estándar: F hexadecimal para números positivos y D hexadecimal para números negativos. El formato decimal empaquetado tiene el siguiente aspecto:

IMAGEN 22

La parte del signo del byte de orden inferior indica si el valor numérico representado en las partes de dígito es positivo o negativo. La Figura 110 en el tema 9.1.3.2 muestra cuál sería el aspecto del número decimal 8191 en formato decimal empaquetado.

En un archivo descrito en el programa, especifique los campos decimales empaquetados de entrada, de salida y de matrices o tablas, con las entradas siguientes:

Campo decimal empaquetado de entrada: Especifique **P** en la posición 43 de las especificaciones de entrada.

Campo decimal empaquetado de salida: Especifique **P** en la posición 44 de las especificaciones de salida. Esta posición debe estar en blanco si se especifica edición.

Campo decimal empaquetado de matriz o de tabla: Especifique **P** en la posición 43 o en la posición 55 de las especificaciones adicionales. Las matrices y las tablas que se cargan en tiempo de compilación no pueden estar en formato decimal empaquetado.

Para un archivo descrito externamente, el formato de los datos se especifica en la posición 35 de las especificaciones de descripción de datos.

Utilice la fórmula siguiente para hallar la longitud en dígitos de un campo decimal empaquetado:

$$\text{Número de dígitos} = 2n - 1,$$

...en la que **n = número de posiciones de registro de entrada empaquetado que se utilizan.**

Esta fórmula proporciona el número máximo de bytes que puede representar en el formato decimal empaquetado; el límite superior es 30.

Los campos empaquetados pueden tener hasta 16 bytes de longitud. El diagrama en Tabla 13 muestra el empaquetado para campos decimales con zona de hasta 16 dígitos de longitud:

Tabla 13. Equivalencias de empaquetado para campos decimales de hasta 16 dígitos de longitud	
Decimal con zona Longitud en dígitos	Número de bytes utilizados en el campo decimal empaquetado
1	1
3	2
5	3
.	.
.	.
.	.
29	15
30	16

Por ejemplo, un campo de entrada leído en formato decimal empaquetado tiene una longitud de cinco posiciones (tal como se ha especificado en las especificaciones de entrada o de descripción de datos). El número de dígitos en este campo es igual a $2(5) - 1$, o sea, 9. Por lo tanto, cuando utilice el campo en las especificaciones de cálculo, el campo del resultado deberá ser de nueve posiciones de longitud.

Cuando un campo decimal empaquetado en un programa se convierte a un campo decimal con zona en otro programa, el campo decimal con zona contiene siempre un número impar de bytes. Si un campo está en formato decimal empaquetado en un programa y a continuación se desempaqueta en otro programa, la longitud del campo puede incrementarse en 1. Si un campo está empaquetado y se desempaqueta posteriormente en el mismo programa, la longitud del campo no varía. Esto debe tenerse en cuenta cuando se empaqueten campos para el almacenamiento en un dispositivo intermedio y otro programa los utilice posteriormente.

9.1.2 Formato decimal con zona

El formato decimal con zona significa que cada byte de almacenamiento puede contener un dígito o un carácter. Cualquier campo de carácter o numérico puede leerse en formato decimal con zona. En formato decimal con zona, cada byte de almacenamiento se divide en dos partes: una parte de zona de 4 bits y una parte de dígito de 4 bits.

El formato decimal con zona tiene el siguiente aspecto:

IMAGEN 23

La parte de zona del byte de orden inferior indica el signo (+ o -) del número decimal. Se emplean los signos estándar: F hexadecimal para los números positivos y D hexadecimal para los números negativos. En formato decimal con zona, cada dígito de un número decimal incluye una parte de zona; sin embargo, sólo la parte de zona de orden inferior se utiliza como signo. La Figura 110 en el tema 9.1.3.2 muestra el aspecto del número 8191 en formato decimal con zona.

También debe tener en cuenta las variaciones de longitud de campo cuando codifique la posición final en las posiciones 40 a 43 de las especificaciones de salida. Para hallar la longitud de campo después de que este se haya empaquetado, utilice la siguiente fórmula:

$$\text{Longitud de campo} = \frac{n}{2} - 1$$

...en la que n = número de dígitos del campo decimal con zona.

(Se ignora cualquier resto de la división.)

En un archivo descrito por programa, se especifica el formato decimal con zona mediante un blanco en la posición 43 de las especificaciones de entrada, en la posición 44 de las especificaciones de salida o en la posición 43 o 45 de las especificaciones adicionales. En un archivo descrito externamente, el formato de los datos se especifica en la posición 35 de las especificaciones de descripción de datos.

El RPG/400 convierte internamente los datos decimales con zona en datos de tipo carácter. Durante esta conversión, se corrigen automáticamente los errores de los datos decimales. Los errores de los datos decimales sólo pueden detectarse en los campos definidos en formato decimal empaquetado.

9.1.3 *Formato binario*

Formato binario significa que el signo (+ o -) está en el bit más a la izquierda del campo y el valor entero en los restantes bits del campo. Los números positivos tienen un cero en el bit de signo; los números negativos tienen un uno en el bit de signo y están en forma de complemento a dos. En formato binario, cada campo debe tener 2 o 4 bytes de longitud.

Subtemas

- 9.1.3.1 Archivo descrito por programa
- 9.1.3.2 Archivo descrito externamente

9.1.3.1 Archivo descrito por programa

A cada campo de entrada leído en formato binario, el compilador le asigna una longitud de campo (número de dígitos). A un campo binario de 2 bytes se le asigna una longitud de cuatro; a un campo binario de 4 bytes se le asigna una longitud de nueve. Debido a estas limitaciones de longitud, el valor decimal mayor que puede asignarse a un campo binario de 2 bytes es de 9999 y el valor decimal mayor que puede asignarse a un campo binario de 4 bytes es de 999 999 999.

Para los archivos descritos por programa, especifique los campos de entrada binario, salida binario y matriz binario o tabla con las entradas siguientes:

- Campo de entrada binario:* Especifique una **B** en la posición 43 de las especificaciones de entrada.
- Campo de salida binario:* Especifique una **B** en la posición 44 de las especificaciones de salida. Esta posición debe estar en blanco si se especifica edición.

La longitud de un campo que haya de grabarse en formato binario no puede exceder de nueve dígitos. Si la longitud del campo es de uno a cuatro dígitos, el compilador asume una longitud de campo de 2 bytes. Si la longitud del campo es de cinco a nueve dígitos, el compilador asume una longitud de campo binario de 4 bytes.

Ya que el compilador convierte un campo de entrada de 2 bytes en formato binario a un campo decimal de cuatro dígitos, es posible que el valor de entrada sea demasiado grande. Si esto sucede, se pierde el dígito más a la izquierda del número. Por ejemplo, un campo de entrada tiene un valor binario de hexadecimal 7000. El compilador convierte este valor a 28 672 en decimal. El 2 se pierde y el resultado es 8672.

- Campo de matriz o tabla binario:* Especifique una **B** en la posición 43 y/o 55 de las especificaciones adicionales. Las matrices y tablas que se cargan en tiempo de compilación no pueden estar en formato binario.

9.1.3.2 Archivo descrito externamente

En un archivo descrito externamente, el formato de los datos se especifica en la posición 35 de la especificaciones de descripción de datos. El número de dígitos del campo es exactamente el mismo que la longitud en la descripción DDS. Por ejemplo, si define un campo binario en la especificación DDS para que tenga 7 dígitos y 0 posiciones decimales, el compilador del RPG/400 maneja los datos de la siguiente forma :

1. Se define el campo como un campo binario de 4 bytes en la especificación de entrada
2. Se genera un campo Packed(7,0) para el campo en el programa en RPG/400.

Si desea retener toda la información del campo binario, vuelva a definir el campo como un subcampo binario en una estructura de datos.

La Figura 110 muestra el aspecto que tiene el número decimal 8191 en varios formatos.

IMAGEN 24

Figura 110. Representación binaria, empaquetada y decimal con zona del número 8191

(1) Si se lee 8191 dentro del almacenamiento como un campo decimal con zona, ocupa 4 bytes. Si se convierte a formato decimal empaquetado, ocupa 3 bytes. Cuando se vuelve a convertir a formato decimal con zona, ocupa 5 bytes.

(2) Para obtener el valor numérico de un número binario positivo añada los valores de los bits que están en (1), no incluya el bit del signo. Para obtener el valor numérico de un número negativo en binario, añada los valores de los bits que están desactivados (0) más uno (no se incluye el bit del signo).

9.1.4 Signos

El programa en RPG/400 asegura que un signo más o menos consistente esté presente en todos los campos numéricos. Los signos estándares para todos los campos empaquetados y numéricos con zona son la F hexadecimal F para más y la D hexadecimal para menos.

Subtemas

9.1.4.1 Formatos externos

9.1.4.2 Formato interno

9.1.4.1 Formatos externos

Cuando se graba un signo para campos numéricos, el signo (+ o -) se incluye en la posición de unidades del campo de datos a no ser que se haya efectuado la edición. Consulte la publicación *RPG/400 Reference*.

Puede especificar un formato de registro alternativo para el formato decimal con zona. En el formato de signo alternativo, el campo numérico va inmediatamente precedido o seguido por un signo + o -. Un signo más es un hexadecimal 4E, y un signo menos es un hexadecimal 60.

Para archivos descritos por programa, especifique el signo más o menos precedente (entrada **L**) o siguiente (entrada **R**) en las siguientes posiciones:

Campo de entrada: Posición 43 de las especificaciones de entrada

Campo de salida: Posición 44 de las especificaciones de salida

Campo de matriz o tabla: Posición 43 y/o posición 55 de las especificaciones adicionales.

Cuando se especifica un formato de signo alternativo, la longitud del campo debe incluir una posición adicional para el signo. Por ejemplo, si un campo tiene 5 dígitos de longitud y se especifica formato de signo alternativo, debe especificarse una longitud de campo de 6 posiciones.

9.1.4.2 *Formato interno*

Todos los campos numéricos, excepto los subcampos de una estructura de datos, se almacenan en un formato decimal empaquetado para el proceso interno. En formato decimal empaquetado, el signo se almacena en los últimos 4 bits del byte de más a la derecha del campo. Consulte la Figura 110 en el tema 9.1.3.2.

9.2 Estructuras de datos

El programa en RPG/400 le permite definir un área en almacenamiento y la distribución de los campos dentro de la misma, denominados subcampos. Este área en almacenamiento recibe el nombre de estructura de datos. Puede utilizar una estructura de datos para:

- Definir la misma área interna múltiples veces utilizando formatos de datos distintos
- Operar con un campo y cambiar su contenido
- Dividir un campo en subcampos sin utilizar los códigos de operación **MOVE** o **MOVEL**
- Definir una estructura de datos y sus subcampos de la misma forma que se define un registro
- Definir múltiples apariciones de un grupo de datos
- Agrupar datos no contiguos en posiciones del almacenamiento interno contiguas.

Además existen tres estructuras de datos especiales, cada una con una finalidad específica:

- Una estructura de datos de área de datos (identificada por una **U** en la posición 18 de la sentencia de estructura de datos)
- Una estructura de datos de información de archivo (a la que se hace referencia mediante la palabra clave INFDS en una línea de continuación de las especificaciones de descripción de archivos)
- Una estructura de datos de estado del programa (identificada mediante una **S** en la posición 18 de la sentencia de estructura de datos).

Las estructuras de datos pueden describirse en el programa o externamente.

Una estructura de datos descrita por programa se identifica mediante un blanco en la posición 17 de la sentencia de estructura de datos. Las especificaciones de los subcampos para una estructura de datos descrita en el programa tienen que seguir inmediatamente a la sentencia de la estructura de datos.

Una estructura de datos descrita externamente, identificada mediante una **E** en la posición 17 de la sentencia de estructura de datos, es aquella cuyas descripciones de subcampos están contenidas en un archivo descrito externamente que contiene un formato de registro. En el tiempo de compilación, el programa en RPG/400 utiliza el nombre externo para localizar y obtener la descripción externa de los subcampos de la estructura de datos. En el programa puede asignarse un nuevo nombre a un nombre de subcampo externo y pueden añadirse subcampos adicionales a una estructura de datos descrita externamente.

Consulte algunos ejemplos de estructuras de datos en "Ejemplos de estructura de datos" en el tema 9.5.

Subtemas

- 9.2.1 Formato de subcampos de estructura de datos en almacenamiento
- 9.2.2 Especificaciones de sentencias de estructuras de datos
- 9.2.3 Estructuras de datos de apariciones múltiples

9.2.1 Formato de subcampos de estructura de datos en almacenamiento

Los subcampos en una estructura de datos se almacenan en el formato especificado en la posición 43 de las especificaciones de subcampo de la estructura de datos. Las posibles entradas para una estructura de datos descrita en el programa son:

Entrada	Explicación
Blanco	El subcampo está en formato decimal con zona o son datos de tipo carácter, dependiendo de la entrada en la posición 52 de las especificaciones del subcampo.
P	El subcampo está en formato decimal empaquetado
B	El subcampo está en formato binario

Debido a que los subcampos en una estructura de datos se mantienen en el formato especificado, el compilador genera las conversiones necesarias para ejecutar la función requerida. Estas conversiones pueden tener lugar en los momentos siguientes:

- Cuando está leyéndose un registro
- En tiempo de cálculo de detalle o de totales
- En tiempo de salida de detalle o de totales

Las reglas para determinar la longitud de un subcampo en formato decimal empaquetado, formato decimal con zona y formato binario, son las mismas que aquellas para determinar la longitud de un campo en formato decimal empaquetado, formato decimal con zona y formato binario. (Consulte "Formato decimal empaquetado" en el tema 9.1.1, "Formato decimal con zona" en el tema 9.1.2, y "Formato binario" en el tema 9.1.3.)

9.2.2 Especificaciones de sentencias de estructuras de datos

Las sentencias de las estructuras de datos se definen en las especificaciones de entrada y tienen que seguir a todas las especificaciones de entrada de registros. Las especificaciones para las sentencias de estructuras de datos son:

Tabla 14. Especificaciones para sentencias de estructuras de datos	
Posición	Entrada
6	I
7-12	Nombre de la estructura de datos que se está definiendo. Esta entrada es optativa para una estructura de datos descrita por programa y es requerida para una estructura de datos descrita externamente, una estructura de datos de información de archivo (INFDS) y una estructura de datos de área de datos.
13-16	Blanco
17	<i>Blanco:</i> Estructura de datos descrita por programa <i>E:</i> Estructura de datos descrita externamente. Las definiciones de subcampos de la estructura de datos se recuperan de un formato de registro descrito externamente.
18	<i>Blanco:</i> No es una estructura de estado del programa, del área de datos o de los datos inicializados. I: Estructura de datos inicializados globalmente <i>S:</i> Estructura de datos del estado del programa. <i>U:</i> Estructura de datos del área de datos
19-20	DS
21-30	<i>Blanco:</i> La estructura de datos es descrita por programa. <i>Entrada:</i> Este es el nombre del archivo cuyo primer formato de registro contiene las descripciones de campo utilizadas como descripciones de subcampo para esta estructura de datos.
31-43	Blanco
44-47	<i>Blanco:</i> Estructura de datos de una sola aparición <i>nnnn:</i> Un número (ajustado por la derecha) que indica el número de apariciones de la estructura de datos. Nota: Esta entrada tiene que estar en blanco para una estructura de datos de área de datos, para una estructura de datos de información de archivo y para una estructura de datos de estado del programa.
48-51	Longitud de la estructura de datos (optativa). Esta entrada tiene que estar ajustada por la derecha.
52-74	Blanco

Subtemas

9.2.2.1 Reglas para la especificación de sentencias de estructuras de datos

9.2.2.1 Reglas para la especificación de sentencias de estructuras de datos

Al especificar sentencias de estructuras de datos, recuerde lo siguiente:

- El nombre de la estructura de datos tiene que ser un nombre simbólico con un máximo de seis caracteres. El nombre puede aparecer en sólo una especificación de estructura de datos, no puede ser un campo de consulta anticipada y puede especificarse en cualquier parte en donde esté permitido un campo de caracteres.
- Todas las entradas para una estructura de datos y sus subcampos tienen que aparecer juntas; no pueden estar mezcladas con entradas de otras estructuras de datos.
- La longitud de la estructura de datos está determinada por la primera especificación del programa que defina una longitud de una de las formas anteriores. Las longitudes posteriores que entren en conflicto con ésta no son válidas. La longitud de una estructura de datos de una de las siguientes:
 - La longitud especificada en las especificaciones del campo de entrada si el nombre de la estructura de datos es un campo de entrada.
 - La longitud especificada en las posiciones 48 a 51 de la sentencia de la estructura de datos
 - La posición A más alta de un subcampo dentro de una estructura de datos si el nombre de la estructura de datos no es un campo de entrada.
- No se puede utilizar una matriz de tiempo de compilación o de tiempo de preejecución en una estructura de datos del área de datos o en una estructura de datos de apariciones múltiples.
- Las estructuras de datos están formadas por datos de tipo carácter con una longitud de 1 a 9999 caracteres.
- Una estructura de datos y un subcampo de una estructura de datos no pueden tener el mismo nombre.

9.2.3 Estructuras de datos de apariciones múltiples

Una estructura de datos de apariciones múltiples es una estructura de datos cuya definición se repite en un programa para formar una serie de estructuras de datos con formatos idénticos. Especifique el número de apariciones de la estructura de datos en las posiciones 44 a 47 de la sentencia de estructura de datos. Cuando las posiciones 44 a 47 no contienen una entrada, la estructura de datos no es una estructura de datos de apariciones múltiples. Todas las apariciones de una estructura de datos tienen los mismos atributos y se puede hacer referencia a ellas individualmente. El código de operaciones **OCUR**, que solamente puede utilizarse con estructuras de datos de apariciones múltiples, le permite especificar qué aparición de una estructura de datos se utiliza para las operaciones subsiguientes del programa.

Nota: No se permiten apariciones múltiples para estructuras de datos de estado del programa, de información de archivo o de área de datos.

Consulte algunos ejemplos de estructuras de datos de apariciones múltiples en "Ejemplos de estructura de datos" en el tema 9.5.

9.3 Estructuras de datos especiales

Las estructuras de datos especiales incluyen:

- Estructuras de datos de área de datos
- Estructuras de datos de información de archivo (INFDS)
- Estructuras de datos de estado del programa.

Subtemas

- 9.3.1 Estructura de datos de área de datos
- 9.3.2 Estructura de datos de información de archivo
- 9.3.3 Estructura de datos de estado del programa

9.3.1 Estructura de datos de área de datos

Una estructura de datos de área de datos, identificada por una **U** en la posición 18 de la sentencia de la estructura de datos, indica al programa en RPG/400 que debe leer y bloquear el área de datos de igual nombre en la inicialización del programa y debe grabar y desbloquear la misma área al finalizar el programa. El tipo de datos en las estructuras de datos de área de datos, así como en todas las demás estructuras de datos, es de caracteres. Un área de datos leída dentro de una estructura de datos del área de datos tiene que ser también de caracteres. El área de datos y la estructura de datos del área de datos tiene que tener el mismo nombre a menos que asigne un nuevo nombre al área de datos dentro del programa en RPG/400 utilizando la sentencia ***NAMVAR DEFN**.

Puede especificar operaciones de área de datos (**IN**, **OUT**, y **UNLCK**) y tener el tipo para un área de datos implícitamente leída y grabada. Antes de utilizar estas operaciones en una estructura de datos de área de datos, tiene que especificar esa área de datos en el campo del resultado de una sentencia ***NAMVAR DEFN**.

Una estructura de datos del área de datos no se puede especificar en el campo del resultado de una operación **PARM**.

Si especifica blancos para la estructura de datos del área de datos (posiciones 7 a 12 de la línea de especificaciones de entrada que contiene una **U** en la posición 18), el programa en RPG/400 utiliza un área de datos local. Para proporcionar un nombre a un área de datos local, utilice la operación ***NAMVAR DEFN** con ***LDA** en el factor 2 y el nombre del campo del resultado.

Para obtener información general acerca de las áreas de datos, consulte el Capítulo 11, "Comunicación con objetos en el sistema".

9.3.2 Estructura de datos de información de archivo

Puede especificar una estructura de datos de información de archivo (definida por la palabra clave **INFDS** en una línea de continuación de las especificaciones de descripción de archivo) para cada archivo del programa. Esto le proporciona información sobre el estado de las condiciones de excepción/error que hayan ocurrido. El nombre de la estructura de datos de información de archivo tiene que ser único para cada archivo. Una estructura de datos de información de archivo contiene subcampos predefinidos que proporcionan información sobre las condiciones de excepción/error que hayan ocurrido. Consulte en "Manejo de excepciones/errores" en el tema 4.8 las estructuras de datos de información de archivo y sus subcampos.

Defina y asigne nombres para una estructura de datos de información de archivo en una línea de continuación de las especificaciones de descripción de archivo mediante las entradas siguientes:

Tabla 15. Entradas para definir y asignar nombre a una estructura de datos de información de archivo	
Posición	Entrada
6	F
7-52	Blancos (si la información está especificada en una línea de continuación distinta)
53	K (indica una línea de continuación)
54-59	INFDS (identifica esta estructura de datos como la estructura de datos de información de archivo)
60-65	nombre de la estructura de datos de información de archivo.

9.3.3 Estructura de datos de estado del programa

Una estructura de datos de estado del programa se identifica mediante una **S** en la posición 18 de la sentencia de estructura de datos, se utiliza para proporcionar al programa información sobre las condiciones de excepción/error del programa. Consulte en "Manejo de excepciones/errores" en el tema 4.8 las estructuras de datos de estado del programa y sus subcampos predefinidos.

9.4 Especificacions de subcampos de estructuras de datos

Los subcampos de una estructura de datos descrita por el programa tienen que ir inmediatamente después de la sentencia de especificación de estructura de datos a la que se hacen referencia. Los subcampos de una estructura de datos descrita externamente se describen de forma externa al programa en RPG/400. Las especificaciones de los subcampos se incorporan al programa en RPG/400 en tiempo de compilación. Puede asignarse un nuevo nombre a los subcampos de una estructura de datos descrita externamente o pueden aparecer especificaciones de subcampos a continuación de la sentencia de estructura de datos. Todos los subcampos externos a los que se haya asignado un nuevo nombre que se hayan inicializado deben preceder a cualquier especificación de subcampo adicional. Para añadir subcampos a una estructura de datos descrita externamente, se siguen las mismas reglas que para los subcampos de una estructura de datos descrita en el programa. Los subcampos descritos internamente se añaden a las descripciones recuperadas.

Las especificaciones para los subcampos son:

Tabla 16. Especificaciones para subcampos	
Posición	Entrada
6	I
7	Blanco
8	I : Indica un subcampo inicializado. (Especifique el valor de inicialización en la posición 21-42 o deje en blanco en caso de valor de inicialización por omisión)
9-20	Blanco
21-42	posiciones 21-26: Valor de inicialización de la constante con nombre si la posición 8 contiene una I . Deje en blanco las posiciones restantes. o posiciones 21-42: Valor de inicialización del literal si la posición 8 contiene una I . o posiciones 21-42: Blancos como valor de inicialización por omisión si la posición 8 contiene una I . o posiciones 21-30: Nombre externo para asignar nombre a un subcampo en una estructura de datos descrita externamente. (Especifique el nombre que se va a utilizar en el programa en las posiciones 53 a 58). Deje en blanco las posiciones restantes.
43	P : Indica que el subcampo tiene formato decimal empaquetado. B : Indica que el subcampo tiene formato binario. Blancos : Indica que el subcampo tiene formato decimal con zona o está formado por datos de tipo carácter.
44-47 48-51	Números de 1 a 4 dígitos : Las posiciones 44 a 47 contienen la posición inicial y las posiciones 48 a 51 contienen la posición final del subcampo. Estas entradas tienen que estar ajustadas por la derecha, pudiendo omitirse los ceros iniciales. o Palabras clave : Para una estructura de datos de estado del programa o una estructura de datos de información de archivo (INFDS), ponga una palabra clave especial (INFDS) ajustada por la izquierda en esta posición. Una palabra clave puede empezar en la posición 44 y continuar hasta la posición 51. Consulte en "Manejo de excepciones/errores" en el tema 4.8 las palabras clave y sus descripciones.
52	0-9 : Indica el número de posiciones decimales en un campo numérico o en una matriz.

Blancos: Indica un campo de tipo carácter.

Nota: Esta posición tiene que contener una entrada para un subcampo numérico. Sin embargo, no se requiere una entrada para una matriz. Si se efectúa una entrada para una matriz, la entrada tiene que ser igual a la que se efectúa en las especificaciones adicionales.

53-58

El nombre del subcampo.

Nota: Si se especifica una matriz como nombre de subcampo, la longitud indicada en las posiciones 44 a 51 debe ser igual a la cantidad de almacenamiento que se requiere para almacenar la matriz completa (por ejemplo, 10 elementos de media palabra binaria requieren 20 bytes de almacenamiento).

59-74

Blanco

Subtemas

9.4.1 Reglas para especificaciones de subcampos

9.4.1 Reglas para especificaciones de subcampos

Al especificar especificaciones de subcampos recuerde lo siguiente:

- Si la longitud (posiciones 44 a 51) o posiciones decimales (posición 52) del subcampo difieren de anteriores definiciones efectuadas en el programa, se utiliza la primera definición, no siendo válidas las subsiguientes definiciones conflictivas.
- Si la posición A (48 a 51) especificada para un subcampo es mayor que la longitud que se ha definido para el campo de entrada del mismo nombre o que la longitud definida de la estructura de datos, no es válida la especificación del subcampo.
- Para redefinir subcampos, tienen que especificarse las mismas o parte de las mismas posiciones Desde y A (posiciones 44 a 51) para otro subcampo de la misma estructura de datos.
- Para definir un subcampo de una sola posición, entre el mismo valor en las posiciones 44 a 47 y 48 a 51.
- No pueden utilizarse subcampos solapados en la misma especificación de cálculo.
- Si se especifica una matriz o un elemento de matriz con un índice variable en el factor 1, factor 2 o campo del resultado de una especificación de cálculo, se utiliza la totalidad de la matriz para determinar si existe solapamiento.
- Tiene que asegurarse de que los campos numéricos empaquetados, con zona o numéricos binarios se inicializan con datos numéricos antes de utilizarlos en operaciones aritméticas o de edición.
- Un nombre de campo de entrada no puede:
 - Aparecer como un nombre de subcampo y como nombre de la estructura de datos
 - Aparecer más de una vez como nombre de subcampo.
- En las operaciones de cálculo siguientes se efectúa una comprobación para determinar si se solapan los subcampos:
 - El factor 1 y el campo del resultado, y el factor 2 y el campo del resultado en las operaciones **ADD**, **SUB**, **MULT**, **DIV**, **Z-ADD**, y **Z-SUB**. El factor 1 y el factor 2 de las operaciones precedentes pueden solaparse.
 - Se comprueban el factor 2 y el campo del resultado de las operaciones **MOVE**, **MOVEL**, o **MOVEA** para determinar si existe solapamiento.
 - Se comprueba el factor 2 y el campo del resultado y el factor 1 y el campo del resultado de una operación **PARM** para detectar si existe solapamiento.

9.5 Ejemplos de estructura de datos

La Figura 111 a la Figura 116 muestran algunas formas típicas de utilizar estructuras de datos.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
IFILEIN NS 01 1 CA 2 CB
I.....PDesdeA+++DNomcamL1M1FrP1NeCe...*
I 3 18 NUMPIE
I 19 29 NOMBRE
I 30 40 NUPAT
I 41 61 DIB
IDsnombre..NODsSal-arch++.....NvecLon+.....*
INUMPIE DS
I.....campoexter.....PDesdA+++DCampo+.....*
I 1 4 FAB
I 5 10 COLOR
I 11 13 CONSIG
I 14 16CANT
I*
```

Figura 111. Utilización de una estructura de datos para definir subcampos dentro de un campo

Se puede hacer referencia a los subcampos de la estructura de datos por el nombre **NUMPIE** o por los subcampos **FAB**, **COLOR**, **CONSIG**, o **CANT**.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
ITRANSACTNS 01 1 C1 2 C2
I.....PDesdeA+++DNomcamL1M1FrP1NeCe...*
I 3 10 NUMPIE
I 11 16CANTID
I 17 20 TIPO
I 21 21 CODIGO
I 22 25 SITUAC
IDsnombre..NODsSal-arch++.....NvecLon+.....*
IPRTKEY DS
I.....campoexter.....PDesdA+++DCampo+.....*
I 1 4 SITUAC
I 5 12 NUMPIE
I 13 16 TIPO
I*
```

Figura 112. Utilización de una estructura de datos para agrupar campos

Cuando utiliza una estructura de datos para agrupar campos, puede hacerse que campos que ocupan posiciones no adyacentes en el registro de entrada, ocupen posiciones internas adyacentes. En este caso, puede hacerse referencia al área mediante el nombre de la estructura de datos y/o el nombre de los subcampos individuales.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I*
I* Se utiliza una estructura de datos de apariciones múltiples para
I* acumular una serie de totales por código específico, grabándose
I* los totales de cada una de las apariciones de la estructura de
I* datos.
I* La estructura de datos descrita por programa, TOTDS, tiene 99
I* apariciones (posiciones 46 y 47). Puede especificarse la longitud
I* de la estructura de datos en las posiciones 48 a 51.
I*
IDsnombre..NODsSal-arch++.....NvecLon+.....*
ITOTDS DS 99
I.....campoexter.....PDesdA+++DCampo+.....*
I 1 50TOTCNT
I 6 12TOT1
I 13 20TOT2
I*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* Un campo de código numérico, CODIGO, contiene un valor de 01 a 99.
C* Este valor es diferente cada vez que se ejecuta la operación OCUR.
C* Cuando se procesa la operación OCUR, se utiliza el campo CODIGO
C* para establecer la aparición actual de TOTDS. Si la operación
C* OCUR se ejecuta correctamente, el programa bifurca la subrutina
C* RUTSUM donde se efectúa una cuenta de registros y se suman los
```

Guía del usuario RPG/400
Ejemplos de estructura de datos

C* valores de entrada a los subcampos de la estructura de datos. Si
C* el campo CODIGO contiene un valor no comprendido entre 01 y 99, se
C* activa el indicador 25 y el programa bifurca a CODERR.

```
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C          CODIGO      OCUR TOTDS          25
C  25          GOTO CODERR
C          EXSR RUTSUM          +-----+
C          "                |  Cálculos  |
C          "                +-----+
C          CODERR      TAG
C          "                +-----+
C          "                |  Cálculos  |
C          "                +-----+
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

C*

C* Cuando se tienen que grabar los totales para los códigos específicos
C* en estructura de datos de apariciones múltiples, se utiliza la
C* salida de excepción. La operación EXCPT PRTHDG produce que se
C* graben todas las líneas de excepción de las especificaciones de
C* salida con el nombre de PRTHDG. La opción efectuar grupo
C* inicialmente establece el campo X en 1. El valor X establece la
C* aparición actual de TOTDS. La operación Z-ADD añade TOTCNT a un
C* campo de ceros y coloca la suma en el campo de resultados TOTCNT.
C* Si TOTCNT contiene un valor más, se activa el indicador 27.
C* La operación EXCPT PRTHDG produce la grabación de la aparición actual
C* de la estructura de datos. Si se produce un desbordamiento mientras
C* se graba la aparición actual de la estructura de datos, se activa el
C* indicador OF, se produce un salto de página y se graban todas las
C* líneas de excepción en las especificaciones de salida con el nombre
C* PRTHDG. La operación SETOF desactiva el indicador OF.

C*

C* La opción Efectuar grupo continúa el proceso hasta que el campo X es
C* mayor de 99, el número máximo de apariciones para la estructura de
C* datos de apariciones múltiples. Cuando X es mayor que 99, el control
C* pasa a la sentencia siguiente a la sentencia END.

```
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C*
C          EXCPTPRTHDG
C          DO  99          X          30
C  X          OCUR TOTDS
C          Z-ADDTOTCNT    TOTCNT    27
C  27          EXCPTPRTHDG
C  OF          EXCPTPRTHDG
C  OF          SETOF          OF
C          END
C          "                +-----+
C          "                |  Cálculos  |
C          "                +-----+
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

C*

C* La subrutina ADDRTRN actualiza la aparición actual de los subcampos
C* de la estructura de datos de apariciones múltiples.

```
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C          ADDRTRN      BEGSR
C          ADD  1          TOTCNT
C          ADD  FLD1      TOT1
C          ADD  FLD2      TOT2
C*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OIMPRE  E  206          IMPCAB
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O          "          +-----+
O          "          | Entradas Título Informe|
O          "          +-----+
O          "
O  E  2          PRTHDG
O          "          +-----+
O          "          |Entradas Cabecera Columnas Informe|
O          "          +-----+
O          "
O  E          PRTHDG
O          X          10
O          TOTCNTZ    20
O          TOT1  J    35
O          TOT2  J    50
O*
```

Figura 113. Utilización de una estructura de datos para acumular totales -
Ejemplo 1

En el ejemplo siguiente, se utiliza una vez más una estructura de datos de apariciones múltiples, **TOTED**, para acumular una serie de totales por códigos específicos, imprimiéndose los totales de cada una de las apariciones de la estructura de datos. Existen 70 códigos.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
E*
E* En las especificaciones adicionales se ha especificado una matriz
E* MAC, en tiempo de compilación. Esta matriz tiene 70 entradas.
E* Existen 10 entradas en cada registro y cada elemento de la matriz
E* tiene 6 posiciones de longitud. Se utiliza la situación relativa
E* del código alfabético dentro de la matriz (por ejemplo, la entrada
E* número 37) para establecer la aparición actual de la estructura de
E* datos.
E*
E...Arch(de)Arch(a)+NombreN/rN/tbLonPDSNommatLonPDSComentarios++++*
E          MAC      10  70  6          MATRIZ DE CODIGOS

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* La operación Z-ADD establece el campo X en 1. La operación LOKUP
C* comienza por el primer elemento de MAC y efectúa una búsqueda
C* hasta que encuentra el primer elemento igual al código contenido
C* en CODIGA. El campo CODIGA es un campo de caracteres de 6
C* posiciones. El valor del índice X se establece en el número de la
C* posición que ocupa el elemento localizado. Si la operación LOKUP
C* no encuentra un elemento igual a CODIGA, no se activa el
C* indicador 20 y la operación GOTO, condicionada por N20, bifurca a
C* CODERR TAG. Si la operación LOKUP encuentra un elemento igual a
C* CODIGA, la operación OCUR utiliza el valor de X para establecer la
C* aparición actual de TOTED y el programa bifurca a la subrutina
C* RUTSUM, donde se efectúa una cuenta de registros y se suman los
C* valores de entrada a los subcampos de la estructura de datos. La
C* subrutina RUTSUM no se muestra en la figura. Si la aparición está
C* fuera del rango válido para la estructura de datos, se activa el
C* indicador 26 y el programa bifurca a FINIMP TAG.
C*
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios++++*
C          Z-ADD1          X          30
C          CODIGA      LOKUPMAC,X          20
C  N20          GOTO  BADCOD
C          X          OCUR  TOTDS          26
C  26          GOTO  ENDPRT
C          EXSR  ADDRTN
C          "          +-----+
C          "          | Cálculos |
C          CODERR  TAG          +-----+
C          "
C          "
C          FINIMP  TAG          +-----+
C          "          | Cálculos |
C          "          +-----+

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
O*
O* Los cálculos para imprimir la estructura de datos no se muestran.
O* Sólo se muestra una parte de las especificaciones de salida.
O* La sentencia IMPRED utiliza el valor del campo X, que contiene la
O* aparición actual de la estructura de datos, como índice para
O* imprimir el código alfabético correspondiente.
O*
ONombre++DFADSaSdN01N02N03Nomexc.....*
O          E          PRTDS
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O          ARC,X      10
O          TOTCNTZ    20
O          TOT1  J    35
O          TOT2  J    50
```

Figura 114. Utilización de una estructura de datos para acumular totales -
Ejemplo 2

Los dos programas (1 y 2) mostrados en Figura 115 utilizan estructura de datos de área de datos (definida por la U en la posición 18 de las especificaciones de entrada). El programa 1 utiliza los subcampos de la estructura de datos para acumular una serie de totales. El programa 2 utiliza después los totales de los subcampos para efectuar los cálculos.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I* PROGRAMA 1
I*
IDsnombre..NODsSal-arch++.....NvecLon+.....*
ITOTALS      UDS
I.....Ext-field+.....PFromTo++DField+.....*
I              1   82TOTIMP
I              9   182TOTBRU
I              19  282TOTNET
I*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C              "          +-----+
C              "          |  Cálculos  |
C              "          +-----+
C              ADD  AMOUNT  TOTIMP
C              ADD  GROSS   TOTBRU
C              ADD  NET    TOTNET
C*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I* PROGRAMA 2
I*
IDsnombre..NODsSal-arch++.....NvecLon+.....*
ITOTALS      UDS
I.....campoexter.....PDesdA+++DCampo+.....*
I              1   82TOTIMP
I              9   182TOTBRU
I              19  282TOTNET
I*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C              "          +-----+
C              "          |  Cálculos  |
C              "          +-----+
C              IMPOR2  COMP TOTIMP          9191
C              BRUTO2  COMP TOTBRU          9292
C              NETO2   COMP TOTNET          9393
C              "
C              "
C*

```

Figura 115. Estructuras de datos de área de datos

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
IDsnombre..NODsSal-arch++.....NvecLon+.....*
IEDUNO      E DSREGEXT
I.....campoexter.....PDesdA+++DCampo+.....*
I              CARACTER          CARC
I              1   16 CARZO

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A.....T.Nombre+++Lon++TDpB.....Funciones+++++++*
A              R RECORD          TEXT('REG DESCR EXTERNAMENTE')
A              CARACTER          10
A              ZONA              6S 2
A              EMPAQ            4P 0
A              BINARI          4B 0
A*

```

Figura 116. Asignación de nombre a subcampos de una estructura de datos descrita externamente

En la sentencia de estructura de datos mostrada en la Figura 116, las posiciones 7 a 12 contienen el nombre de la estructura de datos que se está definiendo (**EDUNO**), la posición 17 contiene una E para denotar que está descrita externamente y las posiciones 19 y 20 contienen **DS** para denotar que se trata de una estructura de datos. Las posiciones 21 a 30 contienen el nombre del archivo (**REGEXT**) cuyo primer formato de registro contiene las descripciones de campo utilizadas como descripciones de subcampo para esta estructura de datos (**RECORD**).

En la primera especificación de descripción de datos, la posición 17 contiene una R para denotar formato de registro y las posiciones 19 a 28 contienen el nombre del formato de registro (**RECORD**). En las especificaciones de descripción de datos subsiguientes, las posiciones 19

a 28 contienen los nombres de los campos (**CARACTER**, **ZONA**, **EMPAQ**, y **BINARI**).

Para su utilización en el programa, los campos de una estructura de datos también pueden volver a definirse. Los campos **CARACTER** y **ZONA** están descritos también como un campo (**CARZO**) en las especificaciones de entrada.

En el programa en RPG/400 un nombre de campo no puede contener más de 6 caracteres. Por lo tanto, al nombre de campo **CARACTER** se le asigna el nombre **CHAR** en la especificaciones de entrada. Entonces la estructura de datos utiliza **CHAR** como el nombre de subcampo.

Capítulo 10. Constantes con nombre, inicialización y tipos de datos SAA

10.0 Capítulo 10. Constantes con nombre, inicialización y tipos de datos SAA

En este capítulo se describe cómo se pueden utilizar las constantes con nombre y los tipos de datos SAA en el programa en RPG/400. En el capítulo también se trata la inicialización de un programa en RPG/400.

Subtemas

10.1 Constantes con nombre

10.2 Inicialización

10.3 Tipos de datos SAA

10.1 Constantes con nombre

El usuario puede dar un nombre a una constante. Este nombre representa un valor específico que no se puede cambiar cuando se está ejecutando el programa.

Subtemas

10.1.1 Reglas para constantes con nombre

10.1.1 Reglas para constantes con nombre

- Las constantes con nombre pueden especificarse en el Factor 1 y Factor 2 de las especificaciones de cálculo y en los campos Nombre de Campo, Constante o Editar Palabra de las especificaciones de salida. También se pueden utilizar como índices de matrices y como nombre de formato en una especificación de salida **WORKSTN** o como valores de inicialización en una especificación de entrada.
- La constante con nombre no tiene un tipo inherente. Esto es, la definición no implica ninguna precisión. La precisión real se define por el contexto que se especifica.
- La constante con nombre puede definirse en cualquier lugar de las especificaciones de entrada.
- Las constantes de caracteres con nombre deben empezar y finalizar con un apóstrofo (').
- Si se especifica una constante alfanumérica, literal transparente o hexadecimal, entonces puede continuarse en el campo de constante de la línea siguiente codificando un guión (-) al final de la constante en lugar de un apóstrofo. Si se especifica una constante numérica, entonces puede continuarse en el campo de constante de la línea siguiente codificando un guión (-) al final de la constante inmediatamente después del último dígito.
 - El guión puede especificarse en cualquier posición del campo.
 - El guión funciona de la misma manera que el signo menos cuando se continúan los mandatos en los programas en CL. Cualquier blanco del siguiente registro de entrada que siga al apóstrofo inicial y preceda el primer carácter no blanco, se incluye en la constante con nombre.
 - Los guiones están permitidos en la primera posición de un literal de constante con nombre para permitir trasladar los datos de doble byte. Consulte "Traslado de datos de doble byte entre corchetes y supresión de caracteres de control (SUBR40R3)" en el tema 11.2.3 para obtener más información sobre el traslado de datos de doble byte.
 - Para constantes hexadecimales, el número de dígitos hexadecimales de una línea de continuación no necesita ser un número par. Sin embargo, el número total de dígitos hexadecimales de toda la constante debe ser par. (Los literales hexadecimales deben empezar por una X mayúscula, seguida de una comilla simple (X')). Los literales hexadecimales, como las constantes de caracteres, finalizan con una sola comilla (').
 - La siguiente especificación de entrada debe contener una sola entrada en la entrada de la constante (además de una I en la posición 6). Si se continúa una constante alfanumérica, literal transparente o literal hexadecimal, el primer carácter de cada continuación (posición 21) debe contener un apóstrofo.
 - La constante puede continuarse tantas veces como se desee siempre que la longitud total de la constante no exceda los 256 caracteres de un solo byte. Una constante numérica no puede tener una longitud mayor que 30 dígitos, con un máximo de nueve posiciones a la derecha del punto decimal. Un literal hexadecimal no puede tener una longitud mayor de 512 dígitos hexadecimales representando 256 bytes y debe contener un número par de dígitos.
 - La constante con nombre representa a la constante que es la concatenación de todas las constantes de la especificación principal de la constante con nombre y líneas de continuación.

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7...*
I*
I* El siguiente es un ejemplo de una constante con nombre de
I* caracteres:
I*
I*.....Constconnombre+++++++C.....Nomcam.....
I*
I*          'ABCDEF'          C          CHAR
I*
```

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7...*
I*
I* El siguiente es un ejemplo de una constante transparente
```

```

I* continuada. Los caracteres de Desplazamiento a Teclado
I* Ideográfico (SO) y a Teclado Estándar (SI) se representan
I* por una o y una i.
I*
I.....Constconnombre+++++++C.....Nomcam.....
I*
I          'oK1K2K3i-          C          TRANS
I          'oK4K5i'
I*

*...1....+....2....+....3....+....4....+....5....+....6....+....7...*
I*
I* El siguiente es un ejemplo de una constante con nombre de
I* caracteres continuada. El blanco inmediatamente anterior al
I* guión en cada línea y los 3 blancos de la última línea
I* de la constante se incluirán en la constante. El valor de la
I* constante LONGNC será la serie:
I* ESTA ES UNA CONSTANTE LARGA QUE TIENE TRES BLANCOS  AQUÍ
I*
I.....Constconnombre+++++++C.....Nomcam.....
I*
I          'ESTA ES UNA -          C          LONGNC
I          'CONSTANTE LARGA QUE -
I          'TIENE TRES BLANCOS-
I          '  AQUÍ'
I*

*...1....+....2....+....3....+....4....+....5....+....6....+....7...*
I*
I* El siguiente es un ejemplo de una constante numérica
I* continuada.
I*
I.....Constconnombre+++++++C.....Nomcam.....
I*
I          123456-          C          CHAR
I          789
I*

*...1....+....2....+....3....+....4....+....5....+....6....+....7...*
I*
I* El siguiente es un ejemplo de un literal hexadecimal.
I*
I.....Constconnombre+++++++C.....Nomcam.....
I*
I          X'C1F2C3'          C          HEX1
I*

*...1....+....2....+....3....+....4....+....5....+....6....+....7...*
I*
I* El siguiente es un ejemplo de un literal hexadecimal
I* continuado.
I*
I.....Constconnombre+++++++C.....Nomcam.....
I*
I          X'c1-          C          HEX2
I          'c2c3'
I*

```

Figura 117. La utilización de constantes con nombre

10.2 Inicialización

El soporte de inicialización proporcionado por el compilador del RPG/400 consta de tres partes: la subrutina de inicialización, los códigos de operación **CLEAR** y **RESET** y la inicialización de la estructura de datos.

Subtemas

- 10.2.1 Subrutina de inicialización (*INZSR)
- 10.2.2 Códigos de operación CLEAR y RESET
- 10.2.3 Inicialización de estructuras de datos
- 10.2.4 Consideraciones especiales para la inicialización de estructuras de datos
- 10.2.5 Reglas para la inicialización de subcampos
- 10.2.6 Inicialización y el ciclo del programa
- 10.2.7 Ejemplos de inicialización

*10.2.1 Subrutina de inicialización (*INZSR)*

La subrutina de inicialización le permite procesar especificaciones de cálculo antes de la salida **1P**. Se declara como cualquier otra subrutina, pero con el nombre especial ***INZSR** en el factor 1. Esta subrutina se invocará automáticamente al final del paso de inicialización del programa en RPG/400 antes de la salida **1P**. Puede entrar cualquier cálculo que desee en esta subrutina y también puede llamarla explícitamente utilizando un código de operación **EXSR** o **CASxx**.

10.2.2 Códigos de operación CLEAR y RESET

El código de operación **CLEAR** establece una variable o todas las variables de una estructura en blancos, ceros o '0' según el tipo (carácter, numérico o indicador). Si especifica una estructura (formato de registro, estructura de datos o matriz) todos los campos de esa estructura se borran en el orden en el cual están declarados.

El código de operación **RESET** establece una variable o todas las variables de una estructura en su valor inicial. El valor inicial para una variable es el valor que tenía al final del paso de inicialización en el ciclo del RPG/400, después de haberse invocado la subrutina de inicialización. Puede utilizar la inicialización de estructura de datos para asignar valores iniciales a los subcampos y después cambiar los valores durante la ejecución del programa y utilizar el código de operación **RESET** para volver a establecer los valores de los campos en sus valores iniciales. Como el valor inicial es el valor que tenía la variable después de ejecutarse la subrutina de inicialización, puede utilizar la subrutina de inicialización para asignar los valores iniciales a una variable y después utilizar **RESET** para volver a establecer la variable en ese valor inicial. Esto se aplica solamente a la subrutina de inicialización cuando se ejecuta automáticamente como una parte del paso de inicialización.

Para obtener más información sobre la subrutina de inicialización y los códigos de operación **CLEAR** y **RESET** consulte la publicación *RPG/400 Reference*.

10.2.3 Inicialización de estructuras de datos

La inicialización de estructuras de datos le permite inicializar estructuras de datos y subcampos en blanco, cero o un valor específico.

Por omisión, una estructura de datos se considera que es un campo de caracteres y, a menos que se especifique, se inicializa en blancos. Sin embargo, si los subcampos numéricos no se inicializan con datos numéricos antes de que se utilicen en operaciones aritméticas o de edición, se producen errores de datos decimales. La inicialización de estructuras de datos proporciona un medio por el cual los subcampos de las estructuras de datos pueden inicializarse en tiempo de compilación, al principio del paso *INIT, antes de que se ejecute cualquier otra inicialización de programa.

Las estructuras de datos pueden inicializarse globalmente o en base a un subcampo.

Una estructura de datos inicializada globalmente, identificada por una **I** en la columna 18 de la especificación de la estructura de datos, se inicializa con todos los caracteres establecidos en blanco y todos los numéricos establecidos en cero. Debido a que cada subcampo se inicializa por orden de aparición, debe asegurarse de que los campos solapados se declaren en un orden en que se inicialicen correctamente.

Una estructura de datos inicializada en base a un subcampo se identifica por una **I** en la columna 8 y un valor de inicialización para el subcampo en las columnas 21-42 de la especificación del subcampo de la estructura de datos. Si las columnas 21-42 contienen blancos, el subcampo se inicializará en blancos o ceros, según el subcampo sea de caracteres o numérico. Puede especificar un valor literal o un nombre de constante con nombre como el valor de inicialización en un formato similar al de las constantes con nombre. Si las columnas 21-42 contienen una constante con nombre o un literal, el subcampo se inicializará en el valor de inicialización especificado.

Se puede inicializar globalmente una estructura de datos e inicializar individualmente los subcampos de la estructura, especificando una **I** en la columna 18 de la especificación de la estructura de datos y una **I** en la columna 8 de cada especificación de subcampo de la estructura de datos. Los subcampos se inicializan en el mismo orden en el cual se declaran en la estructura de datos.

10.2.4 Consideraciones especiales para la inicialización de estructuras de datos

La estructura de datos de apariciones múltiples se inicializa por valor de subcampo o si se inicializa globalmente la estructura, las apariciones de la estructura se inicializan con el mismo valor.

Se aplican las reglas siguientes al inicializar matrices:

- Si se especifica un valor de inicialización para una matriz en tiempo de ejecución, cada elemento de la matriz se inicializa con el mismo valor. Para especificar valores diferentes para cada elemento de la matriz, debe utilizarse una matriz en tiempo de compilación o en tiempo de preejecución.
- Ya que las matrices de tiempo de compilación o de preejecución se inicializan por definición, no pueden inicializarse utilizando el soporte de inicialización de subcampo. Cuando aparece una matriz de tiempo de compilación o de tiempo de preejecución como parte de una estructura de datos inicializada globalmente, no se incluye como parte de la inicialización global. Las matrices de tiempo de compilación se inicializan en el mismo orden en el cual se declaran sus datos después del programa y las matrices de tiempo de preejecución en el orden en el cual se declaran los archivos de datos de entrada de matrices.
- Si una inicialización de subcampo solapa una matriz de tiempo de compilación o de tiempo de preejecución, la inicialización de la matriz se efectúa la última, sin tener en cuenta el orden de las definiciones.
- Si una definición de subcampo y de matriz de tiempo de ejecución se solapan en una estructura de datos, se inicializarán en el orden en el cual se definen.

Las reglas siguientes se aplican a la inicialización de estructuras de datos especiales:

- Las estructuras de datos del área de datos, por definición, se inicializan leyéndose en el momento de la inicialización del programa, de modo que no se requiere el soporte de inicialización para estas estructuras de datos.
- Otras estructuras de datos, como el área de datos local y el área de datos PIP, se pueden inicializar.
- Debido a que la mayoría de campos de las estructuras de datos de información de archivo y de estructuras de datos de estado de programa se inicializan por el compilador en el momento de la inicialización, no se soporta la inicialización para estas estructuras.

10.2.5 Reglas para la inicialización de subcampos

Las reglas siguientes se aplican a la inicialización de subcampos:

- Un valor de inicialización debe coincidir con el tipo del subcampo y no debe exceder la longitud o el número de posiciones decimales.
- Para continuar un literal en más de una línea, el indicador de valor de inicialización (**I** en la columna 8) se especifica sólo en la primera línea del literal. Todas las demás reglas para continuación de la línea siguen las convenciones utilizadas para continuar las constantes con nombre. Consulte "Constantes con nombre" en el tema 10.1.
- Se puede declarar una constante con nombre, que se utilice como valor de inicialización, antes o después del subcampo donde se utiliza. La constante con nombre debe estar justificada por la izquierda en las columnas 21-26 de la especificación del subcampo.
- Para estructuras de datos descritas externamente:
 - Un valor de inicialización para un subcampo sólo puede especificarse una vez. Si se encuentra más de un valor de inicialización, se utiliza el primer valor especificado. Las demás especificaciones se ignoran y se emiten mensajes de error.
 - Si la especificación de inicialización para un subcampo al que se le ha asignado un nuevo nombre sigue directamente la especificación de asignación de nombre, no se tiene que especificar el nombre de subcampo en la especificación de inicialización.
 - Si va a asignar un nombre a un subcampo y a inicializarlo, debe asignar el nombre al subcampo antes de inicializarlo. Si la especificación de inicialización precede la especificación de asignación de nombre, el compilador considera el campo como no definido y se produce un error.
- Para subcampos descritos en el programa, si aparece más de una especificación de inicialización para un subcampo, se tratan las especificaciones como definiciones duplicadas del campo.

Nota: Como la inicialización en tiempo de compilación forma parte del paso de inicialización del programa, si el programa finaliza con LR desactivado, los subcampos no se inicializarán automáticamente durante la siguiente llamada al programa. Primero debe desactivarse el programa utilizando la operación **FREE**.

10.2.6 Inicialización y el ciclo del programa

La Figura 118 muestra el orden de inicialización en un programa en RPG/400. El valor inicial para un campo es cualquier valor que tenga el campo después de ejecutar *INZSR.

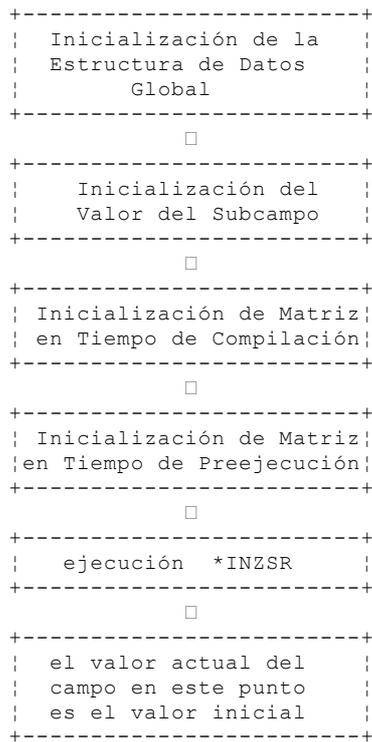


Figura 118. Orden de inicialización en un programa en RPG/400.

10.2.7 Ejemplos de inicialización

Las figuras Figura 119 a Figura 123 muestran algunas inicializaciones típicas de estructuras de datos.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I*
I* La I en la columna 18 inicializa globalmente la estructura de
I* datos. Los subcampos numéricos se inicializan a 0. Los subcampos
I* de caracteres se inicializan en blancos.
I*
IDsnombre..NODSSal-arch+.....NvecLon+.....*
IDS1          IDS
I.I.....Valorinici+++++++PDesdeA++DCampo+.....*
I              1   52DS1S1
I              6   10 DS1S2
I              11  15 DS1S3
I              12  162DS1S4
I*
```

Figura 119. Estructura de datos inicializada globalmente

```
*..1....+....2....+....3....+....4....+....5....+....6....+....7...*
I*
I* En el ejemplo siguiente, se especifica la inicialización global
I* de estructuras de datos para DS1, de modo que el campo AMOUNT
I* se inicializará a cero. AMNTCH se ha inicializado a '1'
I* utilizando la inicialización de valor de subcampo, pero debido
I* a que AMOUNT se declara con posterioridad en la estructura de
I* datos y solapa a AMNTCH, los dos campos contendrán ceros.
I* Si desea que AMNTCH se inicialice a '1', colóquelo después de
I* AMOUNT en la estructura de datos.
I*
I.....Constconnombre+++++++C.....Nomcam.....
I*
IDS1          IDS
I I          '1'              1   6 AMNTCH
I              1   60AMOUNT
I*
```

Figura 120. Inicialización de estructuras de datos a 0 ó 1

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I*
I* La estructura de datos siguiente se inicializa por subcampo. Cada
I* subcampo sólo se inicializa si se especifica una I en la columna
I* 8 de la especificación del subcampo. Observe que el subcampo
I* DS2S2 no se inicializará explícitamente a ningún valor. El
I* subcampo DS2S4 se inicializa a un valor literal largo continuado
I* en varias líneas. Los subcampos DS2S5 y DS2S6 se inicializan a
I* constantes con nombre y campos numéricos respectivamente. El
I* subcampo DS2S7 se inicializa a un valor literal transparente.
I*
I.....campoexter.....PDesdA+++DCampo+.....*
I              -1234567890.234-      C          NUM2
I              56
I              'CHAR-CONST'      C          ALPH1
I*
IDsnombre..NODSSal-arch+.....NvecLon+.....*
IDS2          DS
I.I.....Valorinici+++++++PDesdeA++DCampo+.....*
I I          123              1   30DS2S1
I              4   5 DS2S2
I I          '5CHAR'          6   10 DS2S3
I I          'ESTO ES UN VALOR- 11  70 DS2S4
I              'DE INICIALIZACIÓN-
I              'EN 3 LÍNEAS'
I I          ALPH1              71  80 DS2S5
I I          NUM2              81  915DS2S6
I I          'oAABBCCDDEEi-    92  118 DS2S7
I              'oFFGGHHi'
I*
```

Figura 121. Estructura de datos inicializada por subcampo

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I*
I* La estructura de datos DS3 es una estructura de datos descrita
I* externamente inicializada globalmente. Observe que se han
I* especificado los valores de inicialización del subcampo para los
I* subcampos que se muestran. Los subcampos que no se muestran,
I* DS3S2 y DS3S5, no se inicializan a valores específicos, sino que
I* se inicializarán a blancos o 0. A LONGEXTNM se le asigna el
I* nombre DS3S6 utilizando una especificación de asignación de
I* nombres y se inicializa al valor de constante con nombre NUM1.
I*
I.....campoexter.....PDesdA+++DCampo+.....*
I          123                C          NUM1
I          'CHAR-CONST'      C          ALPH1
I*
IDsnombre..NODSSal-arch+.....NvecLon+.....*
IDS3          EIDS
I.I.....Valorinici+++++++PDesdeA++DCampo+.....*
I I          123                DS3S1
I I          '5CHAR'          DS3S3
I I          ALPH1            DS3S4
I          LONGEXTNM          DS3S6
I I          NUM1
```

Figura 122. Inicialización de una estructura de datos descrita externamente

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
E*
E* DS4 es una estructura de datos de apariciones múltiples
E* inicializada globalmente que contiene una matriz. Aquí, cada
E* elemento de la matriz ARR1 se inicializará a la serie '10CHAR STR'
E* y todas las apariciones del subcampo DS4S2 se inicializarán a 0.
E*
E...Arch(de)Arch(a)+NombreN/rN/tbLonPDSNommatLonPDSComentarios++++*
E*
E          ARR1          15 10

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
IDsnombre..NODSSal-arch+.....NvecLon+.....*
IDS4          IDS          5
I.I.....Valorinici+++++++PDesdeA++DCampo+.....*
I I          '10CHAR STR'      1 150 ARR1
I          151 1610DS4S2
```

Figura 123. Una estructura de datos de apariciones múltiples inicializada globalmente

10.3 Tipos de datos SAA

El compilador del RPG/400 le permite utilizar los tipos de datos de la base de datos SAA de fecha, hora e indicación de la hora, así como campos de longitud variable y campos que permiten valores nulos.

Subtemas

- 10.3.1 Campos de longitud variable
- 10.3.2 Campos de fecha, hora e indicación de la hora
- 10.3.3 Soporte de tipo de datos gráficos DBCS
- 10.3.4 Soporte de valores nulos
- 10.3.5 Manejo de errores para tipos de datos SAA

10.3.1 Campos de longitud variable

Al especificar el valor *VARCHAR en la palabra clave CVTOPT de los mandatos CRTRPGPGM o CRTRPTPGM, el compilador del RPG/400 definirá internamente los campos de longitud variable de un archivo descrito externamente o de una estructura de datos como un campo de caracteres RPG/400 de longitud fija. Cuando no se especifica *VARCHAR, se ignoran los campos de longitud variable y no se pueden acceder desde programas en RPG/400. Para obtener más información, consulte la palabra clave **CVTOPT** en la página 3.2.

Las condiciones siguientes se aplican cuando se especifica ***VARCHAR** en el mandato CRTRPGPGM o CRTRPTPGM:

- Si se extrae un campo de longitud variable de un archivo descrito externamente o de una estructura de datos descrita externamente, éste se declara en un programa en RPG/400 como un campo de caracteres de longitud fija.
- Para campos de caracteres de un solo byte, la longitud del campo RPG/400 declarado es la longitud del campo de las DDS más 2 bytes.
- Para campos de datos gráficos DBCS, la longitud del campo RPG/400 declarado es dos veces la longitud del campo de las DDS más 2 bytes. Para obtener más información sobre los tipos de datos gráficos DBCS, consulte "Soporte de tipo de datos gráficos DBCS" en el tema 10.3.3. Los dos bytes adicionales del campo RPG/400 contienen un número binario que representa la longitud actual (medida en dobles bytes) del campo de longitud variable. La Figura 124 en el tema 10.3.1 muestra la longitud del campo RPG/400 de campos de longitud variable.

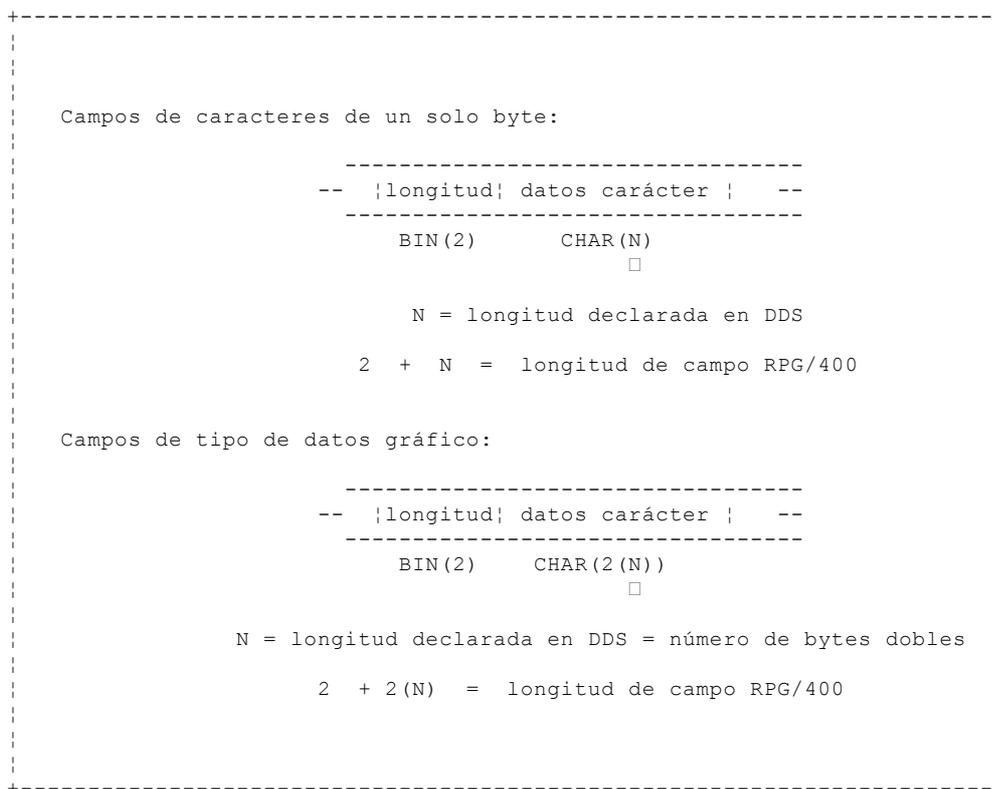


Figura 124. Longitud de campo RPG/400 de campos de longitud variable

- El programa en RPG/400 puede realizar cualquier operación de cálculo de caracteres en el campo declarado de longitud fija. Sin embargo, debido a la estructura del campo, los dos primeros bytes del campo deben contener datos binarios válidos. Se producirá un error de excepción de E/S para una operación de salida si los dos primeros bytes del campo contienen datos de longitud de campo no válida.
- Los indicadores de nivel de control, las entradas de campo coincidentes y los indicadores de campo no están permitidos en una especificación de entrada si el campo de entrada es un campo de longitud variable de un archivo de entrada descrito externamente.
- El proceso secuencial entre límites no está permitido cuando un archivo contiene campos clave de longitud variable.
- Las operaciones con clave no están permitidas cuando el factor 1 de una operación con clave corresponde a un campo clave de longitud variable en un archivo descrito externamente.

- Si elige la salida selectiva de ciertos campos en un registro y no está especificado el campo de longitud variable en la especificación de salida o si el campo de longitud variable se ignora en el programa en RPG/400, el compilador del RPG/400 colocará un valor por omisión en el almacenamiento intermedio de salida del registro recién añadido. El valor por omisión es 0 en los dos primeros bytes y blancos en todos los bytes restantes.
- Si desea cambiar campos de longitud variable, asegúrese de que la longitud de campo actual sea correcta. Una manera de hacerlo es:
 1. Definir una estructura de datos con el nombre de campo de longitud variable como un nombre de subcampo.
 2. Definir un subcampo binario de dos bytes al principio del campo para recuperar la longitud de campo actual.
 3. Actualizar el campo.

Alternativamente, puede trasladar al campo otro campo de longitud variable alineado a la izquierda. Seguidamente se proporciona un ejemplo de cómo cambiar un campo de longitud variable en un programa en RPG/400.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*
A*   El archivo MASTER contiene un campo de longitud variable
A*
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++*****
A*
A           R REC
A           FLDVAR           100           VARLEN

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*
F*   El nombre del archivo descrito externamente es MASTER.
F*   Compile el programa en RPG/400 con CVTOPT(*VARCHAR).
F*
FNombarchIPEAF....LregLClAlDePcLaEDispos+.....KSalidaEntradA....U1.*
F*
FMASTER UF E           DISK

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I*
I*   FLDVAR es un campo de longitud variable definido en las DDS
I*   con una longitud de DDS de 100. Observe que la longitud de campo
I*   RPG es 102.
I*
I.....campoexter.....PDesdeA++DCampo+L1C1FrPoNeCe...*
I*
I           DS
I           1 102 FLDVAR
I           B 1 20FLDLEN
I           3 102 FLDCHR

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C*
C           READ MASTER           LR
C           MOVEL'SALES'           FLDCHR
C           Z-ADD5           FLDLEN
C NLR           UPDATREC
```

Figura 125. Cambio de un campo de longitud variable en un programa en RPG/400

Si se necesitan campos gráficos de longitud variable, puede codificar un campo binario de 2 bytes para guardar la longitud y un subcampo de 2(N) de longitud para guardar la parte de los datos del campo.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I*
I*   El campo gráfico de longitud variable VGRAPH se declara en las
I*   DDS como de longitud 3. Esto significa que la longitud máxima del
I*   campo es de 3 doble bytes o 6 bytes. La longitud total del
I*   campo, contando la parte de la longitud, es de 8 bytes.
I*
I*   Compile el programa en RPG/400 con CVTOPT(*VARCHAR *GRAPHIC).
I*
```

```

I.....campoexter.....PDesdeA++DCampo+L1C1FrPoNeCe...*
I*
IVGRAPH      DS                                8
I              B  1  20VLEN
I              3  8  VDATA

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C*  Sponga que GRPH es un campo gráfico de longitud fija de
C*  2 doble bytes de longitud. Copie GRPH en VGRAPH y
C*  establezca la longitud de VGRAPH en 2.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C*
C              MOVELGRPH      VDATA
C              Z-ADD2         VLEN

```

Figura 126. Utilización de un campo gráfico de longitud variable en un programa en RPG/400

10.3.2 Campos de fecha, hora e indicación de la hora

Los campos de fecha, hora e indicación de la hora se llevan al programa en RPG/400 solamente si se especifica el valor *DATETIME en la palabra clave **CVTOPT** de los mandatos CRTRPGPGM o CRTRPTPGM. Si no se especifica *DATETIME, se ignoran los campos de fecha, hora e indicación de la hora y no se pueden acceder en el programa en RPG/400. Para obtener una descripción y la sintaxis de **CVTOPT**, consulte el parámetro **CVTOPT** en la página 3.2.

Los campos de fecha, hora o indicación de la hora se llevan a un programa en RPG/400 como campos de caracteres de longitud fija. El programa en RPG/400 puede efectuar cualquier operación de caracteres válida en los campos de longitud fija.

Como los tipos de datos de fecha, hora e indicación de la hora tienen cada uno su propio formato, si hay una salida de un campo que contenga información de fecha, hora o indicación de la hora hacia el archivo de la base de datos, el formato y los separadores del campo deben ser exactamente los requeridos por el formato declarado. Si no se utiliza el mismo formato o los mismos separadores, se producirá una condición de excepción/error. Para obtener más información sobre los formatos de estos tipos de datos de la base de datos, consulte *DDS Reference*.

Si añade un registro al archivo de la base de datos y no se especifican los campos de fecha, hora e indicación de la hora de salida, el compilador del RPG/400 colocará los valores por omisión en los campos de salida. El valor por omisión de la fecha para cualquier formato es año 1, mes 1 y día 1. El valor por omisión de la hora para cualquier formato es hora 0, minuto 0 y segundo 0. Se colocarán los separadores adecuados en el campo de acuerdo al formato declarado.

Las operaciones con clave no están permitidas cuando el factor 1 de una operación con clave corresponde a un campo clave ignorado de fecha, hora o indicación de la hora.

No está permitido el proceso secuencial entre límites si hay un campo clave ignorado de fecha hora o indicación de la hora en el archivo.

10.3.3 Soporte de tipo de datos gráficos DBCS

El tipo de datos gráficos DBCS es una serie de caracteres donde cada carácter está representado por 2 bytes. El tipo de datos gráficos no contiene caracteres de desplazamiento a teclado ideográfico (SO) ni a teclado estándar (SI). La diferencia entre datos de un solo byte y datos gráficos DBCS se muestra en la figura siguiente:

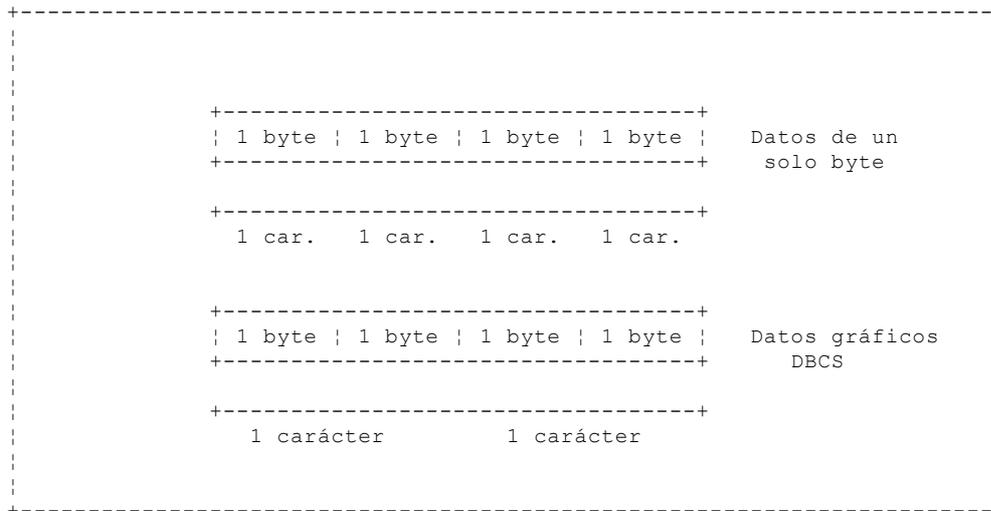


Figura 127. Comparación entre datos de un solo byte y datos gráficos DBCS

Los datos gráficos DBCS solamente se llevan al programa en RPG/400 si se especifica el valor *GRAPHIC en la palabra clave **CVTOPT** de los mandatos CRTRPGPGM o CRTRPTPGM. Si no se especifica *GRAPHIC, se ignoran los datos gráficos y no se pueden acceder en el programa en RPG/400. Para una descripción y la sintaxis de **CVTOPT**, vea el parámetro **CVTOPT** en la página 3.2.

Se aplican las siguientes condiciones cuando se especifica *GRAPHIC en el mandato CRTRPGPGM o CRTRPTPGM:

- Los datos gráficos se llevan a un programa en RPG/400 como campos de caracteres de longitud fija.
- Cada **carácter** de datos gráficos DBCS tiene una longitud de 2 bytes.
- Cada **campo** de datos gráficos de longitud fija tiene una longitud de 2 bytes multiplicado por el número de caracteres del campo. Para obtener una descripción de la longitud de campo de los campos de datos gráficos de longitud variable, consulte "Campos de longitud variable" en el tema 10.3.1.
- El programa en RPG/400 puede efectuar cualquier operación de caracteres válida en los campos de longitud fija.
- Si añade un registro al archivo de base de datos y no se especifica ningún campo gráfico de salida, el compilador del RPG/400 colocará espacios de doble byte en los campos de salida. Si no se especifica ningún campo gráfico variable de salida, el compilador RPG/400 colocará un cero binario seguido por blancos de doble byte en los campos de salida. Las condiciones siguientes propiciarán la colocación de blancos en los campos de salida:
 - No se especifican los campos de salida en la especificación de salida.
 - No se satisfacen los indicadores de condicionamiento del campo.
 - No se especifican los valores requeridos en la palabra clave **CVTOPT**.
- Las operaciones con clave no están permitidas cuando el factor 1 de una operación con clave corresponde a un campo gráfico ignorado.
- No se permite el proceso secuencial entre límites si hay un campo de clave gráfico ignorado en el archivo.

10.3.4 Soporte de valores nulos

Los campos que permiten nulos que contienen valores nulos en un archivo de la base de datos se pueden leer para el programa en RPG/400 si especifica el valor ***YES** en la palabra clave **ALWNULL** de los mandatos CRTRPGPGM o CRTRPTPGM. Actualmente, el soporte de valores nulos sólo se aplica a los archivos de sólo lectura descritos externamente (los archivos sin ninguna adición especificados en la especificación de archivo). Para obtener más información, consulte el parámetro **ALWNULL** en la página 3.2.

Cuando un archivo descrito externamente contiene campos que permiten nulos y se especifica ***NO** en la palabra clave **ALWNULL**, se aplican las siguientes condiciones:

- Un registro que contenga valores nulos recuperado de un archivo de entrada o actualización producirá un error de correlación de datos y se emitirá un mensaje de error.
- No se pueden acceder a los datos del registro y no se puede actualizar ningún campo RPG/400 del registro con los valores del registro de entrada que contiene los valores nulos.
- El compilador del RPG/400 no puede colocar valores nulos en los campos que permiten nulos para actualizar o añadir un registro. Si desea colocar valores nulos en campos que permiten nulos, puede utilizar SQL/400 u otros productos que tengan soporte completo de valores nulos.

Cuando un archivo de sólo entrada descrito externamente contiene campos que permiten nulos y se especifica ***YES** en la palabra clave **ALWNULL**, se aplican las siguientes condiciones:

- Cuando se recupera un registro de un archivo de base de datos y hay algunos campos que contienen valores nulos en el registro, los valores por omisión de la base de datos para los campos que permiten nulos se colocarán en aquellos campos que contengan valores nulos. El valor por omisión será los valores por omisión de las DDS definidas por el usuario o los valores por omisión del sistema.
- Los indicadores de nivel de control, las entradas de campo coincidentes y los indicadores de campo no están permitidos en una especificación de entrada si el campo de entrada es un campo que permite nulos de un archivo de entrada descrito externamente.
- Las operaciones con clave no están permitidas cuando el factor 1 de una operación de cálculo de entrada con clave corresponde a un campo clave que permite nulos de un archivo de sólo entrada descrito externamente.
- El proceso secuencial entre límites no está permitido cuando un archivo contiene campos clave que permiten nulos.
- Los programas creados con archivos de proceso de nulos utilizan una rutina en tiempo de ejecución que sólo puede utilizarse con V2R1M1 o un release posterior.

Nota: Para un archivo descrito por programa, un valor nulo en el registro produce siempre un error de correlación de datos, sin tener en cuenta el valor especificado en la palabra clave **ALWNULL**.

10.3.5 Manejo de errores para tipos de datos SAA

Para una operación de entrada o salida, un error de correlación hará que se emita un mensaje de error grave. Para la salida agrupada por bloques, si uno o más registros del bloque contiene errores de correlación de datos y el archivo se cierra antes de alcanzar el final del bloque, se emite un mensaje de error grave y se crea un vuelco del sistema.

11.0 Capítulo 11. Comunicación con objetos en el sistema

En este capítulo se describe cómo un programa en RPG/400 se comunica con otros programas en el sistema. La función de llamada disponible en un programa en RPG/400 le permite llamar a otros programas o a subrutinas especiales. El programa en RPG/400 proporciona también la función de retorno para permitir el retorno del programa que realiza la llamada, desde un programa llamado.

Subtemas

- 11.1 Llamada a otros programas
- 11.2 Llamada a subrutinas especiales
- 11.3 Retorno desde un programa llamado
- 11.4 Areas de datos

11.1 Llamada a otros programas

El programa en RPG/400 permite la comunicación con otros programas

El código de operación **CALL** (llamar a un programa) y el código de operación **RETRN** (retorno al solicitante) permiten a un programa en RPG/400 llamar a otros programas (por ejemplo, otro programa en RPG/400 o un programa en CL) y retornar al solicitante. Las operaciones **PLIST** (identificar una lista de parámetros) y **PARM** (identificar parámetros) permiten el acceso a los mismos datos tanto al programa de llamada como al programa llamado.

La Figura 128 muestra una visión de concepto de programas en RPG/400 llamando otros programas (en RPG/400 y CL) y programas en CL que llaman a otros programas (en RPG/400 y CL).

IMAGEN 25

Figura 128. Llamada a programas en RPG/400 y en CL

Consulte la Figura 129 para obtener un ejemplo de codificación de un programa en RPG/400 que llama a otro programa en RPG/400 utilizando la función **CALL/RETRN**. Consulte la publicación *CL Guía del Programador* para obtener información sobre la transferencia de parámetros entre un programa en RPG/400 y un programa en CL.

La función **CALL/RETRN** proporciona las siguientes posibilidades:

- Pueden especificarse los códigos de operación **PLIST** y **PARM** con la operación **CALL** para permitir que un programa de llamada y un programa llamado tengan acceso a los mismos datos.

Quando se llama por primera vez un programa en RPG/400, se localiza el programa, se preparan los campos y se pasa el control al programa. En cada llamada posterior, si no ha terminado el programa llamado, todos los campos, indicadores y archivos en el programa llamado son los mismos que había cuando el programa devolvió el control en la llamada precedente. En cada llamada posterior, si el programa llamado ha terminado o si se ha especificado **FREE**, queda disponible una nueva copia del programa.

- El código de operación **FREE** puede especificarse para eliminar un programa llamado de la lista de programas activos. Si se llama de nuevo al programa, se comporta como si se le estuviese llamando por primera vez. Sin embargo, cualquier archivo que esté abierto o cualquier área de datos que esté bloqueada por el programa llamado no son afectados por la operación **FREE**; los archivos o las áreas de datos están todavía asignados al programa llamado.
- La operación **CALL** puede ser dinámica; es decir, el nombre del programa que hay que llamar puede suministrarse en tiempo de ejecución.
- Se suministra un retorno explícito a través del código de operación **RETRN**.
- Se suministra un retorno implícito si se activan los indicadores **LR**, **RT**, o **H1** a **H9** o si, cuando se produce una condición de excepción/error, recibe el control la rutina de manejo de excepción/error del RPG/400.

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7...*
C*
C* Este ejemplo muestra un programa en RPG (PRIN) que utiliza
C* la función CALL/RETRN para llamar a otro programa en RPG (TRADU).
C* La operación EXFMT del programa PRIN graba el registro REPRES en
C* la pantalla. ATRADU y DETRAD son campos del registro. El usuario
C* de la estación de trabajo puede entrar datos con el teclado en el
C* campo ATRADU. El programa TRADU traduce la información que
C* contiene el campo ATRADU.
C*
C* El retorno desde el programa TRADU es a la sentencia
C* inmediatamente posterior a la última sentencia PARM del programa
C* PRIN. El programa PRIN completa la transacción. Cuando se ejecuta
C* la operación GOTO, el programa bifurca hacia atrás al principio
C* de los cálculos. Este bucle continúa hasta que el usuario de la
C* estación de trabajo pulsa una tecla de atención de mandato que
C* activa el indicador 98 para finalizar el programa. (En las DDS
C* de formato del registro REPRES una tecla de atención de mandato
C* está asociada al indicador 98.)
C*
```

C* Cuando el indicador 98 está activado, el programa bifurca a la
C* sentencia FINPRO TAG y la operación FREE libera el programa TRADU.
C* El programa PRIN finaliza cuando se activa el indicador LR.
C*

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* PROGRAMA PRINCIPAL
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C          COMIEN      TAG
C          EXFMTRPRESE
C  98      GOTO FINPRO
C          "
C          "
C          CALL 'TRADU'
C          PARM          ATRADU
C          PARM          DETRAD
C          "              +-----+
C          "              |  Cálculos  |
C          "              +-----+
C          "
C          GOTO COMIEN
C          FINPRO      TAG
C          FREE 'TRADU'
C          SETON          LR
```

```
*..1....+....2....+....3....+....4....+....5....+....6....+....7...*
C*
C* Cuando se ejecuta la operación CALL 'TRADU' en el programa PRIN,
C* se utilizan los nombres CPOY y CPOX en el programa TRADU para
C* acceder a los datos en los campos ATRADU y DTRAD de la lista de
C* parámetros especificada en el programa PRIN. Usando estos datos,
C* el programa TRADU convierte el campo ATRADU, al que se le llama
C* CPOY en el programa TRADU, y coloca el resultado de la operación
C* en el campo CPOX. Se ejecuta a continuación la operación RETRN
C* en el programa TRADU. (Al campo convertido se le llama DETRAD
C* en el programa PRIN). Se especifica una operación RETRN sin
C* el indicador LR activado para mantener intacto el programa y
C* todas sus áreas de trabajo.
C*
C* PROGRAMA DE TRADUCCION
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C          *ENTRY      PLIST
C          PARM          CPOY
C          PARM          CPOX
C          INICIO      TAG
C          "              +-----+
C          "              |  Cálculos  |
C          "              +-----+
C          "
C          RETRN
```

Figura 129. Función CALL/RETRN

Subtemas

- 11.1.1 CALL (Llamada a un programa)
- 11.1.2 PLIST (Identificar una lista de parámetros) y PARM (Identificar parámetros)
- 11.1.3 Soporte gráfico del OS/400
- 11.1.4 FREE (Desactivar un programa)

11.1.1 CALL (Llamada a un programa)

La operación **CALL** transfiere el control del programa de llamada al programa que se llama. Un nombre **PLIST** es optativo en el campo resultado. Si se especifica, éste nombre una lista de datos que pueden comunicarse entre el programa de llamada y el programa al que se llama. Si el programa al que se llama accede a datos del programa de llamada y el campo del resultado está en blanco, la operación **CALL** debe ir inmediatamente seguida por operaciones **PARM**.

Cuando se procesa la operación **CALL**, el programa de llamada transfiere el control al programa llamado. Después de ejecutar el programa llamado, el control se devuelve a la primera sentencia ejecutable después de la operación **CALL** del programa de llamada. Si se produce un error durante el proceso de la operación **CALL** (por ejemplo, no se encuentra el programa llamado), recibe el control la rutina de manejo de la condición de excepción/error del RPG/400. Consulte "Manejo de excepciones/errores" en el tema 4.8 para obtener información detallada sobre el manejo de la condición de excepción/error.

Puede consultar los nombres de los programas llamados mediante una constante con nombre o un literal en un programa en RPG/400 utilizando el mandato de **CLDSPPGMREF**. Si llama a un programa mediante una variable utilizando el código de operación **CALL**, verá una entrada de programa con el nombre de programa *VARIABLE (y sin nombre de biblioteca) que le indica que en el programa hay una llamada por nombre de variable.

Al especificar **CALL** recuerde lo siguiente::

- Un programa puede contener múltiples **CALL** al mismo programa con la misma o distinta **PLIST** especificada.
- La primera **CALL** a un programa provoca la inicialización del programa. En las siguientes **CALL** al mismo programa, se elude la inicialización del programa a menos que se haya especificado la operación **FREE** o de que se haya terminado el programa en una **CALL** anterior.
- El direccionamiento de parámetros se limita a los formatos de datos comunes al programa de llamada y al llamado.
- Cuando un programa de llamada llama a un programa que no está en RPG/400 se activa el indicador de las posiciones 56 y 57 cuando el programa llamado finaliza en error.
- Un programa en RPG/400 no puede llamarse a sí mismo ni a otro programa superior en la pila de invocación. Por ejemplo, si el programa A llama al programa B, el programa B no puede llamar ni al programa A ni al B. Si el programa B retorna, con independencia de si **LR** está activado, y si el programa A llama entonces al programa C, el programa C puede llamar al programa B pero no al programa A ni al C.
- Existen algunas restricciones que se aplican al utilizar el código de operación **CALL**. Para obtener una descripción detallada de estas descripciones, consulte la publicación *RPG/400 Reference*.

11.1.2 *PLIST (Identificar una lista de parámetros) y PARM (Identificar parámetros)*

Las operaciones **PLIST** y **PARM** son operaciones de cálculo que se pueden utilizar con **CALL**. La operación **PLIST**:

- Define un nombre por el cual la lista de parámetros siguiente (**PARM**) puede especificarse en una operación **CALL**
- Define la lista de parámetros de entrada (***ENTRY PLIST**) en un programa llamado.

El factor 1 de la sentencia **PLIST** debe contener el nombre **PLIST**. Este nombre puede especificarse en el campo del resultado de una o más operaciones **CALL**. Si la lista de parámetros es la lista de parámetros de entrada de un programa llamado, el factor 1 debe contener ***ENTRY**. En un programa sólo puede especificarse una ***ENTRY PLIST**.

Puede utilizarse el campo ***PARMS** en la estructura de datos de estado del programa (**PSDS**) para determinar el número de parámetros transferidos a un programa desde un programa de llamada. Utilizando este campo, pueden evitarse las referencias a los parámetros no transferidos desde el programa de llamada y el programa llamado puede dar soporte a parámetros adicionales sin forzar la recopilación o los cambios en el programa de llamada.

Los parámetros que forman parte de la **PLIST** están definidos por las operaciones **PARM** inmediatamente posteriores. El campo del resultado de una sentencia **PARM** identifica los datos que puede direccionar el programa llamado. La conexión entre el programa de llamada y el programa llamado es mediante el direccionamiento; por ello, los parámetros son independientes de los nombres.

Subtemas

11.1.2.1 Reglas para la especificación de PLIST

11.1.2.2 Reglas para la especificación de PARM

11.1.2.1 Reglas para la especificación de *PLIST*

Recuerde lo siguiente al especificar una sentencia **PLIST**:

- Si se especifica **PLIST**, inmediatamente detrás de esta deben ir las **PARM** que le sean aplicables. Si una sentencia **PLIST** no tiene ninguna sentencia **PARM** detrás, la sentencia **PLIST** no es válida.
- En un programa pueden aparecer múltiples sentencias **PLIST**.
- En un programa sólo puede haber una ***ENTRY PLIST**.
- Una **PLIST** y sus correspondientes **PARM** pueden aparecer en cualquier lugar de los cálculos.

11.1.2.2 Reglas para la especificación de PARM

Recuerde lo siguiente al especificar una sentencia a **PARM**:

- Una o más sentencias **PARM** deben ir inmediatamente detrás de una sentencia **PLIST**.
 - Una o más sentencias **PARM** pueden ir inmediatamente detrás de una operación **CALL**.
 - Si en el programa de llamada hay mas parámetros que en el programa llamado, el programa no se ejecutará.
 - Si en el programa llamado hay más parámetros que en el programa de llamada, se producirá un error al utilizar un parámetro no resuelto.
 - Los campos especificados como parámetros en una ***ENTRY PLIST** pueden utilizarse en tiempo de primera página (**1P**).
 - Cuando se especifica una estructura de datos de apariciones múltiples en el campo del resultado de una sentencia **PARM**, todas las apariciones de la estructura de datos se transfieren como un campo único.
 - El campo del resultado de una sentencia **PARM** no puede contener:
 - ***IN**
 - ***INxx**
 - ***IN,xx**
 - Un nombre de área de datos
 - Un nombre de estructura de datos de área de datos
 - Una etiqueta
 - Un literal
 - Un campo de consulta anticipada
 - Una constante con nombre
 - Un nombre de tabla
 - Una palabra reservada para la fecha del usuario.
- Además, no se permite en el campo del resultado de las **PARM** especificadas para una ***ENTRY PLIST**, un elemento de una matriz, ni un nombre de subcampo de estructura de datos, ni el nombre de una matriz en tiempo de compilación ni el nombre de un estado del programa o de la estructura de datos de información de archivo (INFDS), ni una estructura de datos especificada en una ***NAMVAR DEFN** El nombre de un campo puede especificarse sólo una vez en una ***ENTRY PLIST**.
- Cuando se transfieren parámetros a un programa en RPG/400 que se ha llamado a través del CL, los parámetros pueden especificarse en el mandato que llama al programa.
 - El factor 1 de una sentencia **PARM** no puede contener un literal, un campo de consulta anticipada, una constante con nombre ni una palabra reservada para la fecha del usuario.
 - El factor 1 y el factor 2 deben estar en blanco si el campo del resultado contiene el nombre de una estructura de datos de apariciones múltiples.

11.1.3 Soporte gráfico del OS/400

El programa en RPG/400 le permite utilizar la operación **CALL** para llamar gráficos del OS/400, entre los que se incluyen el Gestor de Representación Gráfica de Datos (GDDM) -un conjunto de gráficos elementales para dibujar figuras- y las Rutinas de Gráficos de Presentación -un conjunto de rutinas para diagrama de gestión-. El factor 2 debe contener el literal o constante con nombre '**GDDM**' (no un nombre de campo o elemento de matriz).

Utilice las operaciones **PLIST** y **PARM** para transferir los parámetros siguientes:

- El nombre de la rutina de gráficos que desea ejecutar.
- Los parámetros apropiados para la rutina de gráficos especificada. Estos parámetros deben ser del tipo de datos que requiera la rutina de gráficos.

El programa en RPG/400 no efectúa ninguna inicialización ni terminación implícita para las rutinas de gráficos del OS/400.

Para obtener más información sobre Gráficos del OS/400 y rutinas y parámetros de los gráficos consulte las publicaciones *GDDM Programming Guide* y *Programación del gestor de pantallas de datos gráficos*.

11.1.4 FREE (Desactivar un programa)

El código de operación **FREE**:

- Elimina un programa de la lista de programas activados
- Libera almacenamiento estático si el usuario ya no necesita el programa
- Asegura la inicialización del programa (primer ciclo del proceso) cuando se llama a un programa.

FREE no cierra archivos ni desbloquea áreas de datos. El usuario es responsable de cerrar los archivos y desbloquear las áreas de datos de su propio programa. En un entorno interactivo, puede cerrar archivos y desbloquear áreas de datos utilizando los mandatos de CL RETURN (desde el nivel 1 de la pantalla de entrada de mandatos) o SIGNOFF. (Consulte la publicación *CL Reference* para obtener más información sobre cómo utilizar los mandatos RETURN y SIGNOFF).

Cuando se especifica la operación **FREE**, el programa nombrado en el factor 2 se libera de la lista de programas activados. Si se vuelve a llamar al programa mediante la operación **CALL**, éste funciona como si se llamara por primera vez (primer ciclo de proceso). Si la operación **FREE** no es satisfactoria, la rutina de manejo de la condición de exception/error del RPG/400 recibe el control. Consulte "Manejo de excepciones/errores" en el tema 4.8 para obtener información detallada sobre la rutina de manejo de la condición de excepción/error.

+----- Interfaz de programación de uso general -----+

11.2 Llamada a subrutinas especiales

Las tres subrutinas especiales que están disponibles en un programa en RPG/400 son:

- Subrutina de recuperación de mensajes (**SUBR23R3**)
- Traslado de datos de doble byte entre corchetes y supresión de caracteres de control (**SUBR40R3**)
- Traslado de datos de doble byte entre corchetes y adición de caracteres de control (**SUBR41R3**).

Nota: Para obtener información detallada sobre la utilización de los códigos de operación **CALL** y **PARM**, consulte la publicación *RPG/400 Reference*.

Subtemas

- 11.2.1 Subrutina de recuperación de mensajes (SUBR23R3)
- 11.2.2 Soporte de la interfaz común de programación SAA
- 11.2.3 Traslado de datos de doble byte entre corchetes y supresión de caracteres de control (SUBR40R3)
- 11.2.4 Traslado de datos de doble byte entre corchetes y adición de caracteres de control (SUBR41R3)

11.2.1 Subrutina de recuperación de mensajes (SUBR23R3)

La subrutina de recuperación de mensajes (**SUBR23R3**) le permite recuperar mensajes de un miembro de mensajes de usuario **QUSERMSG**. Si desea utilizar otros archivos de mensaje, puede utilizar el mandato de **CL OVRMSGF** para alterar temporalmente el archivo de mensajes. Después de recuperar el mensaje, éste puede modificarse y grabarse en un archivo de salida.

El enlace con **SUBR23R3** se realiza mediante el código de operación **CALL** y se transfieren los parámetros de entrada a **SUBR23R3** mediante los códigos de operación **PARM**. Especifique **CALL** en las columnas 28 a 32 y '**SUBR23R3**' en las columnas 33 a 42 para utilizar **SUBR23R3**. Deben especificarse cinco códigos de operación **PARM** después de la operación **CALL** con las entradas de campo de resultado siguientes:

Campo del Resultado	Descripción
Identidad del Mensaje (MSGID)	<p>Si NIVEL = 1 ó 2, nombre de un campo numérico de 4 dígitos que tendrá como prefijo 'USR' para formar la identidad del mensaje que va a recuperarse.</p> <p>o</p> <p>Si NIVEL = 3 ó 4, nombre de un campo de carácter de 7 posiciones que contiene el identificador del mensaje que va a recuperarse. El formato de este campo es <i>aaannnn</i>, en el que <i>a</i> es cualquier valor de la A a la Z o los caracteres Ñ,@, o \$ y <i>n</i> es cualquier valor de del 0 al 9 o de la A a la F.</p>
Area de Texto	Nombre del campo alfanumérico o estructura de datos en la que se lee el texto del mensaje. La longitud máxima de un mensaje de nivel 1 es de 132 caracteres y la de un mensaje de nivel 2 es de 3000 caracteres. (Deben utilizarse estructuras de datos cuando el mensaje tenga más de 256 caracteres).
Nivel	Nombre de un campo numérico de un dígito que designa el nivel del miembro del mensaje del usuario. Un valor 1 ó 3 en este campo indica un mensaje de nivel 1; un valor 2 ó 4 indica un mensaje de nivel 2. (Deben utilizarse estructuras de datos cuando el mensaje tenga más de 256 caracteres). El valor 1 ó 2 indica que el campo MSGID es un campo numérico de 4 dígitos y el valor 3 ó 4 indica que MSGID es un campo alfanumérico de 7 dígitos.
Código de Retorno	<p>Nombre de un campo numérico de un dígito que contiene los códigos de retorno. Los códigos de retorno y su significado son los siguientes:</p> <p><i>Código de Retorno Significado</i></p> <p>0 El mensaje se ha recuperado satisfactoriamente, sin truncamiento El mensaje puede contener texto o no.</p> <p>1 El mensaje se ha recuperado satisfactoriamente; pero se ha truncado debido a que la longitud del área del texto era inferior a la longitud del mensaje.</p> <p>2 No se ha encontrado el mensaje.</p> <p>3 El nivel del mensaje no es válido (No es 1,2,3 ó 4)</p> <p>4 Se diagnosticó un valor incorrecto MSGID. (El valor no estaba entre 0000 y FFFF.)</p> <p>5 No se ha encontrado el archivo de mensajes, el usuario no tiene la autorización correcta o la longitud del texto del mensaje supera la longitud máxima del nivel 1.</p> <p>6 Se ha diagnosticado un valor TXTL no válido.</p>
Longitud del Texto	Nombre de un campo numérico de 4 dígitos que contiene la longitud de área de texto definida en el programa de

```
| (TXTL)          | llamada.          |
+-----+-----+

```

El área de texto, especificada por la segunda operación **PARM** se borra a blancos antes de recuperar un mensaje; por lo tanto, en algunas condiciones, se retorna al programa del usuario un área de texto en blanco cuando el valor del código de retorno sea 2 o mayor. Un total de 132 posiciones (para los mensajes de nivel 1) del área de texto se borran a blancos a menos que ésta tenga menos de 132 caracteres de longitud (3000 para los mensajes de nivel 2).

Nota: Debe asegurarse que el área de texto que especifique no exceda el área de texto proporcionada. Si el área de texto excede el área proporcionada, pueden producirse resultados no esperados dado que los datos de su programa pueden ser grabados encima cuando se recupera el mensaje.

11.2.2 Soporte de la interfaz común de programación SAA

El archivo fuente **QIRGINC** en las bibliotecas **QRPG** y **QRPGP** contiene miembros que dan soporte a las inclusiones para múltiples Interfaces comunes de programación SAA. Estas inclusiones describen las interfaces de argumento o parámetros. Los archivos son propiedad de IBM y no deben cambiarse. Los archivos son propiedad de IBM y no deben cambiarse. Si desea adaptar una o más de las inclusiones, copie el o los miembros que desee cambiar a un archivo fuente en una de sus bibliotecas.

Nota: Dado que las bibliotecas del producto **QRPG** y **QRPGP** se añaden a la lista de bibliotecas del producto al compilar, la biblioteca que contiene sus inclusiones adaptadas debe definirse explícitamente. De lo contrario, se utilizarán las inclusiones proporcionadas por IBM.

Si copia las inclusiones a su biblioteca, debe renovar estas copias cuando se instale una nueva release o cuando se realicen cambios mediante PTF. IBM sólo proporcionará mantenimiento a las inclusiones que residen en las bibliotecas **QRPG** y **QRPGP**.

11.2.3 Traslado de datos de doble byte entre corchetes y supresión de caracteres de control (SUBR40R3)

La rutina de traslado y edición **SUBR40R3** mueve contenido de un campo a otro campo. Si los caracteres de control S/O y S/I se encuentran como primero y último carácter en el campo, la rutina **SUBR40R3** los borra. La rutina **SUBR40R3** se invoca tal como se muestra en la Figura 130.

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7...*
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          CALL 'SUBR40R3'
C          PARM          EMPNO  10          CAMPO EMISOR
C          PARM          SOCSEC  8          CAMPO RECEPTOR
C          PARM          RETCDE  1          CODIG. RETORNO
C          PARM          RECLEN  30         LONGITUD REC.
C*
```

Figura 130. Llamada a SUBR40R3

Si desea que el campo receptor contenga todos los datos que se hallaban presentes en el campo emisor, tiene que especificar una longitud para el campo receptor que tenga dos posiciones menos que la longitud del campo emisor. Ello permite dos posiciones para cada carácter de doble byte (o una para cada carácter EBCDIC) al mismo tiempo que se borran los caracteres de control S/O y S/I (y las dos posiciones que ocupaban). Si especifica un campo receptor más largo que el campo emisor menos dos posiciones, todos los datos del campo emisor se trasladan y el campo receptor queda relleno con blancos por la izquierda (blancos de 1 byte EBCDIC). Si el campo receptor es más corto que el campo emisor menos dos posiciones, los datos que se trasladan quedan truncados por la izquierda.

Deben especificarse cinco campos **PARM** cuando se llama a **SUBR40R3**. Los dos primeros especifican los campos de envío y recepción para el traslado. El tercer campo es donde se graban los códigos de retorno para indicar el estado de la operación de traslado. Los campos cuarto y quinto deben cargarse con las longitudes de los campos de envío y recepción. Estas son las longitudes de los campos especificados en los dos primeros **PARM** para la llamada a **SUBR40R3** (en Figura 130, necesitaría cargar las longitudes de **EMPNO** y **SOCSEC**). El campo de código de retorno debe estar definido como un campo alfanumérico de una posición; los campos de longitud deben estar definidos como campos numéricos de tres posiciones con cero posiciones decimales.

Para información sobre datos gráficos DBCS (los datos DBCS que no utilizan caracteres de control S/O (desplazamiento a teclado ideográfico) ni caracteres de control S/I (desplazamiento a teclado estándar)), consulte "Soporte de tipo de datos gráficos DBCS" en el tema 10.3.3.

La rutina **SUBR40R3** genera códigos de retorno para indicar el estado de la operación de mover. La lista siguiente contiene dichos códigos de retorno y sus significados:

Código de Retorno	Explicación
0	Datos trasladados; no hay errores
1	Datos trasladados; se ha producido relleno
2	Datos trasladados; se ha producido truncamiento
3	Datos trasladados; no se han encontrado los caracteres de control S/O y S/I.
4	Datos no trasladados. Se ha encontrado un campo de longitud impar, un campo de longitud cero, una longitud mayor que 256 o un carácter no válido en la longitud del campo. La longitud especificada en los parámetros cuarto y quinto es mayor que la longitud del campo de los parámetros primero y segundo respectivamente.

Si puede emitirse más de un código de retorno, sólo se devuelve el código de retorno más alto.

11.2.4 Traslado de datos de doble byte entre corchetes y adición de caracteres de control (SUBR41R3)

La rutina de traslado y edición **SUBR41R3** mueve contenido de un campo a otro campo. Si los caracteres de control S/O y S/I no se encuentran en la primera y en la última posición del campo, **SUBR41R3** los añade al campo cuando se mueve.

La rutina **SUBR41R3** se invoca tal como se muestra en la Figura 131.

```
*...1....+....2....+....3....+....4....+....5....+....6....+....7...*
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C          CALL 'SUBR41R3'
C          PARM          SOCSEC  8
C          PARM          EMPNO  10
C          PARM          RETCDE  1
C          PARM          SNDLEN  30
C          PARM          RECLEN  30
C*
```

Figura 131. Llamada a SUBR41R3

Si desea que el campo receptor contenga todos los datos que hay en el campo emisor, ha de especificar una longitud para el campo receptor con dos posiciones más que la longitud del campo emisor (para contener los caracteres de control S/O y S/I). Si especifica un campo receptor más largo que el campo emisor más dos, los datos se rellenan por la derecha cuando se trasladan al campo receptor. Si el campo receptor es más corto que el campo emisor más dos, los datos se truncan por la izquierda cuando se trasladan. Si el campo receptor se especifica más largo o más corto que el campo emisor más dos posiciones, el carácter de control S/I queda todavía colocado en su posición correcta (la que está más a la derecha).

Deben especificarse cinco campos **PARM** cuando se llama a **SUBR41R3**. Los dos primeros especifican los campos de envío y recepción para el traslado. El tercer campo es donde se graban los códigos de retorno para indicar el estado de la operación de traslado. Los campos cuarto y quinto deben cargarse con las longitudes de los campos de envío y recepción. Estas son las longitudes de los campos especificados en los dos primeros **PARM** para la llamada a **SUBR41R3** (en Figura 131, necesitaría cargar las longitudes de **EMPNO** y **SOCSEC**). El campo de código de retorno debe estar definido como un campo alfanumérico de una posición; los campos de longitud deben estar definidos como campos numéricos de tres posiciones con cero posiciones decimales.

Para información sobre datos gráficos DBCS (los datos DBCS que no utilizan caracteres de control S/O (desplazamiento a teclado ideográfico) ni caracteres de control S/I (desplazamiento a teclado estándar)), consulte "Soporte de tipo de datos gráficos DBCS" en el tema 10.3.3.

La **SUBR41R3** produce códigos de retorno para indicar el estado del traslado. La lista siguiente contiene dichos códigos de retorno y sus significados:

Código de Retorno	Explicación
0	Datos trasladados; no hay errores
1	Datos trasladados; se ha producido relleno por la izquierda del carácter de control S/I.
2	Datos trasladados; se ha producido truncamiento por la izquierda del carácter de control S/I.
3	Datos trasladados; S/O y S/I ya están presentes.
4	Datos no trasladados. Se ha encontrado una longitud de campo impar, un campo de longitud cero, una longitud mayor que 256 o un carácter no válido en la longitud del campo. La longitud especificada en los parámetros cuarto y quinto es mayor que la longitud del campo de los parámetros primero y segundo respectivamente.

Si puede emitirse más de un código de retorno, sólo se emite el código de retorno más alto.

+----- Fin de Interfaz de programación de uso general -----+

11.3 Retorno desde un programa llamado

Un programa en RPG/400 al que se ha llamado devuelve el control al programa de llamada de una de las siguientes formas:

- Con terminación normal
- Con terminación anormal
- Sin terminación.

Sigue a continuación una descripción de las maneras de retornar desde un programa llamado.

Para obtener una descripción detallada de dónde se prueban los indicadores **LR**, de **H1** a **H9** y **RT** y la operación **RETRN** en el ciclo del programa en RPG/400, consulte la sección sobre el ciclo del programa en RPG/400 en la publicación *RPG/400 Reference*.

Subtemas

- 11.3.1 Una terminación normal
- 11.3.2 Una terminación anormal
- 11.3.3 Retorno sin una terminación

11.3.1 Una terminación normal

Un programa normalmente finaliza y devuelve el control al programa de llamada cuando el indicador **LR** está activado y los indicadores **H1** a **H9** no están activados. (Para obtener más información sobre el indicador **LR**, consulte la publicación *RPG/400 Reference*.) El indicador **LR** sólo lo puede activar:

- El último registro procesado de un archivo primario o secundario durante el ciclo del programa en RPG/400.
- El programador.

Un programa también termina normalmente si:

- Se procesa la operación **RETRN**, no están activados los indicadores **H1** a **H9** y el indicador **LR** está activado
- El indicador **RT** está activado, los indicadores **H1** a **H9** no están activados y el indicador **LR** está activado.

Cuando un programa finaliza de forma normal, sucede lo siguiente:

- Los parámetros se mueven desde el factor 2 al campo del resultado.
- Se graban todas las matrices y las tablas que tengan un 'A nombre de archivo' especificado en las especificaciones adicionales y todas las estructuras de datos bloqueadas del área de datos
- Se desbloquean todas las áreas de datos bloqueadas por el programa.
- Se cierran todos los archivos que estén abiertos
- Se establece un código de retorno para indicar al programa de llamada que el programa ha forma normal y se devuelve el control al programa de llamada.

La próxima vez que se llame al programa, habrá disponible una copia nueva para su proceso.

11.3.2 Una terminación anormal

Un programa termina de forma anormal y devuelve el control al programa de llamada cuando sucede una de las causas siguientes:

- Un indicador **H1** a **H9** está activado y se toma la opción de cancelar cuando se emite un mensaje.
- Se toma la opción de cancelar cuando se emite un mensaje de error del RPG/400.
- Se procesa una sentencia ***CANCL ENDSR** en una subrutina ***PSSR** o **INFSR** (para obtener más información sobre el punto de retorno ***CANCL** para las subrutinas ***PSSR** e **INFSR**, consulte "Manejo de excepciones/errores" en el tema 4.8).
- Un indicador **H1** a **H9** está activado cuando se procesa una operación **RETRN**.
- Un indicador **H1** a **H9** está activado cuando tiene lugar el proceso del último registro (**LR**) en el ciclo del RPG/400.

Cuando un programa finaliza de forma anormal, sucede lo siguiente:

- Se cierran todos los archivos que estén abiertos
- Se desbloquean todas las áreas de datos bloqueadas por el programa.
- Se establece un código de retorno por error en la estructura de datos del programa para indicar al programa que lo ha llamado que el programa ha terminado de forma anormal
- Se emite el mensaje de escape **RPG9001** y se devuelve el control al programa de llamada.

La próxima vez que se llame al programa, habrá disponible una copia nueva para su proceso. (Para obtener más información sobre la estructura de datos de estado del programa, consulte "Manejo de excepciones/errores" en el tema 4.8).

11.3.3 Retorno sin una terminación

Un programa puede devolver el control al programa de llamada sin terminar cuando se procesa la operación **RETRN** o el indicador **RT** está activado y los indicadores **LR** ni **H1** a **H9** no están activados. La operación **RETRN** hace que el control se devuelva inmediatamente al programa de llamada. El indicador **RT** hace que el control se devuelva al programa de llamada después de que se hayan probado los indicadores **H1** a **H9** y el indicador **LR**. (Para obtener más información sobre el indicador **RT**, consulte la publicación *RPG/400 Reference*.)

Un programa también retorna sin terminar cuando algo fuera del programa finaliza su activación. Por ejemplo:

- Un programa A en RPG/400 llama a otro programa (tal como un programa en CL) el cual emite un mensaje de escape directamente al programa de llamada A.
- Un programa en COBOL llama a un programa en RPG/400 el cual llama a otro programa COBOL, que termina utilizando una sentencia **STOP RUN**. **STOP RUN** termina la unidad de ejecución del COBOL, la cual incluye al programa en RPG/400.

Si se llama a un programa y se devuelve sin terminar, cuando se vuelva a llamar al programa, todos los campos, indicadores y archivos del programa contendrán los mismos valores que tenían cuando se dejó el programa, a menos que se llame a otro programa primero.

Puede utilizar el código de operación **RETRN** o el indicador **RT** junto con el indicador **LR** y los indicadores **H1** a **H9**. Debe tener en cuenta la secuencia de prueba en el ciclo del programa en RPG/400 para la operación **RETRN**, el indicador **RT**, el indicador **LR** y los indicadores **H1** a **L9**.

11.4 Áreas de datos

Un área de datos es un objeto utilizado para comunicar datos tales como valores variables entre programas dentro de un trabajo y entre trabajos. Un área de datos puede crearse y declararse a un programa antes de emplearse en ese programa o trabajo. Consulte la publicación *CL Guía del Programador* la información sobre cómo crear y declarar un área de datos. Un programa en RPG/400 no da soporte a áreas de datos definidas por el mandato de CL RTDTAARA en el que se especifica ***LGL** como parámetro TYPE. Además, en la estructura de datos de área de datos no se da soporte a las áreas de datos con el tipo *DEC. Puede utilizarse la sentencia *NAMVAR DEFN para acceder a un área de datos con el tipo *DEC. La biblioteca que contiene el área de datos debe especificarse en la lista de bibliotecas.

El programa en RPG/400 proporciona acceso a un área de datos a través de una estructura de datos de área de datos, de las operaciones de área de datos **IN** y **OUT** o de una combinación de ambas. Para información sobre cómo especificar una estructura de datos del área de datos, consulte "Estructuras de datos" en el tema 9.2.

Para una estructura de datos de área de datos, si el área de datos existe en una biblioteca que esté especificada en la lista de bibliotecas, el área de datos se copia en el programa. Si el área de datos no existe en una biblioteca que esté especificada en la lista de bibliotecas, el nombre y la longitud de la estructura de datos se utiliza para generar un área de datos en la biblioteca temporal del trabajo (QTEMP).

El programa en RPG/400 recupera y bloquea el contenido del área de datos al inicializarse el programa, cuando está definida en el programa una estructura de datos de área de datos. Al final del programa, el programa en RPG/400 graba la estructura de datos de área de datos en el área de datos de la que procede (biblioteca temporal o permanente) y desbloquea la estructura de datos del área de datos. Sin embargo, si una estructura de datos del área de datos se desbloquea en el momento en que el programa en RPG/400 efectúa la actualización, el programa en RPG/400 no la graba al final del programa. Al final del trabajo, se suprime la biblioteca temporal del trabajo QTEMP.

Las operaciones **IN** y **OUT** recuperan y graban en un área de datos. La posibilidad de bloqueos es optativa con estas operaciones.

Si el programa llama a otro programa que utiliza la misma área de datos de llamada, debe desbloquear el área de datos (con la operación **UNLCK**) antes de llamar a otro programa. Dos programas no puede utilizar simultáneamente la misma área de datos para salida.

Un área de datos sólo puede bloquearse una vez. Un programa en RPG/400 no puede recuperar y bloquear un área de datos que ya ha sido bloqueada. Los programas que intentan recuperar y bloquear áreas de datos son :

- Programas que utilizan un área de datos como una estructura de datos del área de datos
- Programas que utilizan una operación **IN** con ***LOCK** especificado en el factor 1.

El programa en curso que está ejecutándose no puede bloquear el área de datos, si:

- El área de datos se ha utilizado en un mandato de CL ALCOBJ en el mismo o en otro paso de direccionamiento.
- El área de datos ha sido bloqueada por un programa que ha llamado al programa en curso.

Para acceder en un área de datos que ha sido bloqueada con autorización para lectura, ha de recuperarla utilizando una operación **IN** con blancos en el factor 1. En este caso, el programa puede recuperar el área de datos, pero no puede alterarla usando la operación **OUT**.

El programa en RPG/400 utiliza los siguientes estados de bloqueo:

- Una operación **IN** que tenga especificado ***LOCK**, tiene un estado de bloqueo de lectura permitida exclusivamente (***EXCLRD**).
- Una operación **IN** que no tenga especificado ***LOCK**, tiene un estado de bloqueo de lectura compartida (***SHRRD**) mientras se transfieren datos. Al terminar la transferencia, el programa en RPG/400 libera el estado de bloqueo.
- Una operación **OUT** tiene un estado de bloqueo exclusivo (***EXCL**) durante la transferencia de datos y este estado se libera a continuación. El programa en RPG/400 libera entonces el estado de bloqueo de lectura permitida exclusivamente (***EXCLRD**) establecido por la operación **IN**.

Otro estado de bloqueo de programa en un área de datos puede interferir con la operación de algunos o de todos los estados de bloqueo del programa en RPG/400. Consulte la exposición sobre la asignación de recursos en la publicación *CL Guía del Programador* para más información sobre cuales son los bloqueos compatibles con otros bloqueos.

La Figura 132 muestra una estructura de datos de área de datos y las operaciones **IN** y **OUT** (dentro de un mismo programa) que tienen acceso a la misma área de datos.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I*
I* En este ejemplo, el RPG lee y graba implícitamente la estructura de
I* datos del área de datos y la actualiza explícitamente. Al
I* inicializarse el programa, el RPG recupera y bloquea
I* implícitamente TOTALS. Después de que se efectúan los cálculos
I* con TOTALS, la sentencia *LOCK OUT actualiza y mantiene el estado
I* de bloqueo de TOTALS. Al final del programa, el RPG graba y
I* desbloquea implícitamente TOTALS
I*
IDsnombre..NODsSal-arch++.....NvecLon+.....*
ITOTALS      UDS
I.....campoexter.....PDesdA+++DCampo+.....*
I
I          1  82TOTIMP
I          9 182TOTBRU
I         19 282TOTNET
*...1....+...2....+...3....+...4....+...5....+...6....+...7...*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          "          +-----+
C          "          |  Cálculos  |
C          "          +-----+
C          ADD  AMOUNT  TOTIMP
C          ADD  GROSS   TOTBRU
C          ADD  NET     TOTNET
C          *LOCK  OUT  TOTALS
C          "          +-----+
C          "          |  Cálculos  |
C          "          +-----+
C          ADD  AMOUNT  TOTIMP
C          ADD  GROSS   TOTBRU
C          ADD  NET     TOTNET
C          "          +-----+
C          "          |  Cálculos  |
C          "          +-----+
C          *NAMVAR  DEFN      TOTALS

```

Figura 132. Operaciones de estructura de datos y de área de datos

Subtemas

11.4.1 Área de datos de parámetros de inicialización del programa (PIP)

Area de datos de parámetros de inicialización del programa (PIP)

11.4.1 Area de datos de parámetros de inicialización del programa (PIP)

Si el programa en RPG/400 es un trabajo prearrancado que recibirá los datos del parámetro de inicialización del programa (PIP), puede utilizarse el Area de datos de los PIP (PDA) para recuperar los datos. Para definir el PDA, utilice el código de operación ***NAMVAR DEFN** y después de adquirir el dispositivo del programa solicitante, emita un código de operación **IN** especificando en el factor 2 el nombre del PDA que ha definido en el código de operación **DEFN**.

A diferencia de otras áreas de datos, no puede efectuar las operaciones **LOCK**, **UNLOCK**, o grabar datos en PDA utilizando el código de operación **OUT**. Para obtener más información sobre cómo definir los PDA consulte la publicación *RPG/400 Reference*.

Para obtener más información sobre trabajos prearrancados, consulte el manual *ICF Programmer's Guide*.

12.0 Capítulo 12. El generador automático de informes

Este capítulo contiene información sobre la función de generación automática de informes del RPG/400. Es un programa que opera con anterioridad al compilador del RPG/400. El generador automático de informes en el sistema AS/400 está pensado para la conversión de programas del generador automático de informes ya existentes. La utilización del generador automático de informes con las mejoras del RPG/400 en el AS/400 tales como los archivos descritos externamente o el código de operación **DEFN** pueden producir resultados no deseados.

Subtemas

- 12.1 Impresión por grupos
- 12.2 Especificaciones de la sentencia /COPY
- 12.3 Formato de generador de informes
- 12.4 Especificaciones generadas
- 12.5 Ayudas de programación
- 12.6 Utilización de CRTRPTPGM para compilar un programa generador automático de informes
- 12.7 Ejemplos de utilización del generador automático de informes

12.1 *Impresión por grupos*

En la impresión por grupos, los datos se resumen para un grupo de registros de entradas y sólo se imprimen los totales en el informe. Los totales pueden tener subtotales junto a un total final o sólo un total final.

Subtemas

12.1.1 Especificaciones

12.1.2 Ejemplos

12.1.1 Especificaciones

Para especificar una impresión por grupos utilizando el generador automático de informes, tiene que entrar una **T** en la posición 15 y ***AUTO** en las posiciones 32 a 36. Puede especificar un indicador de nivel de control en las posiciones 23 a 31. Cuando se utiliza la especificación **T-*AUTO**, no se imprime una línea por cada registro, sino sólo después de haberse leído un grupo de control completo.

Los campos y constantes definidos por las especificaciones de descripción de campos que contengan un blanco o una **B** en la posición 39 y sigan a una descripción de registro **T-*AUTO** se imprimen en la línea de totales de nivel más bajo. Los campos definidos con una **A** en la posición 39 no se imprimen en la línea de totales, pero los campos de totales que cree el generador automático de informes sí. Los cálculos generados se imprimen en sus líneas de totales asociadas. Las cabeceras de columnas continuadas (**C** en la posición 39) y los campos indicados por totales (**1 a 9** o **R** en la posición 39) también pueden especificarse mediante descripciones de campo que sigan a una descripción de registro **T-*AUTO**.

Si la posición 39 de las especificaciones de descripción de campos contiene un blanco o una **B**, pueden entrarse indicadores de salida en las posiciones 23 a 31 de una especificación de descripción de campos que siga a una descripción de registro **T-*AUTO**. Si se utilizan indicadores de salida en una descripción de campos que contenga una **A** en la posición 39 y que siga a una especificación **T-*AUTO**, el generador automático de informes ignora dichos indicadores. Los indicadores de salida no pueden utilizarse en una descripción de campos que contenga **C**, **1 a 9**, o **R** en la posición 39.

12.1.2 Ejemplos

La Figura 133 muestra las especificaciones de descripción de archivos y de entrada necesarias para los informes impresos por grupos que se representan en la Figura 135 y en la Figura 137. Los campos **BRANCH** y **REGION** se definen como campos de control.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF....LregLCIAIDePcLaEDispos+.....KSalidaEntradA....U1.*
FSALES IP F 43 DISK
FPRINT O F 120 PRINTER
FDISKSUM O F 25 DISK
F*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
ISALES AA 01
I.....PDesdeA+++DNomcamL1M1FrP1NeCe...*
I 1 7 ITEMNO +-----+
I 8 9 BRANCHL1 |L1 y L2 son los|
I 10 10 REGIONL2 |niveles de |
I 11 25 DESC |control |
I 26 27SOLDQY |especificados |
I 28 342SOLDVA +-----+
I 35 36ONHAND
I 37 432VALUE
I*
```

Figura 133. Especificaciones de descripción de archivo y entrada para informes por grupo impresos

Este programa también produce un archivo resumen **DISKSUM**. El archivo resumen contiene un registro resumen de los datos de las ventas efectuadas por cada sucursal. Las especificaciones de salida para **DISKSUM** ilustran la utilización de las especificaciones de salida en RPG/400 estándar, junto con especificaciones ***AUTO** en el mismo programa. El registro de salida que se describe se graba en el archivo **DISKSUM**, cuando se produce una ruptura de control **L1** (cambia el campo **BRANCH**). Como la especificación **T-*AUTO** está condicionada por **L2**, el generador automático de informes no produce campos para el nivel de control **L1**. Por lo tanto, tienen que utilizarse especificaciones de cálculo estándar del RPG/400 para calcular los totales por **L1**. Los campos de totales **L1** que se graban en el archivo **DISKSUM** (**SOLDQ1**, **SOLDV1**, y **VALUE1**) tienen que definirse en los cálculos.

La Figura 134 muestra las especificaciones de salida y el informe impreso por grupos reflejando los totales de ventas de una compañía. Dado que la especificación **T-*AUTO** está condicionada por **L2**, sólo se imprimen en el informe los totales por **REGION** (**L2**) y por la compañía en su totalidad (**LR**). Los totales por sucursal **BRANCH** (**L1**) no se imprimen.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C 01 SOLDQ1 ADD SOLDQY SOLDQ1 40
C 01 SOLDV1 ADD SOLDVA SOLDV1 92
C 01 VALUE1 ADD VALUE VALUE 92
C*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OPRINT H *AUTO
O.....N01N02N03Campo+YBUpoSPConstante/palabraedición+++*
O 'VENTAS DE UNA COMPAÑIA'
O ' POR SUCURSAL Y REGION'
O*
O 1 T 2 L2 *AUTO
O BRANCH 'SUCURSAL'
O SOLDQY A 'NUMERO DE VENTAS'
O SOLDVA A 'VALOR'
O VALUE A 'VALOR DEL STOCK'
O C 'EXISTENCIAS'
O R 'REGION'
O REGION 2
O 2 'TOTALES'
O R 'TOTAL DE COMPAÑIA'
O
ODISKSUM T L1
O REGION 1
O BRANCH 3
O SOLDQ1 B 7
O SOLDV1 B 16
O VALUE1 B 25
```

O*

Figura 134. Utilización de *AUTO para generar un informe impreso por grupos mostrando estructura de datos para acumular totales - Ejemplo 2

- 1 La **T** en la posición 15 con ***AUTO** en las posiciones 32 a 37 especifica un informe impreso por grupos.
- 2 Dado que **L2** se entra en indicadores de salida, se imprimen las líneas de totales solamente para **L2** y **LR**, aunque también se ha definido **L1** como nivel de control. En la impresión por grupos, las líneas de totales de nivel más bajo que se imprimen (**L2** en este caso), se imprimen a espaciado sencillo, al igual que las líneas de detalle.

IMAGEN 26

Figura 135. Informe impreso por grupos mostrando totales por región y finales

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OPRINT H *AUTO
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O 'VENTAS DE UNA COMPAÑIA'
O ' POR SUCURSAL Y REGION'
O*
O T *AUTO
O BRANCH 'SUCURSAL'
O SOLDQY A 'NUMERO DE VENTAS'
O SOLDVA A 'VALOR'
O VALUE A 'VALOR DEL STOCK'
O C 'EXISTENCIAS'
O 2 'REGION'
O REGION 2
O 2 'TOTALES'
O R 'TOTAL DE COMPAÑIA'
O*
```

Figura 136. Utilización de *AUTO para generar un informe impreso por grupos que muestra totales por región, estructura de datos para acumular totales - Ejemplo 2

Cuando no se entran indicadores de nivel de control en indicadores de salida, se genera una línea de totales por cada nivel de control que haya definido (**L1** y **L2**, en este caso) y para **LR**.

La Figura 137 muestra un informe impreso por grupos similar al que se muestra en la Figura 135. Sin embargo, las especificaciones **T-*AUTO** no están condicionadas por un indicador de nivel de control, razón por la que se imprimen totales para todos los niveles de control definidos y para **LR**.

IMAGEN 27

Figura 137. Informe impreso por grupos que muestra totales por región, por sucursal y finales

12.2 Especificaciones de la sentencia /COPY

La función de copia del generador automático de informes proporciona un medio para incluir especificaciones fuente de RPG/400 en un programa en RPG/400 desde un miembro de archivo fuente. Utilice la función de copia para incluir especificaciones fuente que sean idénticas o casi idénticas, en varios programas distintos, reduciendo de este modo la necesidad de codificar repetidamente las especificaciones que se utilicen en distintos programas. Por ejemplo, si las especificaciones de descripción de archivos y de entrada para un archivo determinado son similares en varios programas, dichas especificaciones pueden introducirse en un miembro de archivo fuente e incluirse en cualquier programa mediante la función de copia.

Las especificaciones del generador automático de informes y cualquier especificación de RPG/400 válida, incluyendo las matrices y tablas pueden copiarse de esta manera. Cuando las tablas o matrices en tiempo de compilación están contenidas en miembros de /COPY, debe asegurarse de que los datos se encuentran en el mismo orden que las especificaciones de la extensión que declara las matrices o tablas. Las especificaciones de la opción del generador automático de informes y otras sentencias de copia no se pueden copiar. Consulte "Ejemplos de utilización del generador automático de informes" en el tema 12.7 para ver un ejemplo de utilización de la función de copia.

Las especificaciones que incluyen la función de copia en un programa del generador automático de informes se sitúan inicialmente inmediatamente a continuación de la sentencia /COPY. Una vez copiadas todas las especificaciones desde un miembro de archivo fuente, todo el programa generador automático de informes se clasifica se clasifican en el orden solicitado por el compilador del RPG/400.

Para solicitar la función de copia utilice la sentencia /COPY. Esta sentencia identifica el miembro de archivo fuente que contiene las especificaciones del RPG/400 que va a incluirse en el programa fuente generado por el generador automático de informes. Las sentencias /COPY tienen que ir después de las especificaciones de opciones del generador automático de informes y tienen que preceder a las matrices y tablas fuente (tablas de conversión de archivos, tablas de orden de clasificación alternativo y tablas y matrices en tiempo de compilación).

El nombre del archivo especificado en una sentencia /COPY no debe modificarse mediante un mandato de alteración temporal del lenguaje de control. En una sentencia /COPY, no puede especificarse como archivo ningún archivo de datos incorporado.

La especificación /COPY del generador automático de informes es similar en cuanto a sintaxis a la directiva /COPY del compilador.

El formato de la sentencia /COPY es:

Posición	Entrada
1-5	Número de página y línea que indica el lugar de la sentencia en la secuencia de especificaciones fuente del generador automático de informes.
6	Esta posición puede contener ahora cualquier entrada excepto H o U , o puede dejarse en blanco.
7-11	Entrar los caracteres /COPY.
12	Blancos.
13-44	Entrar el nombre calificado del archivo (nombre-biblioteca/nombre-archivo en el sistema AS/400 o nombre-archivo.nombre-biblioteca en el Entorno del Sistema/38), seguido por una coma, seguido por el nombre del miembro. Si el nombre de la biblioteca no está especificado, se utiliza la lista de bibliotecas (*LIBL) para localizar el archivo. Si se ha especificado como nombre de archivo F1 , F2 , R1 , o R2 , se asume el nombre de archivo QRPGSRC y se utiliza la lista de bibliotecas para localizar el archivo. Si sólo aparece una entrada, ésta es el nombre del miembro; se asume el nombre del archivo QRPGSRC y se utiliza la lista de bibliotecas para localizar el archivo. El nombre del miembro tiene que existir en QRPGSRC .
45-49	Blancos.
50-80	Entrar cualquier información o comentario. El generador automático de informes no lee el contenido de estas posiciones.

La Figura 138 muestra un ejemplo de la sentencia /COPY.

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

```
I/COPY QGPL/SALES,SALETR
I*      □          □
I*
I*  nombre      nombre de
I*  calificado  miembro
```

Figura 138. Ejemplo de la sentencia /COPY del generador automático de informes

Nota: Es conveniente codificar la sentencia /COPY en las especificaciones de entrada si estas tienen que modificarse cuando se copian.

Subtemas

12.2.1 Modificación de especificaciones copiadas

12.2.1 *Modificación de especificaciones copiadas*

Las sentencias se pueden incluir en las especificaciones del generador automático de informes para cambiar la descripción del archivo y las especificaciones del campo de entrada al copiarse desde un miembro del archivo fuente. No se puede cambiar ningún otro tipo de especificación. Las sentencias **/COPY** modificadoras del programa fuente que añaden, cambian o suprimen entradas de especificaciones de campos de entrada se identifican mediante una **X** en la posición de impresión 6 de la lista del generador automático de informes.

Subtemas

- 12.2.1.1 Modificación de especificaciones de descripción de archivos
- 12.2.1.2 Modificación de especificaciones de campos de entrada

12.2.1.1 Modificación de especificaciones de descripción de archivos

Para modificar una especificación de descripción de archivos que se copia desde un miembro de archivo fuente, introduzca el nombre del archivo en las posiciones 7 a 14 de una especificación de descripción de archivos. (**F** en la posición 6). A continuación incluya en dicha línea solamente aquellas entradas que han de sustituir a las existentes en la especificación que se copia o aquellas que tengan que incluirse como nuevas entradas. Las entradas en blanco en la sentencia modificadora no afectan a la sentencia copiada.

Por ejemplo, supongamos que tienen que copiarse desde un miembro de un archivo fuente las especificaciones de descripción de archivos correspondientes a un archivo utilizado con frecuencia y denominado SALES. La especificación original contiene una **I** en el tipo de archivo (posición 15), definiendo el archivo **SALES** como un archivo de entrada. (Consulte Figura 139 en el tema 12.2.1.2.) Para actualizar el archivo de ventas, tiene que cambiarse el contenido de la posición 15 a **U** incluyendo una descripción de archivo modificador en el programa fuente del generador automático de informes. La sentencia modificadora debe contener el nombre del archivo, **SALES**, y la nueva entrada de tipo de archivo **U**. Como resultado de la sentencia modificadora, el tipo de archivo en la especificación de descripción de archivo copiada cambia de **I** a **U**.

Para establecer una entrada en blancos, entre un ampersand (**&**) en la primera posición de dicha entrada en la sentencia modificadora y deje el resto de las posiciones en blanco. Por ejemplo, para eliminar un indicador de desbordamiento (posiciones 33 y 34) de la especificación que se muestra en la Figura 139 en el tema 12.2.1.2, añada un ampersand a la sentencia modificadora en la posición 33, tal como se muestra en la Figura 140 en el tema 12.2.1.2, y deje la posición 34 en blanco.

Las sentencias modificadoras de las especificaciones de descripción de archivo no tienen que estar en un orden determinado en el programa fuente del generador automático de informes, con la excepción de que no pueden ir inmediatamente detrás de la sentencia **/COPY** si se están modificando las especificaciones del campo de entrada.

No se permiten modificaciones en las especificaciones de descripción de archivos que acompañen a la descripción de un archivo copiado. Para añadir nuevas especificaciones de continuación, sitúelas después de una sentencia modificadora de descripción de archivos para el archivo. Están permitidas un máximo de cinco especificaciones de continuación después de una especificación de descripción de archivo (total combinado de las especificaciones de continuación originales y añadidas).

12.2.1.2 Modificación de especificaciones de campos de entrada

Sólo pueden cambiarse las especificaciones de campo de entrada (las que describen campos individuales en un registro de entrada). Para cambiar una especificación de campo de entrada copiada, entre el nombre del campo en las posiciones 53 a 58 de una sentencia modificadora de un campo de entrada (**I** en la posición 6). Las sentencias modificadoras para especificaciones de campo de entrada deben ir inmediatamente después de la sentencia **/COPY** en el programa del generador automático de informes que copia estas especificaciones. La primera especificación después de la sentencia **/COPY** que no sea una especificación de campo de entrada se considera el final de las sentencias modificadoras del campo de entrada para la sentencia **/COPY**. (Una sentencia de comentario con una **I** en la posición 6 no se considera el final de las sentencias modificadoras del campo de entrada.)

Los campos que pueden modificarse son:

- Posición 43 (empaquetado/binario)
- Posiciones 44-51 (situación del campo)
- Posición 52 (posiciones decimales)
- Posiciones 59-60 (niveles de control)
- Posiciones 61-62 (campos de comparación)
- Posiciones 63-64 (relación campo-registro)
- Posiciones 65-70 (indicadores de campo).

El método para sustituir, añadir o borrar entradas es similar al que se utiliza para modificar las especificaciones de descripción de archivos. Para sustituir o añadir entradas, tiene que codificarse la nueva entrada en la posición correspondiente de la sentencia modificadora; para poner una entrada a blancos, entre un ampersand (&) en la primera posición de la entrada en la sentencia modificadora. La Figura 139 muestra un ejemplo de modificación de especificaciones de una descripción de archivo copiado.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I*
I* Sentencia /COPY para copiar especificaciones para el archivo SALES
I* desde la biblioteca QGPL. El nombre del miembro es SALETR.
I/COPY QGPL/SALES,SALETR

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*
F* Especificación de descripción de archivo tal como está almacenada
F* en el miembro del archivo fuente.
FNombarchIPEAF....LregLC1AIDePcLaEDispos+.....KSalidaEntradA....U1.*
FSALES IP F 43 OF DISK

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*
F* Sentencia modificadora de la función de copia.
FNombarchIPEAF....LregLC1AIDePcLaEDispos+.....KSalidaEntradA....U1.*
FSALES U

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*
F* Especificación de descripción de archivo resultante incluida en
F* el programa fuente en RPG/400.
FNombarchIPEAF....LregLC1AIDePcLaEDispos+.....KSalidaEntradA....U1.*
FSALES UP F 43 OF DISK
```

Figura 139. Modificación de una especificación de description de archivo copiada

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF....LregLC1AIDePcLaEDispos+.....KSalidaEntradA....U1.*
FSALES &
F*
```

Figura 140. Establecimiento de una entrada de descripción de archivo copiada en blancos

La sentencia modificadora cambia todas las especificaciones de campos de entrada que se copien y que tengan el mismo nombre de campo. Si no existe ningún campo de entrada cuyo nombre sea igual, la sentencia modificadora se añade al programa como si fuera una especificación de campo de entrada nueva. Se permiten las sentencias modificadoras con nombres de campo duplicados (la longitud y el número de posiciones decimales tienen que ser iguales también), pero sólo se utiliza la primera para modificar una especificación que se copia. Otros nombres de campos se añaden como

Modificación de especificaciones de campos de entrada

especificaciones de nuevos campos de entrada. Cada sentencia /COPY puede tener un máximo de 20 sentencias modificadoras de campos de entrada.

Al objeto de obtener mejores resultados, deben situarse primero las sentencias que modifican especificaciones de campos de entrada existentes y, a continuación, las que tienen que añadirse como especificaciones de nuevos campos de entrada. Se sugiere este procedimiento ya que las sentencias modificadoras de campos de entrada que no caben en la tabla especial del almacenamiento principal para sentencias modificadoras, se añaden al programa fuente en RPG/400 como especificaciones de nuevos campos de entrada. Este orden al especificar las sentencias modificadoras aumenta la probabilidad de que, en caso de existir exceso de sentencias, éstas sean descripciones de campo válidas. La Figura 141 muestra ejemplos de modificación de especificaciones de entrada.

Las especificaciones de entrada se almacenan en un archivo fuente.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
ISALES AA 01
I.....PDesdeA+++DNomcamL1M1FrP1NeCe...*
I          1  7 ITEMNO
I          8  9 BRANCH
I         10 10 REGION
I         11 25 DESC
I         26 270SOLDQY
I         28 342SOLDVA          13
I         35 360ONHAND
I         37 432VALUE
I*
```

Sentencia /COPY y sentencias modificadoras:

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I/COPY SALETR
I          BRANCHL1 1
I          SOLDVA   & 2
I         1 43 RECORD 3
I*
```

Figura 141. Modificación de especificaciones de campo de entrada copiadas

- 1 Añadir una entrada a la descripción de campo **BRANCH**.
- 2 Borrar el indicador de campo negativo en la descripción **SOLDVA**.
- 3 Añadir una nueva descripción de campo.

Especificaciones de entrada resultantes para el archivo **SALES** mostrado:

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
ISALES AA 01
I.....PDesdeA+++DNomcamL1M1FrP1NeCe...*
I          1  7 ITEMNO
I          8  9 BRANCHL1 1
I         10 10 REGION
I         11 25 DESC
I         26 270SOLDQY
I         28 342SOLDVA          2
I         35 360ONHAND
I         37 432VALUE
I          1 43 RECORD 3
```

Figura 142. Modificación de especificaciones de campo de entrada copiadas

- 1 Añadido el indicador **L1**.
- 2 Blancos en lugar del indicador de campo negativo (posiciones 67 y 68).
- 3 Añadida la descripción de campo.

12.3 Formato de generador de informes

Una de las ventajas del generador automático de informes es que libera al programador de la tarea de especificar el formato del generador de informes en el formulario de especificaciones de salida. El generador automático de informes puede establecer el formato completo del informe, incluido el espaciado, salto, centrado de líneas o cálculo de posiciones finales para campos y constantes.

Subtemas

12.3.1 Espaciado y Salto

12.3.2 Ubicación de encabezamientos y campos

12.3.1 Espaciado y Salto

El usuario puede efectuar las especificaciones de espaciado y salto o dejar que lo haga el generador automático de informes. La Figura 143 en el tema 12.3.2.2 muestra el espaciado y el salto producidos por el generador automático de informes. Consulte en "Especificaciones generadas" en el tema 12.4 las especificaciones utilizadas para producir el informe. Si se dejan en blanco las posiciones 17 a 22 en una especificación **H-*AUTO**, se produce un salto a la línea 06 antes de imprimirse la primera línea de encabezamiento y se dejan dos espacios después de impresa la última línea de encabezamiento. Si se especifica más de una línea de encabezamiento, se produce un espacio después de la primera línea de encabezamiento y de las siguientes a excepción de la última. Para especificar el espaciado y el salto deben seguirse las normas estándares del RPG/400 para espaciado y salto.

Las líneas de encabezamiento de las columnas están espaciadas igual que los encabezamientos de página. Se produce un espacio después para todas las líneas excepto la última. Se producen dos espacios después para una sola línea de encabezamiento o para la última línea del encabezamiento si se especifican más de una. No se pueden especificar entradas de espaciado y de salto para los encabezamientos de columnas. Si se efectúan entradas de espaciado y salto en la especificación de descripción de un registro **D-*AUTO**, las entradas se aplican a la línea de detalle generada. Las entradas no se aplican a los encabezamientos de columnas o a las líneas de totales generadas por el generador automático de informes desde la especificación **D-*AUTO**. Deben seguirse las reglas estándares del RPG/400 para el espaciado y salto. Se supone un espacio después para la línea de detalle generada si no se efectúan las entradas de espaciado y salto.

Se generan dos espacios después para todas las líneas de totales producidas por el generador automático de informes a partir de una especificación **D-*AUTO**. Además, la línea de totales de nivel inferior y la línea de totales final se generan también con un espacio antes.

Si se efectúan entradas de espaciado y salto en una especificación **T-*AUTO**, las entradas son aplicables a la línea de totales de nivel más bajo que se genere, pero no a las cabeceras de columna o a las líneas de totales de niveles superiores. Si no se efectúan entradas de espaciado y salto, las líneas de totales de niveles más bajos se generan con un espacio después; todos los niveles superiores se generan con dos espacios después. Siempre se genera un espacio antes para el total del nivel que sigue al más bajo y para el total final (consulte Figura 136 en el tema 12.1.2 para obtener un ejemplo).

12.3.2 Ubicación de encabezamientos y campos

El generador automático de informes genera las posiciones finales para campos y constantes y centra los encabezamientos de columna, las columnas y las líneas del informe. (Consulte la Figura 143 en el tema 12.3.2.2 para ver un ejemplo.) Si se especifica una posición final para un campo o constante en una línea de descripción de campos **D/T-*AUTO**, esta posición final se utiliza en todas las especificaciones de encabezamiento de columna, de detalle, y de totales generadas a partir de la descripción del campo. (El generador automático de informes puede modificar ligeramente la posición final especificada cuando centra la línea o cuando posiciona el encabezamiento de columna y el campo, relacionándolos entre sí). Si la posición final especificada origina un recubrimiento con un campo o constante precedente, el generador automático de informes produce una nueva posición final.

Las últimas posiciones deben especificarse solamente para eliminar el espaciado automático entre campos o para ensanchar un informe de manera que ocupe toda la página.

Subtemas

- 12.3.2.1 Encabezamientos de página
- 12.3.2.2 Reasignación de formato de encabezamientos de página *AUTO
- 12.3.2.3 Cuerpo del informe
- 12.3.2.4 Desbordamiento de líneas de impresión D/T-*AUTO

12.3.2.1 Encabezamientos de página

Si se imprimen la fecha y el número de página en la línea de encabezamiento de primera página ***AUTO** (es decir, si no se suprimen entrando una **N** en la posición 27 de las especificaciones de opción mediante la opción ***NODATE** del parámetro RPTOPT en el mandato CRTRPTPGM), la fecha se imprime siempre en las posiciones 1 a 8. El número de página se imprime con una posición final igual a la posición final más elevada de la línea más larga del informe. Cuando el primer encabezamiento de página ***AUTO** (incluyendo fecha, título y número de página) es la línea más larga del informe, se produce un blanco de separación entre la fecha y el título, así como entre el título y la palabra PAGE. Si la línea resultante excede la longitud del registro del archivo de impresora, no se imprime la información en exceso a la derecha de la línea.

Si una línea generada a partir de una especificación **D/T-*AUTO** es la línea más larga del informe, dicha línea se imprime comenzando por la posición de impresión 1, y la parte del título de la línea de encabezamiento de la primera página se centra en función a esta línea.

Los encabezamientos de página ***AUTO** adicionales se centran entonces en la primera línea de encabezamiento de página ***AUTO**.

Si la línea más larga del informe es un encabezamiento de página ***AUTO** y existe una especificación **D/T-*AUTO**, todas las demás líneas de encabezamiento de páginas ***AUTO**, así como las líneas generadas a partir de la especificación **D/T-*AUTO** se centran sobre el encabezamiento de página más largo.

Los campos y constantes aparecen en el orden especificado en las especificaciones de salida ***AUTO**, colocados de izquierda a derecha. El generador automático de informes proporciona un espacio en blanco que precede a los campos de la línea de encabezamiento y otro que aparece detrás. No se proporcionan espacios entre constantes.

*12.3.2.2 Reasignación de formato de encabezamientos de página *AUTO*

Puede volver a dar formato a una línea de encabezamiento de una página ***AUTO** si no desea utilizar las posiciones finales para campos y constantes que genera el generador automático de informes. Si desea averiguar cuáles son las posiciones finales que se generan para proporcionar información sobre la página, la fecha y el título, consulte la lista del programa fuente generado que origina el compilador del RPG/400. Consulte "Especificaciones generadas" en el tema 12.4.

IMAGEN 28

Figura 143. Informe ilustrativo del formato generado por el generador automático de informes

12.3.2.3 Cuerpo del informe

La ubicación de los encabezamientos de columnas sobre las columnas está en función de la longitud del encabezamiento o el campo asociado (incluidos los caracteres de edición). Si un encabezamiento de columnas es más largo que el campo con el que está asociado, se centra el campo bajo la constante de encabezamiento de columna más larga. Sin embargo, si el campo es más largo que la constante de encabezamiento de la columna más larga, el encabezamiento de columna se ajusta por la izquierda sobre un campo de caracteres y se ajusta por la derecha sobre un campo numérico. Cuando se especifica más de una línea de encabezamientos de columna, los encabezamientos de columna más cortos se centran siempre con respecto al encabezamiento de columna más largo.

Los campos y las constantes aparecen de izquierda a derecha de la línea en el orden en que estén especificadas en las especificaciones de salida. Aparecen por lo menos dos espacios en blanco delante de cada campo de la línea. No se proporciona ningún espacio delante de una constante; el programador debe incluir blancos entre las constantes para proporcionar espaciado adicional.

La información de totales (campos y constantes especificados con un **1** a **9** o **R** en la posición 39) se sitúa a la izquierda del primer campo de totales (**A** en la posición 39) de la línea de totales correspondiente y que va seguida de dos espacios. Si se especifican dos o más campos o constantes de este tipo para una línea de totales, aparecen a la izquierda del primer total en la línea, de izquierda a derecha, en el orden en que se han especificado. Cada campo va precedido y seguido por un espacio. No se proporcionan espacios para constantes.

12.3.2.4 Desbordamiento de líneas de impresión D/T-*AUTO

Si las líneas generadas a partir de una especificación **D/T-*AUTO** son más largas que la longitud de registro especificado para el archivo de impresora, se genera una segunda línea de impresión (línea de desbordamiento) para cada columna de línea de encabezamiento, línea de detalle (o impresión de grupo) y línea de totales. (Recuerde que no se genera una segunda línea de impresión para las líneas de encabezamiento ***AUTO**.) La información en exceso se sitúa en la línea de desbordamiento en el orden especificado, y ajustado por la derecha.

La figura Figura 144 muestra el resultado de una condición de desbordamiento.

En las especificaciones de salida para el informe que se muestran en la Figura 144, no se ha especificado espaciado ni salto. Sin embargo, si se hubiesen especificado, el generador automático de informes espaciaría el informe de la siguiente forma:

- Las líneas de encabezamiento de columna y las líneas de totales se espacian como se muestra en la Figura 144.
- Las entradas de espaciado antes y salto antes que se han especificado son para la línea de desbordamiento. El generador automático de informes genera un espaciado después para esta línea.
- Las entradas de espacio después y salto después que se especifiquen son para la línea de desbordamiento. El generador automático de informes genera blancos para el espaciado antes y un salto antes para la línea de desbordamiento.

IMAGEN 29

Figura 144. Informe ilustrativo del desbordamiento de líneas de impresión D-*AUTO

12.4 Especificaciones generadas

El generador automático de informes genera las especificaciones estándares en RPG/400 y las combina con las especificaciones del RPG/400 incluidas en la entrada del generador automático de informes y las especificaciones copiadas del miembro del archivo fuente para producir el programa fuente final en RPG/400. Esta sección describe las especificaciones generadas en RPG/400 y el orden de estas especificaciones en el programa fuente en RPG/400.

La Figura 145 en el tema 12.4.2 y la Figura 146 en el tema 12.4.2 muestran las especificaciones del generador automático de informes para un informe de ventas y las especificaciones fuente del RPG/400 resultantes que se generan para este informe. Los números que se insertan en las figuras sirven para mostrar las especificaciones que genera cada función.

Subtemas

12.4.1 Cálculos generados

12.4.2 Especificaciones de salida generadas

12.4.1 Cálculos generados

Se generan los cálculos para acumular totales para los campos que se incluyen en las especificaciones de descripción de campos **AUTO** y que tienen una **A** en la posición 39. (Consulte Figura 147 en el tema 12.4.2.)

Se genera una subrutina en RPG/400 para acumular los valores de estos campos en los campos de totales de nivel más bajo generados. El nombre de la subrutina siempre es **A\$\$SUM**. Las especificaciones de la subrutina están condicionadas de forma diferente, dependiendo de si se especifica la impresión de detalle o por grupos:

- Si se especifica impresión de detalle, tal como se hace en la Figura 147 en el tema 12.4.2, la sentencia **EXSR** está condicionada por el mismo indicador (o indicadores) que condicionan la especificación **D-*AUTO** (01 en este ejemplo). Cada sentencia **ADD** en la subrutina está condicionada por el indicador (o indicadores) de campo que se haya(n) especificado para el campo en su especificación de descripción de campos (ninguno en este ejemplo).
- Si se especifica impresión por grupos, la sentencia **EXSR** y todas las sentencias **ADD** en la subrutina no estarán condicionadas.

Se generan cálculos de totales para remontar el total de los campos de totales definidos a nivel inferior hasta los campos de totales de niveles superiores que se hayan definido y el total final. El cálculo de totales para añadir el total de un nivel al del siguiente nivel superior se condiciona mediante un indicador de nivel de control que corresponde al nombre del campo de nivel inferior. Según se muestra en la Figura 147 en el tema 12.4.2, los cálculos de totales para acumular los totales por **L2** y **LR**, van seguidos por la subrutina que acumula el total de nivel más bajo, **L1**.

Los campos de totales generados se definen (se asigna longitud y número de posiciones decimales) cuando el campo de totales es el campo del resultado en un cálculo generado. En las especificaciones de entrada, **SOLDVA** y **VALUE** son campos numéricos que se definen con una longitud de siete y con dos posiciones decimales. La Figura 147 en el tema 12.4.2 muestra que los campos de totales generados a partir de **SOLDVA** y **VALUE** se definen con dos posiciones más que los campos originales y con el mismo número de posiciones decimales.

Cuando se especifica la impresión por grupo (especificación **T-*AUTO**), el generador automático de informes genera los cálculos de totales para restablecer cada uno de los campos acumulados (**A** en la posición 39) de la línea de totales de nivel más bajo a cero en cada ciclo. Se genera un cálculo **Z-ADD**, condicionado por **LO** para cada campo acumulado. Estos cálculos son los primeros cálculos de totales del programa fuente generado en RPG/400.

12.4.2 Especificaciones de salida generadas

La Figura 148 muestra las especificaciones de salida generadas por el generador automático de informes. Para identificar las especificaciones suministradas por el generador automático de informes (especificaciones de cabecera de columna, especificaciones de totales, indicadores de condicionamiento, valores de espaciado y salto, valores de posiciones finales, espaciado después), compare la lista con las especificaciones del generador automático de informes.

El generador automático de informes genera especificaciones para restablecer los campos acumulados a cero después de que se imprimen. En este ejemplo, se genera un blanco después para los campos acumulados.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF...LregLCIAIDePcLaEDispos+.....KSalidaEntradA....U1.*
FPRINT O F 120 OA PRINTER 1
F*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I*
I* Las especificaciones de entrada o descripción de archivos del
I* archivo PRINT se almacenan en el miembro del archivo fuente SALETR.
I/COPY SALETR 2
I*
I* Las sentencias modificadores siguen a la sentencia /COPY para
I* añadir indicadores de nivel de control.
INomreg++...In.....*
I BRANCHL1
I REGIONL2

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
OPRINT H *AUTO ----+
O 'INFORME VENTAS' | 3
O 'PARA COMPAÑIA' |
O D 01 *AUTO ----+
O L2 REGION 'REGION' ----+
O L1 BRANCH 'SUCURSAL' |
O ITEMNO 'NUMERO' |
O C 'ARTICULO' |
O DESC 'DESCRIPCION' | 4
O SOLDQV 'VENTAS' |
O +-- SOLDVA A 'IMPORTE' |
O | ONHAND 'EXISTENCIAS' |
O +-- VALUE A 'VALOR' |
O | R 'TOTAL FINAL' |
O* ----+

5
```

Figura 145. Especificaciones de generador automático de informes para un informe de transacción de ventas

Nota: Las siguientes claves también hacen referencia a los números correspondientes en el programa fuente generado que se muestra en la Figura 146.

- 1 Descripción del archivo de impresora
- 2 Función de copia y sentencias modificadoras
- 3 Función de encabezamiento de página ***AUTO**
- 4 Función de salida ***AUTO**
- 5 Campos acumulados

Si no especifica una especificación de control, el generador automático de informes le genera una en blanco.

IMAGEN 30

Figura 146. Programa fuente en RPG/400 generado a partir de especificaciones del generador automático de informes

Nota: Estos números hacen referencia a los números correspondientes mostrados en las especificaciones del generador automático de informes que se muestra en la Figura 145.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
O.....N01N02N03Campo+YBUposPConstante/palabraedición+...*
OPRINT H *AUTO
O
O 'INFORME DE VENTAS'
O 'PARA UNA COMPAÑIA'
O D 01 *AUTO
O L2 REGION 'REGION'
O L1 BRANCH 'SUCURSAL'
O ITEMNO 'NUMERO'
O C 'ARTICULO'
O DESC 'DESCRIPCION'
O SOLDQY 'VENTAS'
O SOLDVA A 'IMPORTE'
O ONHAND 'EXISTENCIAS'
O VALUE A 'VALOR'
O R 'TOTAL FINAL'
O*
```

O* Se generan cálculos para los campos con una A en la posición 39.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgcComentarios+++*
C--01 EXSR A$$$SUM
CL1| SOLDV2 ADD SOLDV1 SOLDV2 92
CL1| 1 VALUE2 ADD VALUE1 VALUE2 92
CL2| SOLDVR ADD SOLDV2 SOLDVR 92
CL2+ VALUER ADD VALUE2 VALUER 92
CSR+ A$$$SUM BEGSR
CSR| 2 SOLDV1 ADD SOLDVA SOLDV1 92 3
CSR| VALUE1 ADD VALUE VALUE1 92
CSR+ ENDSR
```

Figura 147. Cálculos generados a partir de la estructura de datos para acumular totales - Ejemplo 2

- 1 Los cálculos de totales se acumulan a los totales de nivel superior.
- 2 La subrutina acumula los totales de nivel inferior (L1 en este ejemplo).
- 3 Longitud y posiciones decimales de los campos de totales generados.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
O.....N01N02N03Campo+YBUposPConstante/palabraedición+...*
OPRINT H *AUTO
O
O 'INFORME DE VENTAS'
O 'PARA UNA COMPAÑIA'
O D 01 *AUTO
O L2 REGION 'REGION'
O L1 BRANCH 'SUCURSAL'
O ITEMNO 'NUMERO'
O C 'ARTICULO'
O DESC 'DESCRIPCION'
O SOLDQY 'VENTAS'
O SOLDVA A 'IMPORTE'
O ONHAND 'EXISTENCIAS'
O VALUE A 'VALOR'
O R 'TOTAL FINAL'
O*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
O.....N01N02N03Campo+YBUposPConstante/palabraedición+...*
OPRINT H 206 1P
O OR OA
O UPDATE Y 8
O 45 'INFORME DE VENTAS'
O 56 'PARA UNA COMPAÑIA'
O 85 'PAGINA'
O PAGE Z 89
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
O.....N01N02N03Campo+YBUposPConstante/palabraedición+...*
OPRINT H 1 1P
O OR OA
O -----+
O 6 'REGION' |
O 14 'SUCURSAL' |
O 21 'ARTICULO' |
O 36 'DESCRIPCION' | 1
O 47 'VENTAS' |
O 62 'IMPORTE' |
O 71 'EXISTENCIAS' |
```

Especificaciones de salida generadas

```

O                               86 'VALOR' -----+
OPRINT   H  2      1P
O         OR      OA
O                               22 'NUMERO'
OPRINT   D  1      01
O                               L2      REGION      3
O                               L1      BRANCH      12
O                               ITEMNO     12
O                               DESC        12
O                               SOLDQYK    46
O                               SOLDVAKB   62
O                               ONHANDK    69
O                               VALUE KB   86

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...*
O.....N01N02N03Campo+YBUposPConstante/palabraedición+...*
OPRINT   T 12      L1
O                               SQLDV1KB   62
O                               VALUE1KB   86
O                               87 '*'
OPRINT   T  2      L2
O                               SQLDV2KB   62
O                               VALUE2KB   86
O                               88 '***'
OPRINT   T 12      LR
O                               SQLDVRKB   62
O                               VALUERKB   86
O                               47 'TOTAL FINAL'
O                               89 '***' -----+

```

Figura 148. Especificaciones de salida generadas estructura de datos para acumular totales - Ejemplo 2

- 1 Se generan dos especificaciones de encabezamiento para los encabezamientos de columna ya que **NUMERO ARTICULO** es un encabezamiento de dos líneas.
- 2 El generador automático de informes genera especificaciones de totales para imprimir los totales acumulados para los campos **SOLDVA** y **VALUE**.

Guía del usuario RPG/400
Ayudas de programación

		indicadores	campo	código de edición	posición final		encabez. columna
A	Blanco	Blanco o indicadores	Nombre campo	Blanco o código de edición	Blanco o posición final	Blanco	Blanco o encabez. columna
C	Blanco	Blanco	Blanco	Blanco	Blanco	Blanco	Encabez. columna
1-9, R	Blanco	Blanco	Nombre campo	Blanco o código de edición	Blanco	Blanco	Blanco o palabra de edición
	Blanco	Blanco	Blanco	Blanco	Blanco	Blanco	Constante

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* Se asume que L1 se ha definido en las posiciones 59 y 60 de las
C* especificaciones de entrada.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C  L1          MOVE 'CONSTANTE' CPOA 8

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
O.....N01N02N03Campo+YBUpoSPConstante/palabraedición++...*
O          D          *AUTO
O          CPOA  B          'ENCABEZAMIENTO COLUMNA'
O*
```

Figura 149. Impresión de una constante sólo en línea de detalles

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
IENTRA  AA 10 1 CA
I.....PDesdeA+++DNomcamL1M1FrP1NeCe...*
I          2 27 NAME L1
I          BB 11 1 C1
I          OR 12 1 CN
I          2 18 DESC
I          19 210QTY
I          22 262SALES
I*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OPRINT  T          L1          *AUTO
O.....N01N02N03Campo+YBUpoSPConstante/palabraedición++...*
O          DESC          'DESCRIPCION'
O          QTY  A          'IMPORTE'
O          SALES A          'VENTAS'
O*
```

Figura 150. Especificaciones *AUTO incorrectas para más de un tipo de registro

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          11          Z-ADDQTY          QTYA  30
C          11          Z-ADDSALES          SALES  52
C*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OPRINT  T          L1          *AUTO
O.....N01N02N03Campo+YBUpoSPConstante/palabraedición++...*
O          DESC          'DESCRIPCION'
O          QTYA  A          'IMPORTE'
O          SALES A          'VENTAS'
O*
```

Figura 151. Especificaciones *AUTO correctas para más de un tipo de registro

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C      1                      Z-ADD0          COUNT  30
C      2      1          ADD COUNT1          COUNT1
C*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdn01N02N03Nomexc.....*
O      D                      *AUTO
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O      3          LR          COUNT  A      'REGISTRO'
O                                  C      'CUENTA'
O*
```

Figura 152. Método de utilización de *AUTO para contar registros

Especificaciones de cálculo

- 1 Esta instrucción sólo se necesita para definir el campo **COUNT** a efectos de acumulación.
- 2 Esta instrucción acumula el total para el primer nivel de control.

Especificaciones de salida

- 3 Esta instrucción produce la generación de las especificaciones de cálculo y de salida para las líneas de detalle y de totales. El indicador de condicionamiento **LR** impide que se efectúen los cálculos de detalle generados. También impide la impresión en tiempo de detalle.

Nota: Si no se especifican niveles de control en el programa, se suma 1 a **COUNTR** en lugar de a **COUNT1** en las especificaciones de cálculo.

12.6 Utilización de CRTRPTPGM para compilar un programa generador automático de informes

Para compilar un programa fuente en RPG/400 que incluya las especificaciones del generador automático de informes debe utilizar el mandato de CL CRTRPGPGM (Crear programa generador automático de informes). Los objetos del programa en RPG/400 se crean con la autorización de uso público ***CHANGE**. Puede que desee modificar esta autorización para mantener una mayor seguridad en el sistema.

El generador automático de informes no diagnostica todas las condiciones de error del programa fuente. El generador automático de informes no duplica los resultados de prueba generador por el compilador del RPG/400. Si un programa no puede generarse satisfactoriamente debido a errores en las especificaciones del generador automático de informes, el generador automático de informes se termina. Si se detiene el generador automático de informes, se emite el mensaje de escape **RPT 9001**. Un programa en CL puede supervisar el mensaje de escape utilizando el mandato de CL MONMSG (Supervisar mensaje).

Si se genera satisfactoriamente un programa fuente en RPG/400 y no se especifica la opción ***NOCOMPILE** en el mandato CRTRPTPGM, el informe automático llama al compilador del RPG/400.

Todos los nombres de objetos especificados en el mandato CRTRPTPGM utilizan totalmente las convenciones de denominación. La longitud de los nombres no puede exceder de diez caracteres. Consulte el manual *CL Reference* para obtener una descripción detallada de las reglas de denominación de objetos en OS/400 así como una descripción completa de la sintaxis de los mandatos en OS/400.

Subtemas

12.6.1 Utilización del mandato CRTRPTPGM

12.6.2 Mandato CRTRPTPGM

12.6.1 Utilización del mandato CRTRPTPGM

Puede llamar al compilador del CRTRPTPGM de tres formas:

- Interactivamente desde una pantalla. Teclee el mandato CRTRPTPGM y a continuación pulse F4
- Utilizando parámetros de palabra clave
- Utilizando parámetros posicionales.

Consulte los detalles de estas funciones del mandato CRTRPGPGM en Capítulo 3, "Compilación de un programa en RPG/400" en el tema 3.0.

12.6.2 Mandato CRTRPTPGM

El mandato CRTRPTPGM es semejante al mandato CRTRPGPGM descrito en Capítulo 3, "Compilación de un programa en RPG/400" en el tema 3.0. Todos los nombres de objeto deben estar formados por caracteres alfanuméricos. El primer carácter siempre tiene que ser alfabético y la longitud del nombre no puede exceder de diez caracteres.

El mandato CRTRPTPGM reconoce todos los parámetros que realiza el mandato CRTRPGPGM. Sin embargo, el generador automático de informes no utiliza algunos de estos parámetros sino que los pasan al compilador del RPG/400. Se trata de los parámetros PGM, OPTION, GENOPT, GENLVL, USRPRF, AUT, TEXT, PHSTRC, TGTRLS, INDENT y REPLACE. El parámetro PRTFILE especifica un archivo que el generador automático de informes utiliza y después pasa al compilador del RPG/400.

El mandato CRTRPTPGM tiene los mismos parámetros que el mandato CRTRPGPGM más otros tres: RPTOPT, OUTFILE y OUTMBR. La descripción de estos parámetros está a continuación del diagrama de sintaxis para el mandato CRTRPTPGM. Los valores por omisión se explican en primer lugar. Consulte "Mandato crear programa en RPG400 (CRTRPGPGM)" en el tema 3.1 para obtener la definición de los demás parámetros.

El mandato CRTRPTPGM puede someterse en una corriente de entrada por lotes, entrada de forma interactiva en una estación de trabajo o en un programa en CL o REXX.

Para obtener información sobre cómo leer diagramas de sintaxis, consulte "Cómo interpretar diagramas de sintaxis" en el tema PORTADA_2.2.

```

-----
>>--CRTRPTPGM-----
|          +-*CURLIB/-----+ +-*CTLSPEC-----+ |
+-PGM--(--+-----+-----+-----+-----+--)+
|          +-nombre-biblioteca/-+ +-nombre-programa/-+
>-----
|          +-*LIBL/-----+ +-QRPGRSRC-----+ | | +-*PGM-----+
+-SRCFILE--(--+-----+-----+-----+-----+--)+ +-SRCMBR--(-----nombre-miembro-archi-+
|          +-*CURLIB/-----+ +-nombre-archivo-fuente+
+-nombre-biblioteca/+
>-----
|          +-9-----+ | | +-*SRCMBRTXT-----+ | +-OPTION--(--| Detalles
+-GENLVL--(-----valor-nivel-gravedad-----)+ +-TEXT--(--+*BLANK-----+--)+
|          +- 'descripción'--+
>-----
|          +-*NONE-----+ | | +-*NONE-----+
+-GENOPT--(--| Detalles GENOPT +--)+ | +-INDENT--(-----valor-carácter-----)+ +-CVTOPT--(-----| Detalles
>-----
|          +-*LIBL/-----+ +-QSYSVRT-----+ | +-RPTOPT--(--| Detalles RPTOPT +--)+
+-PRTFILE--(--+-----+-----+-----+-----+--)+
|          +-*CURLIB/-----+ +-nombre-archivo-+
+-nombre-biblioteca/+
>-----
|          +-*LIBL/-----+ +-*NONE-----+ | | +-*NONE-----+
+-OUTFILE--(--+-----+-----+-----+-----+--)+ +-OUTMBR--(-----nombre-miembro-archivo-f
|          +-*CURLIB/-----+ +-nombre-archivo-+
+-nombre-biblioteca/+
>-----
|          +-*YES--+ | | +-*CURRENT-----+ | | +-*USER--+ |
+-REPLACE--(-----*NO-----)+ +-TGTRLS--(--+*PRV-----+--)+ +-USRPRF--(-----*OWNER-----)+
|          +-nivel-release+
>-----
|          +-*LIBCRTAUT-----+ | | +-*NO--+ | | +-*NONE-----+
+-AUT--(--+*CHANGE-----+--)+ +-PHSTRC--(-----*YES-----)+ | | | <
|          +-*USE-----+ | | |
+-*ALL-----+ | | | +-ITDUMP--(-----nombre-fa
|          +-*EXCLUDE-----+ | | |
+-nombre-lista-autorizaciones+
>-----

```


encabezamiento de página ***AUTO**.

***NOAST**

No genera indicación mediante asteriscos para las líneas de salida de totales.

***AST**

Genera indicación mediante asteriscos para las líneas de salida de totales.

***NOSECLVL**

No imprime el texto de segundo nivel en la línea siguiente al texto del mensaje de primer nivel.

***SECLVL**

Imprime el texto de segundo nivel en la línea siguiente al texto del mensaje de primer nivel.

OUTFILE

Especifica el nombre del archivo en el que ha de colocarse el programa fuente en RPG/400 y la biblioteca en la que se encuentra este archivo. El archivo se utiliza también como el archivo fuente de entrada al compilador del RPG/400 a menos que en el parámetro RPTOPT esté especificado el valor ***NOCOMPILE**.

***LIBL**

Se utiliza la lista de bibliotecas para ubicar el archivo.

***CURLIB**

El nombre de la biblioteca actual. Si no se ha especificado una biblioteca actual, QGPL es la biblioteca actual.

nombre-biblioteca

Entre el nombre de la biblioteca en la que está ubicado el archivo.

***NONE**

Crea un archivo en QTEMP para trasladar el fuente que se ha generado en RPG/400 al compilador del RPG/400.

nombre-archivo

Entre el nombre del archivo que va a contener el programa fuente en RPG/400 completo.

OUTMBR

Especifica el nombre del miembro del archivo que contendrá la salida del generador automático de informes.

***NONE**

Utiliza el primer miembro que se ha creado en el archivo o se ha añadido al mismo como nombre de miembro.

nombre-miembro-archivo

Entre el nombre del miembro que deberá contener la salida del generador automático de informes.

12.7 Ejemplos de utilización del generador automático de informes

Los ejemplos 1 a 4 explican cómo se utiliza el generador automático de informes para producir encabezamientos de página del informe y especificaciones de salida, como por ejemplo encabezamiento de columnas, líneas de detalle y líneas de totales. Los ejemplos 5 y 6 muestran la utilización de la función de copia del generador automático de informes para copiar especificaciones desde un miembro de archivo fuente y para modificar las especificaciones copiadas para un trabajo específico.

Subtemas

- 12.7.1 EJEMPLO 1 - Informe de ventas
- 12.7.2 EJEMPLO 2 - Informe de ventas con tres niveles de totales
- 12.7.3 EJEMPLO 3 - Informe de ventas con indicación de grupo
- 12.7.4 EJEMPLO 4 - Informe de ventas con totales en columnas cruzadas
- 12.7.5 EJEMPLO 5 - Informe de ventas utilizando especificaciones copiadas
- 12.7.6 EJEMPLO 6 - Alteración temporal de especificaciones de entrada copiadas

12.7.1 EJEMPLO 1 - Informe de ventas

Problema	Procedimiento
<p>Producir el siguiente informe de ventas utilizando las funciones *AUTO para encabezamiento de página y para la salida del generador automático de informes.</p>	<ol style="list-style-type: none"> Codifique las especificaciones del RPG normales para describir archivos y la entrada necesaria para ejecutar el trabajo. Codifique los encabezamientos de página *AUTO para producir una línea de encabezamiento de página que incluya la fecha y el número de página. Codifique la salida *AUTO necesaria para generar encabezamientos de columna y totales finales.

IMAGEN 31

1 Especificaciones de descripción de archivos y de entrada de un archivo en RPG/400

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF....LregLCIAIDePcLaEDispos+.....KSalidaEntrada....U1.*
FSALES  IP  F      43          DISK
FPRINT  O  F      120         PRINTER
F*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
ISALES  AA  01
I.....PDesdeA+++DNomcamL1M1FrP1NeCe...*
I              1  7 ITEMNO
I              8  9 BRANCH
I             10 10 REGION
I             11 25 DESC
I             26 27SOLDQY
I             28 34SOLDVA
I             35 36ONHAND
I             37 43VALUE
I*
```

	Nombre del Campo	Contenido
A	ITEMNO	Número de artículo
B	BRANCH	Número de la sucursal que vendió el artículo
C	REGION	Región de ventas en la que se encuentra la sucursal
D	DESC	Descripción del artículo vendido
E	SOLDQY	Cantidad vendida del artículo
F	SOLDVA	Valor total de los artículos vendidos
G	ONHAND	Cantidad del artículo que queda en existencia
	VALUE	Valor total de los artículos que quedan en

```
| H | | existencia |
+-----+
```

2 Especificaciones de encabezamiento de página *AUTO

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OPRINT H A C D *AUTO
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O 'INFORME DE VENTAS' B
O 'PARA UNA COMPAÑIA'
O*
```

- A Entre una **H** en la posición 15 y ***AUTO** en las posiciones 32 a 36 para solicitar un encabezamiento de página del generador automático de informes. Pueden describirse cinco líneas de encabezamiento como máximo. La fecha del sistema se imprime a la izquierda y el número de página a la derecha de la primera línea de encabezamiento en cada página. Para suprimir la fecha y la página, entre una **N** en la posición 27 de las especificaciones de la opción del generador automático de informes o utilice la opción ***NODATE** del parámetro RPTOPT del mandato CRTRPTPGM.
- B El generador automático de informes centra la información del título; no entre ninguna posición en las posiciones 40 a 43. También se pueden utilizar campos y elementos de tabla/matriz.
- C Cuando se dejan en blanco las entradas de espaciado y salto (posiciones 17 a 22), se asume el salto a la línea 06 para la primera línea de encabezamiento; las líneas de encabezamiento se imprimen a espaciado sencillo, se produce un doble espaciado después de la última línea de encabezamiento. (Consulte en "Ejemplo 4" un ejemplo de encabezamiento de página de múltiples líneas).
- D Cuando los indicadores de salida (posiciones 23 a 31) se dejan en blanco, los encabezamientos de página del generador automático de informes se imprimen en cada página (condicionados por 1P o desbordamiento). Si no se define ningún indicador de desbordamiento para el archivo de impresora, el generador automático de informes asigna un indicador de desbordamiento que no se utilice a la línea de impresora.

3 Codifique las especificaciones de salida ***AUTO** para generar:

IMAGEN 32

- A Líneas de detalle del informe
- B Encabezamientos de columna
- C Totales finales

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OPRINT H *AUTO
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O 'INFORME DE VENTAS'
O 'PARA UNA COMPAÑIA'
O A D 01 *AUTO -----+
O REGION 'REGION' |
O BRANCH 'SUCURSAL' |
O ITEMNO 'ARTICULO' |
O DESC 'DESCRIPCION' | B
O SOLDQY 'VENTAS' |
O SOLDVA A 'IMPORTE' |
O ONHAND 'EXISTENCIAS' |
O VALUE A C 'VALOR' -----+
O*
```

- A Entre una **D** en la posición 15 y ***AUTO** en las posiciones 32 a 36 para describir un informe generado por el generador automático de informes con líneas de detalle. El indicador identificador de registro 01 condiciona la impresión de las líneas de detalle.

- B Los encabezamientos de columna se entran en la misma línea que los campos sobre los que aparecen en el informe.

- C Entre una **A** en la posición 39 para hacer que los campos se acumulen. El generador automático de informes genera (1) los campos de totales y los cálculos para acumular los totales y (2) las especificaciones de salida de totales para imprimir los totales.

IMAGEN 33

El generador automático de informes da formato al informe de forma que los encabezamientos de columna y los datos queden limpiamente espaciados y centrados unos respecto a otros.

Todos los campos numéricos para los que se haya especificado un blanco, una **B**, o una **A** en la posición 39, se editan mediante el código de edición **K** a menos que se haya especificado un código de edición distinto.

EJEMPLO 2 - Informe de ventas con tres niveles de totales

12.7.2 EJEMPLO 2 - Informe de ventas con tres niveles de totales

+-----+ Problema+-----+ +-----+	+-----+ Salida *AUTO+-----+ +-----+
Ampliar el informe de ventas del Ejemplo 1 para incluir tres niveles de totales: 1. Total por cada sucursal 2. Total por cada región 3. Total final	(1). Codificar las especificaciones de descripción de archivos y de entrada como en el Ejemplo 1. (2). Añadir indicadores de nivel de control a campos de entrada BRANCH y REGION.

Nota: La función de salida ***AUTO** puede utilizarse también para generar un informe impreso por grupos. Consulte en "Impresión por grupos" en el tema 12.1 una exposición y los ejemplos de la impresión por grupos.

1 Especificaciones de descripción de archivos y de entrada de un archivo en RPG/400

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
ISALES AA 01
I.....PDesdeA+++DNomcamL1M1FrP1NeCe...*
I          1  7 ITEMNO
I          8  9 BRANCHL1  2
I         10 10 REGIONL2
I         11 25 DESC
I         26 27SOLDQY
I         28 342SOLDVA  A
I         35 36ONHAND
I         37 432VALUE
I*
```

A Como se han definido dos niveles de control, los campos **SOLDVA** y **VALUE** se acumulan a dos niveles de totales (sucursal y región) y a un total final (**LR**).

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OPRINT H *AUTO
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O          'INFORME DE VENTAS'
O          'PARA UNA COMPAÑIA'
O      D  A      01      *AUTO
O          REGION      'REGION'
O          BRANCH      'SUCURSAL'
O          ITEMNO      'ARTICULO'
O          DESC        'DESCRIPCION'
O          SOLDQY      'VENTAS'
O          SOLDVA A  B  'IMPORTE'
O          ONHAND      'EXISTENCIAS'
O          VALUE  A    'VALOR'
O*
```

A El generador automático de informes coloca una línea en blanco después de cada línea de totales y una línea en blanco adicional antes del total de nivel más bajo y antes del total final. Si entra valores de espaciado y salto en la especificación **D-*AUTO**, éstas sólo se aplican a la línea de detalla de impresión.

B Como en "EJEMPLO 1 - Informe de ventas", una **A** en la posición 39 de la especificación de salida hace que se acumulen **SOLDVA** y **VALUE**.

C Los campos de totales siempre ocupan dos posiciones más que los

EJEMPLO 2 - Informe de ventas con tres niveles de totales

campos originales, y tienen el mismo número de posiciones decimales.

- D El generador automático de informes imprime los asteriscos (*) a la derecha de la línea de totales que se ha generado para ayudarle a identificarla. Si desea suprimir los asteriscos, entre una **N** en la posición 28 de las especificaciones de opciones del generador automático de informes o utilice la opción ***NOAST** en el parámetro RPTOPT del mandato CRTRPTPGM.

IMAGEN 34

12.7.3 EJEMPLO 3 - Informe de ventas con indicación de grupo

+-----+ Problema+-----+ +-----+	+-----+ Salida *AUTO+-----+ +-----+
<p>Ampliar el informe de ventas de los Ejemplos 1 y 2 para que incluyan:</p> <p>(A). Indicación de grupo para los campos REGION y BRANCH.</p> <p>(B). Segunda línea de encabezamiento de columnas.</p> <p>(C). Literal (constante) en la línea de totales final.</p>	<p>(1). Codificar las especificaciones de descripción de archivos y de entrada como en el Ejemplo 2.</p> <p>(2). Codificar la salida *AUTO con:</p> <p>(A) Indicador de salida en las especificaciones de descripción de campos.</p> <p>(B) C en la posición 39 y una constante en las posiciones 45 a 70.</p> <p>(C) R en la posición 39 y una constante en las posiciones 45 a 70.</p>

1 Especificaciones de descripción de archivos y de entrada de un archivo en RPG/400

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF....LregLCIAIDePcLaEDispos+.....KSalidaEntrada....U1.*
FSALES  IP  F      43          DISK
FPRINT  O   F      120         PRINTER
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
ISALES  AA  01
I.....PDesdeA+++DNomcamL1M1FrP1NeCe...*
I              1  7  ITEMNO
I              8  9  BRANDHL1
I             10 10  REGIONL2
I             11 25  DESC
I             26 270SOLDQY
I             28 342SOLDVA
I             35 360ONHAND
I             37 432VALUE
I*
```

2 Salida *AUTO.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdn01N02N03Nomexc.....*
OPRINT  H          *AUTO
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O              'INFORME DE VENTAS'
O              'PARA UNA COMPAÑIA'
O          D      01      *AUTO
O          A      L2      REGION      'REGION'
O          L1      BRANCH      'SUCURSAL'
O              ITEMNO      'ARTICULO'
O              B  C        'NUMERO'
O              DESC        'DESCRIPCION'
O              SOLDQY      'VENTAS'
O              SOLDVA A    'IMPORTE'
O              ONHAND     'EXISTENCIAS'
O              VALUE A    'VALOR'
O              C  R        'TOTALES FINALES'
O*
```

A Pueden utilizarse indicadores de salida en especificaciones de descripción de campos. En este ejemplo, los indicadores de nivel de control condicionan **BRANCH** y **REGION** de forma que se imprimen sólo para el primer registro del grupo de control correspondiente. Esta

EJEMPLO 3 - Informe de ventas con indicación de grupo

supresión de impresión de campos comunes (indicación de grupo)
reduce la información repetitiva.

- B Puede especificar una o dos líneas de encabezamiento de columna adicionales mediante una entrada **C** en la posición 39 con la información de encabezamiento en las posiciones 45 a 70.
- C El literal **TOTALES FINALES** facilita la localización de dicha línea. Para especificar la información que tiene que aparecer en la línea de total final, tiene que introducirse una **R** en la posición 39 y una constante en las posiciones 45 a 70 o un nombre de campo/nombre de tabla/nombre de matriz indexada en las posiciones 32 a 37. La información se imprime dos espacios a la izquierda del total más a la izquierda de la línea. Si se utiliza más de una especificación de este tipo, las constantes y los campos se imprimen de izquierda a derecha en el mismo orden en que se especifican en el programa.

IMAGEN 35

EJEMPLO 4 - Informe de ventas con totales en columnas cruzadas

12.7.4 EJEMPLO 4 - Informe de ventas con totales en columnas cruzadas

+-----+ Encabezamientos de página *AUTO +-----+	+-----+ Salida *AUTO +-----+
+-----+ Problema+-----+ +-----+	□ +-----+ □ +-----+
<p>Ampliar el informe de ventas de los Ejemplos 1, 2 y 3 para incluir una columna de totales de control y:</p> <p>(A). Una página de informe nueva para cada región.</p> <p>(B). Dos líneas de encabezamiento en cada página.</p> <p>(C). Un campo en una línea de encabezamiento de página</p> <p>(D). Identificación de los totales por sucursal y región.</p>	<p>(1). Codificar las especificaciones de descripción de archivos como en el desbordamiento al archivo de impresora.</p> <p>(2). Codificar especificaciones de cálculo del RPG para calcular totales de control</p> <p>(3). Codificar las especificaciones *AUTO:</p> <p>(A). Indicadores de salida en las especificaciones de encabezamiento por página.</p> <p>(B). Dos líneas de encabezamiento por página.</p> <p>(C). Utilizar un campo en una especificación de encabezamiento por página *AUTO.</p> <p>(D). Campos y constantes de líneas de totales L1 a L9 (1 a 9 en la posición 39).</p>

IMAGEN 36

Nota: Compare las letras (B) de esta página con las de las páginas siguientes para observar la codificación del generador automático de informes, al objeto de obtener este informe.

2 Entre las sentencias de entrada para el generador automático de informes, pueden incluirse cálculos en el RPG/400. Esta especificación calcula un total de control, sumando el valor de las ventas al valor de las existencias. La ubicación del cálculo en relación con los cálculos producidos por el generador automático de informes, se describe en "Especificaciones generadas" en el tema 12.4.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C 01 SOLDVA ADD VALUE TOTVAL 82
C*
```

3 Especificaciones *AUTO.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OPRINT H A L2 *AUTO
O.....N01N02N03Campo+YBUposPConstante/palabraedición+++*
O OR OFNL2
O 'INFORME DE VENTAS'
O 'PARA UNA COMPAÑÍA'
O H *AUTO
O B 'REGION'
O REGION C
O D 01 *AUTO
O L1 BRANCH 'SUCURSAL'
O ITEMNO 'NUMERO'
O C 'ARTICULO'
```

EJEMPLO 4 - Informe de ventas con totales en columnas cruzadas

```

O          DESC          'DESCRIPCION'
O          SOLDQY        'CANTIDAD'
O          C              'VENTAS'
O          SOLDVA A      'VALOR VENTAS'
O          ONHAND        'EXISTENCIAS'
O          C              'EXISTENCIAS'
O          VALUE A       'VALOR EXISTENCIAS'
O          TOTVAL A      'TOTAL' ---+
O          1              'SUCURSAL' |
O          BRANCH 1      |         | D
O          1              'TOTALES' |
O          2              'REGION' ---+
O          REGION 2
O          2              'TOTALES'
O          R              'TOTALES DE COMPAÑIA'
O*
    
```

- A Los encabezamientos se imprimen en una nueva página cuando cambia el número de región (**L2**) o cuando se produce desbordamiento (**OF**). (Tiene que haberse definido **OF** para el archivo de impresora en las especificaciones de descripción de archivos).
- B Se especifica un segundo encabezamiento de páginas en el generador automático de informes. Como no se especifica espaciado, se produce un espacio después del primer encabezamiento y dos espacios después del segundo. Como no se especifican indicadores de salida, el segundo encabezamiento está condicionado igual que el primero.
- C El contenido del campo **REGION** se imprime en el segundo encabezamiento de página.
- D Pueden imprimirse campos y constantes en las líneas de totales generadas, si se entra el número del nivel de control en la posición 39.

12.7.5 EJEMPLO 5 - Informe de ventas utilizando especificaciones copiadas

IMAGEN 37

- 1 Las especificaciones para el archivo **SALES** están almacenadas en un miembro de archivo fuente.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF....LregLC1AIDePcLaEDispos+.....KSalidaEntradA....U1.*
FSALES IP F 43 DISK A
FPRINT O F 120 PRINTER
```

- A Estas especificaciones podrían sustituirse por una sola sentencia según se muestra en la página siguiente.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
ISALES AA 01
I.....PDesdeA+++DNomcamL1M1FrP1NeCe...*
I 1 7 ITEMNO +-
I 8 9 BRANCH |
I 10 10 REGION |
I 11 25 DESC | A
I 26 270SOLDQY |
I 28 342SOLDVA |
I 35 360ONHAND |
I 37 432VALUE +-
I*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdn01N02N03Nomexc.....*
OPRINT H *AUTO
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O 'INFORME DE VENTAS'
O 'PARA UNA COMPAÑIA'
O D 01 *AUTO
O REGION 'REGION'
O BRANCH 'SUCURSAL'
O ITEMNO 'ARTICULO'
O DESC 'DESCRIPCION'
O SOLDQY 'VENTAS'
O SOLDVA A 'IMPORTE'
O ONHAND 'EXISTENCIAS'
O VALUE A 'VALOR'
O*
```

- 2 Codifique la sentencia **/COPY** para incluir las especificaciones de descripción de archivos y de entrada. (Consulte "Especificaciones de la sentencia **/COPY**" en el tema 12.2 para obtener una descripción detallada de la función de copia.)

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF....LregLC1AIDePcLaEDispos+.....KSalidaEntradA....U1.*
FPRINT O F 120 PRINTER
F*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
O/COPY SALETR B
O*
```

- A La posición 6 de una sentencia **/COPY** no debe contener una **U** ni una **H**.

- B La sentencia **/COPY** copia las especificaciones de descripción de archivos y de entrada para el archivo **SALES** desde el miembro denominado **SALETR**.

EJEMPLO 5 - Informe de ventas utilizando especificaciones copiadas

La sentencia **/COPY** puede aparecer en cualquier lugar entre las especificaciones del generador automático de informes que siguen a la sentencia de opciones del generador automático de informes, y precediendo los registros de entrada de tabla y matriz. Cuando se quieren alterar temporalmente las especificaciones de entrada que se han copiado, tal como ocurre en el "Ejemplo 6", conviene codificar la sentencia **/COPY** en las especificaciones de entrada. Una vez copiadas las especificaciones, se clasifican todas las especificaciones en el orden requerido por el compilador del RPG/400.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OPRINT H *AUTO
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O
O 'INFORME DE VENTAS'
O 'PARA UNA COMPAÑIA'
O D 01 *AUTO
O REGION 'REGION'
O BRANCH 'SUCURSAL'
O ITEMNO 'ARTICULO'
O DESC 'DESCRIPCION'
O SOLDQY 'VENTAS'
O SOLDVA A 'IMPORTE'
O ONHAND 'EXISTENCIAS'
O VALUE A 'VALOR'
O*
```

12.7.6 EJEMPLO 6 - Alteración temporal de especificaciones de entrada copiadas

IMAGEN 38

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
ISALES AA 01
I.....PDesdeA+++DNomcamL1M1FrP1NeCe...*
I          1  7 ITEMNO
I          8  9 BRANCH A
I         10 10 REGION
I         11 25 DESC
I         26 27SOLDQY
I         28 34SOLDVA
I         35 36ONHAND
I         37 43VALUE
I*
```

A Para generar un informe con subtotales por sucursal y región, debe asignarse **L1** a **BRANCH** y **L2** a **REGION** cuando se copian las especificaciones desde el miembro de archivo fuente.

2 y 3 Codifique **/COPY** y las sentencias modificadoras. Como resultado de las sentencias modificadoras, se acumulan tres niveles de totales para los campos **SOLDVA** y **VALUE** (**L1**, **L2** y **LR**).

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF...LregLC1AIDePcLaEDispos+.....KSalidaEntrada...U1.*
FPRINT O F 120 PRINTER
F*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I/COPY SALETR
I.....PDesdeA+++DNomcamL1M1FrP1NeCe...*
I          BRANCHL1 A
I          REGIONL2 B
I*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OPRINT H *AUTO
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O          'INFORME DE VENTAS'
O          'PARA UNA COMPAÑIA'
O          D 01 *AUTO
O          REGION 'REGION'
O          BRANCH 'SUCURSAL'
O          ITEMNO 'ARTICULO'
O          DESC 'DESCRIPCION'
O          SOLDQY 'VENTAS'
O          SOLDVA A 'IMPORTE'
O          ONHAND 'EXISTENCIAS'
O          VALUE A 'VALOR'
O*
```

A Las entradas en las sentencias modificadoras alteran temporalmente las entradas correspondientes de las especificaciones que se han copiado.

B Los nombres de campo, **BRANCH** y **REGION**, identifican las especificaciones de los campos de entrada que tienen que modificarse.

Las especificaciones de entrada o de descripción de archivo salvadas se alteran temporalmente de la siguiente forma (consulte los ejemplos en "Especificaciones de la sentencia **/COPY**" en el

tema 12.2):

- Las entradas en una sentencia modificadora alteran temporalmente las entradas correspondientes en una descripción de archivo o en una descripción de campo de entrada que se hayan copiado.
- Las entradas en blanco en una sentencia modificadora permanecen inalteradas en una sentencia copiada.
- Un ampersand (&) en la posición más a la izquierda de una entrada en la sentencia modificadora establece la entrada a blancos en la especificación copiada.
- Pueden añadirse nuevos campos a las especificaciones de entrada mediante nuevas especificaciones de campos de entrada añadidas como sentencias modificadoras.
- Las sentencias modificadoras no cambian las especificaciones salvadas. La modificación sólo se efectúa en el programa en el que se copian las especificaciones.

13.0 Capítulo 13. Programas de ejemplo RPG/400

Este capítulo contiene una aplicación de ejemplo que consta de una serie de programas RPG II que pueden ejecutarse en el sistema OS/400. Los programas de ejemplo están clasificados de manera que pueda utilizar el manual *RPG Debugging Template*, GX21-9129 para comprobar la codificación de los programas.

Para los programas de ejemplo se ha seleccionado una aplicación de información de tiempos. Esta presentación no pretende proporcionar un sistema de información de tiempos completo, sino que está pensada para ilustrar los programas RPG/400. El capítulo consta de:

- Ambito y objetivos de la aplicación
- Visión general del sistema
- Diseño de la base de datos
- Diseño técnico, que incluye:
 - Mantenimiento del archivo maestro
 - Mantenimiento del archivo de control del área de datos
 - Entrada de transacciones
 - Proceso semanal
 - Proceso mensual
 - Proceso de final de año.

En este manual y en *RPG/400 Reference* se mencionan los programas de ejemplo siguientes.

Nota: Antes de que la aplicación de ejemplo se procese satisfactoriamente, los campos físicos deben contener datos iniciales y debe haberse creado e inicializado el área de datos **CTLFIL**.

Subtemas

- 13.1 Lista de comprobación de los ejemplos del programa
- 13.2 Definición de campo de base de datos
- 13.3 Diseño del menú de información de tiempos
- 13.4 Mantenimiento del archivo maestro
- 13.5 Mantenimiento del archivo de control
- 13.6 Entrada de transacciones del archivo de tiempos
- 13.7 Actualización semanal del archivo de tiempos
- 13.8 Proceso mensual
- 13.9 Proceso de final de año

13.1 Lista de comprobación de los ejemplos del programa

Todas las funciones, códigos de operación, y características RPG/400 que se incluyen en los ejemplos de programa se muestra en Tabla 18. Junto a cada función o código de operación figuran los nombres de programa. Cuando una función se utiliza en más de un programa, se listan todas las apariciones.

Nota: Consulte Tabla 19 para obtener una lista de los programas en el orden en que aparecen en este capítulo.

Tabla 18. Funciones, códigos de operación, y características de los programas de ejemplo RPG/400						
Formulario de especificación	Descripción de función/ código de operación	Programas				
Descripción de archivo	Archivos descritos programa	PRG02	PRG09			
	Archivos descritos externam.	PRG01	PRG03	PRG04	PRG05	
		PRG06	PRG07	PRG08		
	Archivos de disco	PRG01	PRG03	PRG04	PRG05	
		PRG06	PRG07	PRG08	PRG09	
Archivos estación trabajo	Archivos de impresora	PRG01	PRG02	PRG03		
	Archivos de tablas	PRG06	PRG07	PRG08	PRG09	
		PRG09				
Extensión	Matriz	PRG01	PRG02	PRG03	PRG05	
		PRG06	PRG07	PRG08		
	Tabla	PRG09				
Entrada	Descrita por el programa	PRG02	PRG09			
	Descrita externamente	PRG01	PRG03	PRG04	PRG05	
		PRG06	PRG07	PRG08		
	Estructuras de los datos	PRG02	PRG03	PRG05	PRG06	
	Constantes con nombre	PRG07	PRG08	PRG09		
		PRG02	PRG03	PRG09		
Cálculo	Códigos de operación:					
	ADD	PRG03	PRG04	PRG06	PRG07	
		PRG08	PRG09			
	ANDXX	PRG01	PRG02	PRG03	PRG09	
	BEGSR	PRG01	PRG02	PRG03	PRG04	
		PRG06	PRG07	PRG08	PRG09	
	CABXX	PRG01				
	CALL	PRG05				
	CASXX	PRG04				
	CAT	PRG08				
	CHAIN	PRG01	PRG03	PRG06	PRG07	
		PRG08	PRG09			
	CLEAR	PRG08				
	CLOSE	PRG05				
	COMP	PRG02				
	DEFN	PRG05				
	DELET	PRG03				
	DIV	PRG02	PRG06	PRG07		
	DOUXX	PRG04				
	DOWXX	PRG03				
	DSPLY	PRG05				
	ELSE	PRG01	PRG02	PRG03	PRG05	
		PRG06				
	END	PRG01	PRG02	PRG03	PRG04	
		PRG05	PRG06	PRG07	PRG08	
		PRG09				
	ENDSR	PRG01	PRG02	PRG03	PRG04	
		PRG06	PRG07	PRG08	PRG09	
	EXCPT	PRG09				
	EXFMT	PRG01	PRG03			
	EXSR	PRG01	PRG02	PRG03	PRG04	
		PRG06	PRG07	PRG08	PRG09	
	FREE	PRG05				
GOTO	PRG01	PRG03				
IFXX	PRG01	PRG02	PRG03	PRG05		
	PRG06	PRG07	PRG08	PRG09		
IN	PRG05					
KFLD	PRG03					
KLIST	PRG03					
LOKUP	PRG09					
Cálculo	Códigos de operación:					
	MOVE	PRG01	PRG02	PRG03	PRG04	
		PRG05	PRG06	PRG09		
	MOVEL	PRG09				
	MULT	PRG06	PRG07			
MVR	PRG02					

Lista de comprobación de los ejemplos del programa

	OPEN	PRG05				
	ORXX	PRG02	PRG03			
	OUT	PRG05				
	PARM	PRG05				
	PLIST	PRG05				
	READ	PRG04				
	READC	PRG03				
	READE	PRG03				
	REDPE	PRG05				
	RESET	PRG08				
	RETRN	PRG03				
	SCAN	PRG08				
	SETGT	PRG05				
	SETLL	PRG03				
	SETOF	PRG03	PRG09			
	SETON	PRG03	PRG09			
	SUB	PRG07				
	SUBST	PRG08				
	TAG	PRG01	PRG03			
	TIME	PRG06				
	UNLCK	PRG05				
	UPDAT	PRG01	PRG03	PRG04		
	WRITE	PRG01	PRG03	PRG06	PRG07	
		PRG08				
	XFOOT	PRG06	PRG07	PRG08		
	Z-ADD	PRG01	PRG02	PRG03	PRG04	
		PRG06	PRG07	PRG08	PRG09	
	Z-SUB	PRG07				
Salida	Archivos de impresora	PRG06	PRG07	PRG09		
	Descrita por el programa	PRG02	PRG09			
	Descrita externamente	PRG01	PRG03	PRG04	PRG05	
		PRG06	PRG07			
	Salida de excepción	PRG09				
Otras caracte- rísticas que coinciden con el registro	Técnicas de programación estructurada	PRG01	PRG02	PRG03	PRG04	
		PRG05	PRG06	PRG07	PRG08	
		PRG09				
	Interrupciones de nivel	PRG06	PRG09			
	Compatible con SAA	PRG09				
	Teclas de función	PRG01	PRG02	PRG03		
	Proceso de subarchivo	PRG03				
	Indicadores externos	PRG04	PRG05			
	Subrutina de inicialización	PRG08				

La Tabla 19 es una lista de los programas de ejemplo en el orden en que aparecen en este capítulo.

Programa	Consulte
PRG01	Figura 172 en el tema 13.4.9
PRG02	Figura 176 en el tema 13.5.3
PRG03	Figura 181 en el tema 13.6.3
PRG05	Figura 186 en el tema 13.7.1
PRG09	Figura 188 en el tema 13.7.3
PRG06	Figura 194 en el tema 13.8.4
PRG07	Figura 197 en el tema 13.8.7
PRG08	Figura 200 en el tema 13.8.10
PRG04	Figura 201 en el tema 13.8.11

Subtemas

- 13.1.1 Diseño de la base de datos
- 13.1.2 Archivos históricos de transacciones
- 13.1.3 Archivo de control del área de datos
- 13.1.4 Mantenimiento del archivo maestro
- 13.1.5 Mantenimiento del archivo de control del área de datos
- 13.1.6 Entrada del archivo de tiempos
- 13.1.7 Actualización semanal del archivo de tiempos
- 13.1.8 Información y actualización mensual del archivo de entrada de tiempos

13.1.1 *Diseño de la base de datos*

La aplicación de información de tiempos consta de tres archivos maestros, dos archivos históricos de transacciones, y un archivo de control del área de datos. A continuación se lista el diseño de cada uno de los archivos:

Subtemas

- 13.1.1.1 Archivo maestro de empleados
- 13.1.1.2 Archivo maestro de proyectos
- 13.1.1.3 Archivo maestro de códigos de razón

13.1.1.1 Archivo maestro de empleados

El archivo maestro de empleados contiene información sobre los empleados incorporados en el sistema de información de tiempos. Entre los elementos de datos se incluyen:

ACREC Código de registro activo

EMPNO Número de empleado

ENAME Nombre del empleado

EMCAT Categoría del empleado

EDEPT Departamento del empleado

ELOCN Ubicación del empleado

EUSRI USRID del empleado (identificador del usuario)

ENHRS Horario semanal normal del empleado

EPHRC Horas proyectadas para el mes actual del empleado

EPHRY Horas proyectadas para el año a la fecha del empleado

EPHRP Horas proyectadas para el año anterior del empleado

ENHRC Horas no proyectadas para el mes actual del empleado

ENHRY Horas no proyectadas para el año a la fecha del empleado

ENHRP Horas no proyectadas para el año anterior del empleado.

13.1.1.2 Archivo maestro de proyectos

El archivo maestro de proyectos contiene información sobre los proyectos que se utilizan en el sistema de información de tiempos. Entre los elementos de datos se incluyen:

ACREC Código de registro activo
PRCDE Código del proyecto
PRDSC Descripción del proyecto
PRRSP Responsabilidad del proyecto
PRSTR Fecha de comienzo del proyecto
PREND Fecha estimada de finalización del proyecto
PRCMP Fecha de finalización del proyecto
PREST Horas totales estimadas del proyecto
PRHRC Horas proyectadas para el mes actual
PRHRY Horas proyectadas para el año a la fecha
PRHRP Horas proyectadas para el año anterior.

13.1.1.3 Archivo maestro de códigos de razón

El archivo maestro de códigos de razón contiene información sobre las tareas no relacionadas con el proyecto, como días festivos por convenio y días libres por asuntos personales. Entre los elementos de datos se incluyen:

- ACREC** Código de registro activo
- RSCDE** Código de razón
- RSDESC** Descripción del código de razón
- RSHRC** Horas del código de razón para el mes actual
- RSERY** Horas del código de razón para el año a la fecha
- RSERP** Horas del código de razón para el año anterior

13.1.2 Archivos históricos de transacciones

Los archivos históricos de transacciones contienen información de detalle entrada por el usuario en una pantalla de entrada de tiempos. El archivo de transacciones semanales contiene todas las entradas para la semana actual. Cuando se producen informes semanales, este archivo se remonta al archivo mensual. Los dos archivos tiene diseños idénticos. Entre los elementos de datos se incluyen:

ACREC Código de registro activo

EMPNO Número de empleado

EUSRI USRID del empleado (identificación del usuario)

ACDAT Fecha real trabajada (opcional)

CWKDT Fecha de final de semana

CMTDT Fecha de final de mes

PRCDE Código del proyecto

RSCDE Código de razón

EHWRK Horas trabajadas

TFRRN Número relativo de registro del archivo de transacciones.

13.1.3 Archivo de control del área de datos

Este archivo de control del área de datos contiene información de control para el sistema de información de tiempos. Entre los elementos de datos se incluyen:

ACREC Código de registro activo

CWKDT Fecha de final de semana

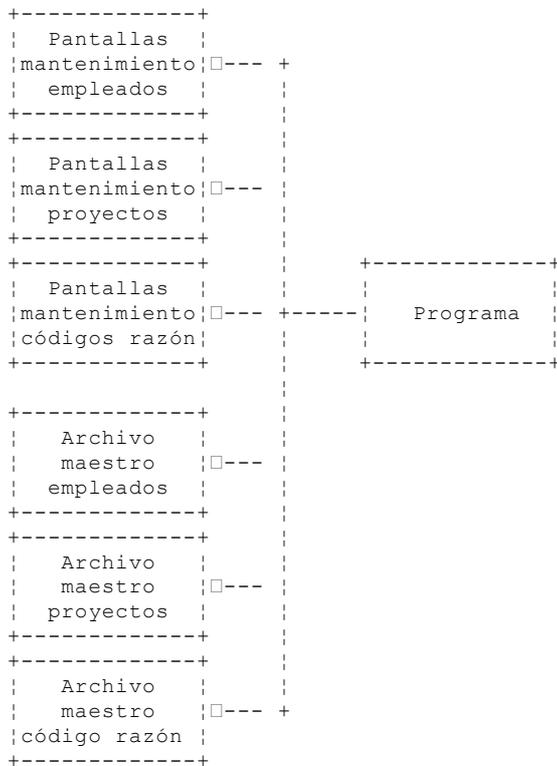
CMTDT Fecha de final de mes

CALLE Todas las entradas convertidas en indicador.

13.1.4 Mantenimiento del archivo maestro

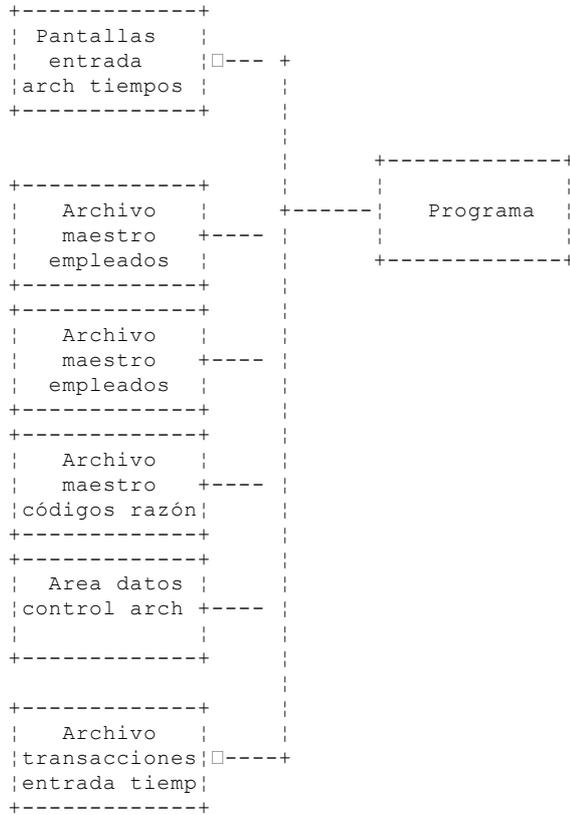
Todos los archivos maestros se mantienen mediante la utilización de los programas de las estaciones de trabajo. Todas las pantallas están diseñadas utilizando la Ayuda para el diseño de pantallas (SDA) y se definen externamente. A continuación aparecen los diagramas de flujo para el proceso de mantenimiento del archivo maestro:

El proceso de mantenimiento del archivo maestro permite adiciones, cambios y supresiones en el archivo maestro de empleados, en el archivo maestro de proyectos y en el archivo maestro de códigos de razón.



13.1.6 Entrada del archivo de tiempos

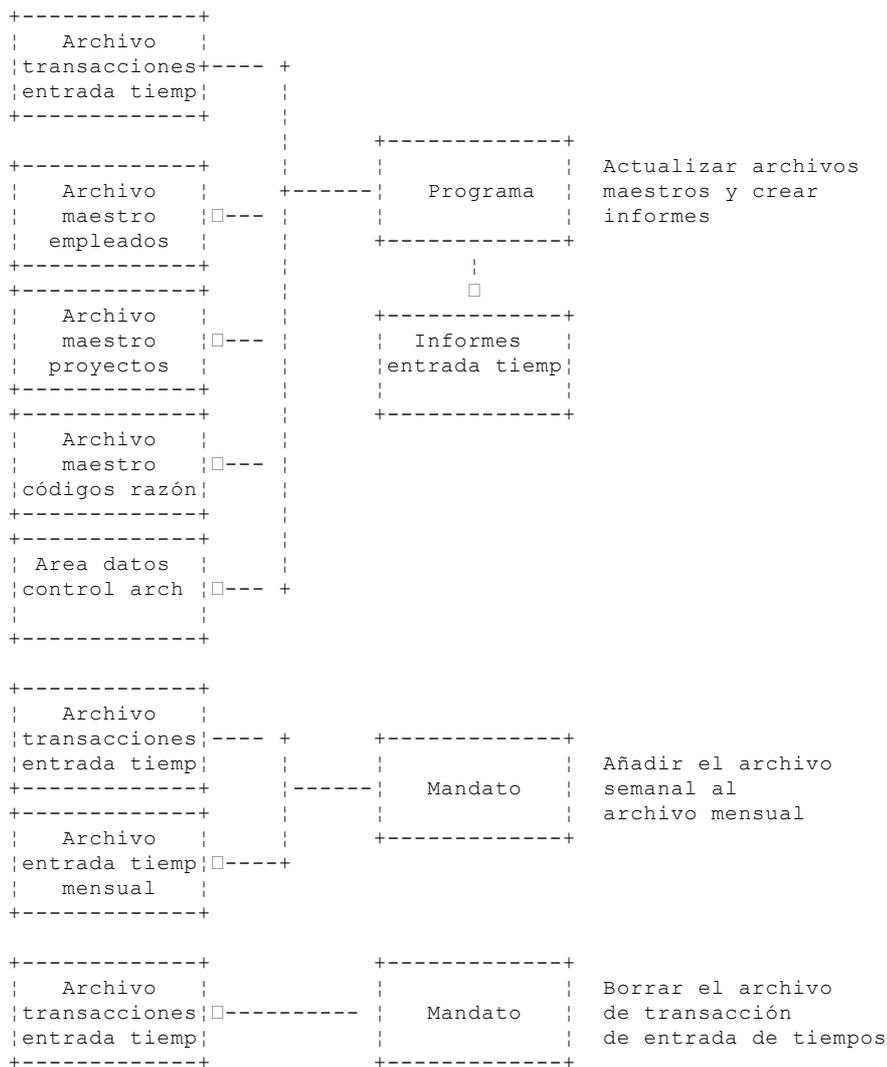
El proceso de entrada del archivo de tiempos se efectúa utilizando el proceso de subarchivo de la estación de trabajo. Las pantallas están diseñadas mediante la utilización de la SDA y se definen externamente. En los archivos maestros se realiza la verificación de los datos entrados. El proceso de entrada del archivo de tiempos permite adiciones, cambios y supresiones en el archivo de transacciones con todos los campos susceptibles de ser mantenidos. El archivo de entrada de datos se utiliza para el informe semanal y para el proceso de actualización del archivo. A continuación aparece el diagrama de flujo de la entrada del archivo de tiempos:



13.1.7 Actualización semanal del archivo de tiempos

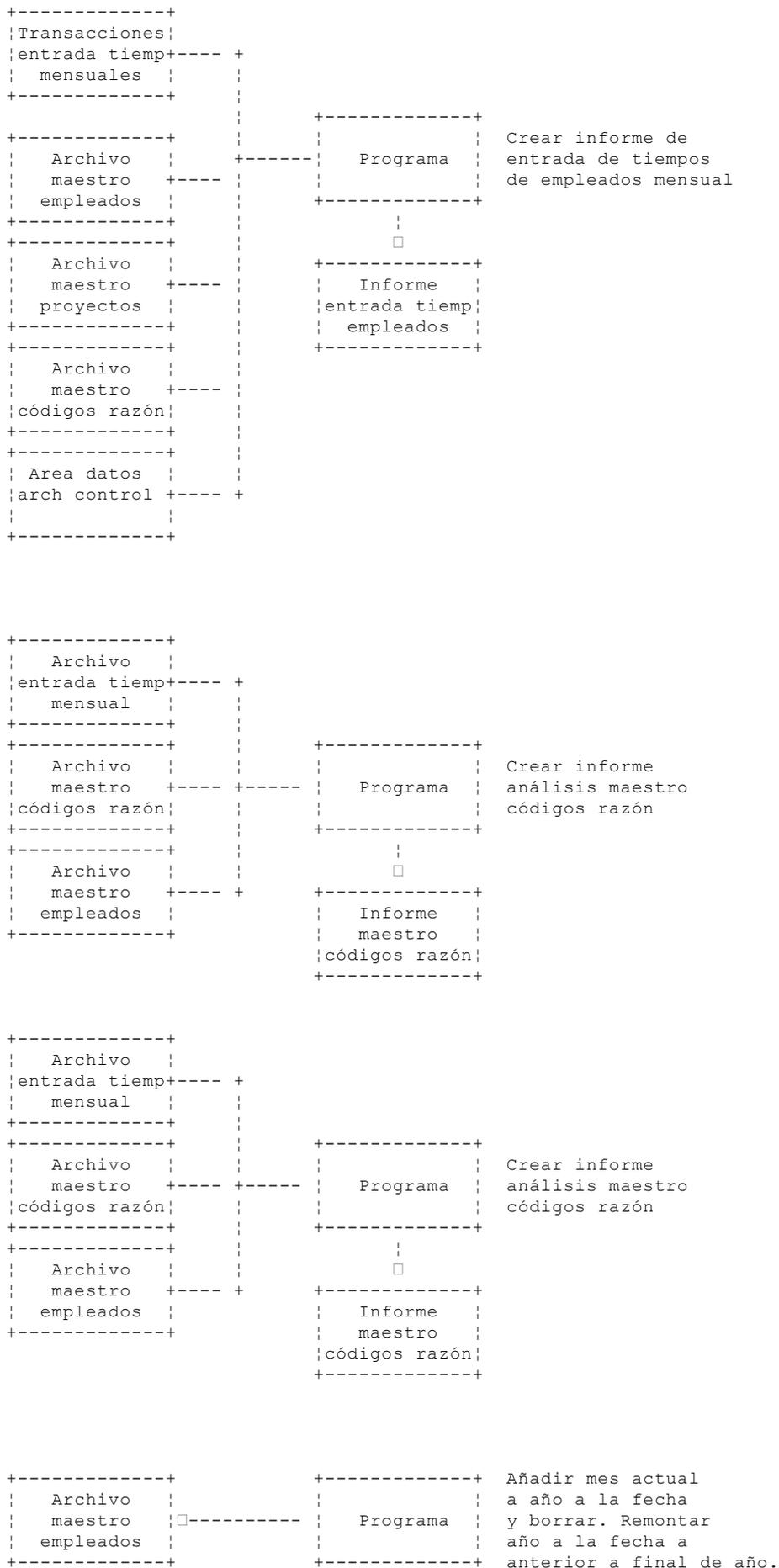
El archivo de transacciones de entrada de tiempos se procesa semanalmente para producir informes de hojas de trabajo de tiempos y para actualizar los archivos maestros con las horas de entrada de tiempos. El proceso de actualización semanal del archivo de tiempos determina si se han efectuado o no todas las entradas de tiempo necesarias. Si faltan entradas, se notifica al empleado/a que faltan sus entradas de tiempos y también se notifica a la persona que solicita la actualización. Esta persona puede cancelar la actualización o continuar. Después de haber efectuado todas las entradas o de que la persona que solicitó la actualización haya optado por continuar, se producen los informes y los archivos se actualiza. El archivo de transacciones semanal se añade al archivo mensual y a continuación se borra. El diagrama de flujo siguiente ilustra este proceso.

IMAGEN 39



13.1.8 Información y actualización mensual del archivo de entrada de tiempos

Después del último proceso semanal del mes, se procesa el archivo mensual de transacciones de entrada de tiempos para producir informes de final de mes y para actualizar los archivos maestros para la preparación de nuevos datos mensuales. El diagrama de flujo siguiente ilustra este proceso.



Información y actualización mensual del archivo de entrada de tiempos

<pre> +-----+ Archivo maestro <input type="checkbox"/>----- proyectos +-----+ </pre>	<pre> +-----+ Programa +-----+ </pre>	<pre> Añadir mes actual a año a la fecha y borrar. Remontar año a la fecha a anterior a final de año. </pre>
<pre> +-----+ Archivo maestro <input type="checkbox"/>----- códigos razón +-----+ </pre>	<pre> +-----+ Programa +-----+ </pre>	<pre> Añadir mes actual a año a la fecha y borrar. Remontar año a la fecha a anterior a final de año. </pre>
<pre> +-----+ Transacciones entrada tiempo <input type="checkbox"/>----- mensuales +-----+ </pre>	<pre> +-----+ Mandato CLRPFM +-----+ </pre>	<pre> Borrar archivo transacciones entrada tiempos mensual </pre>

13.2 Definición de campo de base de datos

Esta sección contiene la definición del campo de base de datos y los atributos de campo del sistema de información de tiempos. Se ha creado un archivo de referencia de base de datos, REF MST, que contiene todas las definiciones detalladas de los campos de todos los archivos. Los campos de cada archivo podrían haberse definido como parte de sus propias especificaciones de descripción de datos (DDS), no obstante, cuando un campo se utiliza en más de un archivo, el campo acaba por estar definido varias veces. Si es necesario efectuar un cambio en la definición de dicho campo, deberá repetirse la misma operación para cada aparición. Mediante la definición de un archivo de referencia de campos, podemos eliminar múltiples definiciones y simplificar la tarea de volver a definir el campo en el futuro. Vea la Figura 154 en el tema 13.2.1.

Subtemas

- 13.2.1 Archivo maestro de referencias de la base de datos - REF MST
- 13.2.2 Archivo de control del área de datos - CTLFIL
- 13.2.3 Archivo maestro de empleados - EMP MST
- 13.2.4 Archivo maestro de proyectos - PRJ MST
- 13.2.5 Archivo maestro de códigos de razón - RSN MST
- 13.2.6 Archivo de entrada de transacciones semanal - TR WEEK
- 13.2.7 Archivo de entrada de transacciones mensual - TR MNTH

13.2.1 Archivo maestro de referencias de la base de datos - REFMST

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A* REFMST - Archivo maestro de referencias de la base de datos
A* DESCRIPCION - Un archivo que contiene información a nivel de
A* campo para los archivos del sistema de
A* información de tiempos. Se hace referencia a esta
A* información a nivel de campo cuando se crean los
A* archivos físicos y lógicos específicos. La SDA también
A* hace referencia a esta información a nivel de campo
A* cuando se crean los formatos de pantalla.
A*****
A.....T.Nombre+++RLon++TdPB.....Funciones+++++++
A R RCREF TEXT('REFERENCIA BASE DATOS')
A* CAMPOS COMUNES
A ACREC 1 TEXT('CODIGO REGISTRO ACTIVO')
A COLHDG('REGISTRO ACTIVO')
A DATFL 6 0 TEXT('CAMPO DATOS')
A COLHDG('FECHA' 'MMDDAA')
A* CAMPOS RELACIONADOS MAESTRO EMPLEADOS
A EMPNO 6 0 TEXT('NUMERO EMPLEADO')
A COLHDG('EMPLEADO' 'NUMERO')
A ENAME 30 TEXT('NOMBRE EMPLEADO')
A COLHDG('EMPLEADO' 'NOMBRE')
A EMCAT 1 TEXT('CATEGORIA EMPLEADO')
A COLHDG('EMPLEADO' 'CATEGORIA')
A EDEPT 5 TEXT('DEPARTAMENTO EMPLEADO')
A COLHDG('EMPLEADO' 'DEPARTAMENTO')
A ELOCN 30 TEXT('UBICACION EMPLEADO')
A COLHDG('EMPLEADO' 'UBICACION')
A EUSRI 8 TEXT('ID USUARIO EMPLEADO')
A COLHDG('EMPLEADO' 'ID USUARIO')
A ENHRS 3 1 TEXT('HORAS SEMANA NORMAL EMPLEADO')
A COLHDG('NORMAL' 'HORAS SEMANA')
A EPHRC 5 1 TEXT('HORAS PROYECTO MES ACTUAL')
A COLHDG('HORAS PROYECTO' 'MES ACTUAL')
A EPNRC 5 1 TEXT('HORAS NO PROYECTO MES ACTUAL')
A COLHDG('HORAS NO PROYECTO' 'MES ACTUAL')

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A.....T.Nombre+++RLon++TdPB.....Funciones+++++++
A EPHRY 7 1 TEXT('HORAS PROYECTO AÑO A LA FECHA')
A COLHDG('HORAS PROYECTO' 'AÑO A LA FECHA')
A EPHRP 7 1 TEXT('HORAS PROYECTO AÑO ANTERIOR')
A COLHDG('HORAS PROYECTO' 'AÑO ANTERIOR')
A EPNRY 7 1 TEXT('HORAS NO PROYECTO AÑO A LA FECHA')
A COLHDG('NO PROYECTO' 'HORAS AÑO A LA FECHA')
A EPNRP 7 1 TEXT('HORAS NO PROYECTO AÑO ANTERIOR')
A COLHDG('HORAS NO PROYECTO' 'AÑO ANTERIOR')
A EHWRK 5 1 TEXT('HORAS TRABAJADAS EMPLEADO')
A COLHDG('HORAS EMPLEADO' 'TRABAJADAS')
A* CAMPOS RELACIONADOS MAESTRO PROYECTOS
A PRCDE 8 TEXT('CODIGO PROYECTO')
A COLHDG('PROYECTO' 'CODIGO')
A PRDSC 50 TEXT('DESCRIPCION PROYECTO')
A COLHDG('PROYECTO' 'DESCRIPCION')
A PRRSP 30 TEXT('RESPONSABILIDAD PROYECTO')
A COLHDG('PROYECTO' 'RESPONSABILIDAD')
A PRSTR R REFFLD(DATFL)
A TEXT('FECHA INICIO PROYECTO')
A COLHDG('INICIO PROYECTO' 'FECHA')
A PREND R REFFLD(DATFL)
A TEXT('FECHA ESTIMADA FIN PROYECTO')
A COLHDG('FECHA ESTIMADA' 'FIN PROYECTO')
A PRCMP R REFFLD(DATFL)
A TEXT('FECHA TERMINACION PROYECTO')
A COLHDG('TERMINACION PROYECTO' 'FECHA')
A PREST 9 1 TEXT('TOTAL HORAS ESTIMADAS PROYECTO')
A COLHDG('TOTAL HORAS' 'ESTIMADAS PROYECTO')
A PRHRC 7 1 TEXT('HORAS PROYECTO MES ACTUAL')

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A.....T.Nombre+++RLon++TdPB.....Funciones+++++++
A PRHRY 9 1 COLHDG('HORAS PROYECTO' 'MES ACTUAL')
A TEXT('HORAS PROYECTO AÑO A LA FECHA')
A PRHRP 9 1 COLHDG('HORAS PROYECTO' 'AÑO A LA FECHA')
A TEXT('HORAS PROYECTO AÑO ANTERIOR')
A COLHDG('HORAS PROYECTO' 'AÑO ANTERIOR')
A* CAMPOS RELACIONADOS MAESTRO CODIGOS RAZON
A RSCDE 8 TEXT('CODIGO RAZON')
A COLHDG('CODIGO' 'RAZON')
A RSDSC 50 TEXT('DESCRIPCION CODIGO RAZON')

```

Archivo maestro de referencias de la base de datos - REFMST

A				COLHDG('CODIGO RAZON' 'DESCRIPCION')
A	RSHRC	7	1	TEXT('CODIGO RAZON HORAS MES ACTUAL')
A				COLHDG('HORAS CODIGO RAZON' 'MES ACTUAL')
A	RSHRY	9	1	TEXT('CODIGO RAZON HORAS AÑO A LA FECHA')
A				COLHDG('CODIGO RAZON' 'HORAS AÑO A FECHA')
A	RSHRP	9	1	TEXT('CODIGO RAZON HORAS AÑO ANTERIOR')
A				COLHDG('CODIGO RAZON HORAS' 'AÑO ANTERIOR')
A*	CAMPOS RELACIONADOS	ARCHIVO	CONTROL	
A	CTCDE	6		TEXT('CODIGO REGISTRO CONTROL')
A				COLHDG('REGISTRO CONTROL' 'CODIGO')
A	CWKDT	6S	0	TEXT('FECHA FINAL SEMANA')
A				COLHDG('FINAL SEMANA' 'FECHA')
A	CMTDT	6S	0	TEXT('FECHA FINAL MES')
A				COLHDG('FINAL MES' 'FECHA')
A	CALLE	1		TEXT('INDICADOR HECHAS TODAS ENTRADAS')
A				COLHDG('ENTRADAS' 'INDICADOR')

Figura 154. Archivo maestro de referencias de la base de datos

13.2.2 Archivo de control del área de datos - CTLFIL

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A*   CTLFIL - Archivo de control del área de datos
A*   DESCRIPCION - Un archivo de control de datos que contiene
A*                   información a nivel de control para el sistema de
A*                   información de tiempos. El área de datos contiene
A*                   un formato de registro.
A*****
A.....T.Nombre+++RLon++TDpB.....Funciones+++++++*
A
A           R RCCTL           REF (REFMST)
A           CTCDE           TEXT ('ARCHIVO CONTROL')
A           CWKDT           R
A           CMTDT           R
A           CALLE           R
A           K CTCDE
```

Figura 155. Archivo de control del área de datos

13.2.3 Archivo maestro de empleados - EMPMST

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A*   EMPMST - Archivo maestro de empleados
A*   DESCRIPCION - Un archivo que contiene un registro para cada
A*                   empleado incorporado al sistema de información de
A*                   tiempos. Se lleva a cabo el mantenimiento de la
A*                   actividad del mes actual, año a la fecha, proyecto
A*                   año anterior, y no relacionada con el proyecto.
A*****
A.....T.Nombre+++RLon++TDpB.....Funciones+++++*****
A                                     UNIQUE
A                                     REF(REFMST)
A      R RCEMP                        TEXT('MAESTRO EMPLEADOS')
A      ACREC                          R
A      EMPNO                          R
A      ENAME                          R
A      EMCAT                          R
A      EDEPT                          R
A      ELOCN                          R
A      EUSRI                          R
A      ENHRS                          R
A      EPHRC                          R
A      EPHRY                          R
A      EPHRP                          R
A      EPNRC                          R
A      EPNRY                          R
A      EPNRP                          R
A      K EMPNO
```

Figura 156. Archivo maestro de empleados

13.2.4 Archivo maestro de proyectos - PRJMST

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A*   PRJMST - Archivo maestro de proyectos
A*   DESCRIPCION - Un archivo que contiene información relacionada
A*                   con la actividad del proyecto. Se lleva a cabo el
A*                   mantenimiento de la actividad del mes actual, año
A*                   a la fecha, y del año anterior. Existe un registro
A*                   para cada código de proyecto.
A*****
A.....T.Nombre++++RLon++TDpB.....Funciones+++++++*
A                                     UNIQUE
A                                     REF (REFMST)
A      R RCPRJ                        TEXT ('MAESTRO PROYECTOS')
A      ACREC                          R
A      PCDE                            R
A      PRDSC                           R
A      PRRSP                           R
A      PRSTR                            R
A      PREND                           R
A      PRCMP                           R
A      PREST                           R
A      PRHRC                           R
A      PRHRY                           R
A      PRHRP                           R
A      K PCDE
  
```

Figura 157. Archivo maestro de proyectos

13.2.6 Archivo de entrada de transacciones semanal - TRWEEK

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A*   TRWEEK - Archivo de entrada de transacciones semanal
A*   DESCRIPCION - Un archivo que contiene todas las entradas
A*                   realizadas al sistema de información de tiempos
A*                   para la semana.
A*****
A.....T.Nombre++++RLon++TDpB.....Funciones+++++++*
A
A           R RCWEEK                REF (REFMST)
A           ACREC                    TEXT ('ENTRADA TRANSACCIONES SEMANAL')
A           EMPNO                     R
A           EUSRI                     R
A           ACDAT                      6S 0
A           CWKDT                     R
A           CMTDT                     R
A           PRCDE                     R
A           RSCDE                     R
A           EHWRK                     R
A           TFERRN                    3 0

A*****
A*   TRWEEKL - Visión lógica del archivo de entrada de transacciones
A*   DESCRIPCION - El programa de entrada de transacciones utiliza
A*                   este archivo para permitir volver a visualizar las
A*                   entradas de empleados existentes y la actualización
A*                   de los registro añadidos o cambiados en la entrada
A*                   de subarchivos.
A*****
A.....T.Nombre++++.Lon++TDpB.....Funciones+++++++*
A           R RCWEEK                PFILE (TRWEEK)
A           K EMPNO
A           K TFERRN
    
```

Figura 159. Archivo de entrada de transacciones semanal

13.2.7 Archivo de entrada de transacciones mensual - TRMNTH

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A*   TRMNTH - Archivo de entrada de transacciones mensual
A*   DESCRIPCION - Un archivo que contiene todas las entradas
A*                   realizadas al sistema de información de tiempos
A*                   para el mes.
A*****
A.....T.Nombre++++RLon++TDpB.....Funciones+++++*****
A
A       R RCMNTH                REF (REFMST)
A       ACREC                   TEXT ('ENTRADA TRANSACCIONES MENSUAL')
A       EMPNO                   R
A       EUSRI                   R
A       CWKDT                   R
A       CMTDT                   R
A       PRCDE                   R
A       RSCDE                   R
A       EHWRK                   R

```

```

A*****
A*   TRMNTHL - Visión lógica del archivo de entrada de transacciones
A*           mensual
A*   DESCRIPCION - Este archivo lo utiliza el sistema de información
A*           mensual de empleados de entrada de tiempos.
A*****
A.....T.Nombre++++.Lon++TDpB.....Funciones+++++*****
A       R RCMNTH                PFILE (TRMNTH)
A       K CWKDT
A       K EMPNO

```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A*   TRMNTHR - Visión lógica del archivo de entrada de transacciones
A*           mensual
A*   DESCRIPCION - Este archivo lo utiliza el sistema de información
A*           mensual del proyecto de entrada de tiempos.
A*****
A.....T.Nombre++++.Lon++TDpB.....Funciones+++++*****
A       R RCMNTH                PFILE (TRMNTH)
A       K PRCDE
A       K CWKDT
A       K EMPNO
A       O PRCDE                COMP (EQ '      ')

```

```

A*****
A*   TRMNTHN - Visión lógica del archivo de entrada de transacciones
A*           mensual
A*   DESCRIPCION - Este archivo lo utiliza el sistema de información
A*           mensual de códigos de razón de entrada de tiempos.
A*****
A       R RCMNTH                PFILE (TRMNTH)
A       K RSCDE
A       K CWKDT
A       K EMPNO
A       O RSCDE                COMP (EQ '      ')

```

Figura 160. Archivo de entrada de transacciones mensual

13.3 Diseño del menú de información de tiempos

La Figura 161 muestra el Menú principal del sistema de información de tiempos. El Menú principal le permite realizar el mantenimiento del archivo, el mantenimiento del archivo de control, la entrada de transacciones, la actualización semanal y la actualización mensual. Vea en la Figura 162 las DDS para el menú TMENU.

En este capítulo se describe detalladamente cada opción del menú. El Menú principal se repite en cada opción, y la opción que se describe aparece resaltada. En la explicación de cada opción se incluye el programa a nivel del control que se llama, el programa RPG/400 que se llama, o el mandato que se procesa.

```

-----+-----
|
| TMENU                               Sistema de Información de Tiempos
|                                     Menú Principal
|
| 1. Mantenimiento archivo maestro      (PRG01)
| 2. Mantenimiento archivo control      (PRG02)
| 3. Entrada transacciones archivo tiempos (PRG03)
| 4. Actualización semanal archivo tiempos (PROC1)
| 5. Actualización e información mensual archivo tiempos (PROC3)
|
| 8. Visualizar mensajes                 (DSPMSG)
| 9. Fin de sesión                       (SIGNOFF)
|
| Selección o mandato
| ==>
| F3=Salir F4=Solicit F9=Recuperar F12=Cancelar
| F13=Soporte usuario F16=Menú principal del sistema
|
|-----+-----

```

Figura 161. Diseño del menú principal del sistema de información de tiempos.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A* TMENU - Descripción de datos del menú principal del sistema de
A* información de tiempos
A* DESCRIPCION - Un archivo de pantalla que describe los formatos
A* que el programa utiliza para permitir el
A* mantenimiento de la estación de trabajo del sistema
A* de información de tiempos.
A*****
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++
A DSPSIZ (24 80 *DS3)
A 27 132 *DS4)
A CHGINPDFT
A INDARA
A PRINT (*LIBL/QSYSPRT)
A R TMENU
A DSPMOD (*DS3)
A LOCK
A SLNO (01)
A CLRL (*ALL)
A ALWROL
A CF03
A HELP
A HOME
A HLPRTN
A 1 2 'TMENU'
A COLOR (BLU)

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++
A 019 2 'Selección o mandato'
A
A 5 7'1. Mantenimiento archivo maestro'
A 6 7'2. Mantenimiento archivo control'
A 7 7'3. Entrada transacciones archivo tiempos'

```

Diseño del menú de información de tiempos

```

A           8 7'4.  Actualización semanal archivo tiempos'
A           9 7'5.  Actualización e información del
A             archivo mensual de tiempos
A          12 7'8.  Visualizar mensajes'
A          13 7'9.  Fin de sesión'
A           1 28'Sistema de información de tiempos'
A           2 34'Menú principal'
A           5 63'(PRG01)'
A           6 63'(PRG02)'
A           7 63'(PRG03)'
A           8 63'(PROC1)'
A           9 63'(PROC3)'
A          12 63'(DSPMSG)'
A          13 63'(SIGNOFF)'
```

```

* ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
TMENUQQ,1
0001 CALL PGM(PRG01)
0002 CALL PGM(PRG02)
0003 CALL PGM(PRG03)
0004 CALL PGM(PROC1)
0005 CALL PGM(PROC3)
0008 DSPMSG
0009 SIGNOFF
```

Figura 162. Especificaciones de descripción de datos TMENU

Nota: El menú TMENUQQ, 1 parte de las DDS anteriores, empieza en la columna 1.

13.4 Mantenimiento del archivo maestro

Seleccione la opción **1** (Mantenimiento del archivo maestro) del Menú Principal del Sistema de Información de Tiempos para efectuar adiciones, cambios, o supresiones en el archivo maestro de empleados, en el archivo maestro de proyectos, o en el archivo maestro de códigos de razón. Realice estos cambios en el archivo maestro antes de realiza sus transacciones de entrada de tiempos. El proceso de entrada de tiempos verifica los datos que ha entrado para estos tres archivos maestros. La Figura 163 muestra el Menú Principal del Sistema de Información de Tiempos. La opción **1** llama al programa PRG01 utilizando el mandato CALL PGM(PRG01).

```

+-----+
| TMENU                               Sistema de Información de Tiempos |
|                                     Menú Principal                   |
|                                                                              |
| 1. Mantenimiento archivo maestro (PRG01)                          |
| 2. Mantenimiento archivo control (PRG02)                             |
| 3. Entrada transacciones archivo tiempos (PRG03)                     |
| 4. Actualización semanal archivo tiempos (PROC1)                     |
| 5. Actualización e información mensual archivo tiempos (PROC3)       |
|                                                                              |
| 8. Visualizar mensajes (DSPMSG)                                       |
| 9. Fin de sesión (SIGNOFF)                                             |
|                                                                              |
| Selección o mandato                                                    |
| ===>                                                                    |
| F3=Salir F4=Solicit F9=Recuperar F12=Cancelar                         |
| F13=Soporte usuario F16=Menú principal del sistema                    |
+-----+
    
```

Figura 163. Menú Principal del Sistema de Información de Tiempos

Subtemas

- 13.4.1 Pantalla Mantenimiento del Archivo Maestro - PRG01FM
- 13.4.2 Selección del maestro de empleados - Formato EMPSEL
- 13.4.3 Mantenimiento del maestro de empleados - Formato EMPMNT
- 13.4.4 Selección del maestro de proyectos - Formato PRJSEL
- 13.4.5 Mantenimiento del maestro de proyectos - Formato PRJMNT
- 13.4.6 Selección del maestro de códigos de razón - Formato RSNSEL
- 13.4.7 Mantenimiento del maestro de códigos de razón - Formato RSNMNT
- 13.4.8 Descripciones de datos de mantenimiento del archivo maestro - PRG01FM
- 13.4.9 Programa en RPG/400 de mantenimiento de archivos maestros - PRG01

13.4.1 Pantalla Mantenimiento del Archivo Maestro - PRG01FM

El formato inicial, en la aplicación de mantenimiento de información de tiempos, le permite seleccionar el archivo maestro que desea mantener. La Figura 164 en el tema 13.4.1.1 muestra la pantalla para el formato de Selección de Mantenimiento. Cada aplicación de mantenimiento del archivo maestro consta de dos formatos (vea la Figura 165 en el tema 13.4.2 a Figura 170 en el tema 13.4.7).

Subtemas

13.4.1.1 Formato SELECT - Selección de mantenimiento

13.4.3 Mantenimiento del maestro de empleados - Formato EMPMNT

La Figura 166 muestra la pantalla para la entrada del Mantenimiento del Maestro de Empleados.

```

+-----+
| PRG01          Sistema de Información de Tiempos          MM/DD/AA |
|                Mantenimiento del Maestro de Empleados      TT:TT:TT |
|                                                         |
|      Número 666666 Nombre  BBBB BBBB BBBB BBBB BBBB BBBB BBBB 1 |
|      Categoría          B                                     2 |
|      Departamento      BBBB                                 3 |
|      Ubicación         BBBB BBBB BBBB BBBB BBBB BBBB BBBB 4 |
|      USRID             BBBB BBBB                           5 |
|      Horas semana norm 999-                                6 |
|                                                         |
|                    Historia de la Información de Tiempos |
|                                                         |
|                               Mes      Año a      Año |
|                               actual  la fecha anterior |
|      Relacionado proyecto 66666 6666666 6666666 7 |
|      No relac proyecto   66666 6666666 6666666 |
|                                                         |
|      000000000000000000000000000000000000000000000000000 8 |
|                                                         |
|      F3-Fin de trabajo F4-Selección de mantenimiento F5-Selección de empleado |
+-----+

```

Figura 166. Mantenimiento del Maestro de Empleados - Formato EMPMNT

Los valores actuales del maestro de empleados se visualizan en una petición de cambio o en una petición de supresión. En una petición de adición todos los campos están en blanco. En una petición de supresión, todos los campos están protegidos. Puede teclear la información de nombre de empleado 1 , categoría 2 , departamento 3 , ubicación 4 , identificador de usuario 5 , y horas semanales normales 6 . La historia de la información de tiempos 7 no puede mantenerse.

Cuando pulsa Intro, el archivo maestro de empleados se actualiza y aparece el formato de selección de mantenimiento de empleados **EMPSEL**. Puede pulsar F3 para finalizar el trabajo, F4 para volver al formato de selección de mantenimiento **SELECT** o F5 para volver al formato de selección de mantenimiento de empleados **EMPSEL**. El programa no actualiza el archivo maestro de empleados cuando utiliza estas teclas de función. El programa no valida los datos entrados en este formato. Si se solicita una **Adición** y el registro del maestro de empleados se ha suprimido anteriormente, el programa visualiza un mensaje de aviso de 8 .

Los mensajes de error posibles son:

Aviso - El registro se suprimió anteriormente.

13.4.4 Selección del maestro de proyectos - Formato PRJSEL

La Figura 167 muestra la pantalla para la selección del Mantenimiento del Maestro de Proyectos.

```

PRG01                Sistema de Información de Tiempos          MM/DD/AA
                    Mantenimiento del Maestro de Proyectos      TT:TT:TT

                    Código proyecto  BBBB BBBB                1
                    Código acción   B   A-Añadir              2
                                       C-Cambiar
                                       D-Suprimir

                                3
                    0000000000000000000000000000000000000000

                    F3-Fin de trabajo F4-Selección de mantenimiento
    
```

Figura 167. Selección del maestro de proyectos - Formato PRJSEL

Debe entrar un código de proyecto de 1 y un código de acción de 2. Cuando pulsa Intro, el programa verifica la combinación de código de proyecto y código de acción. Si no se encuentran errores, aparece el formato de entrada de mantenimiento del maestro de proyectos **PRJMNT** (vea la Figura 168 en el tema 13.4.5). Si se encuentran errores, el programa vuelve al formato **PRJSEL** con un mensaje de error de 3. Puede pulsar F3 para finalizar el trabajo, o F4 para volver al formato de selección de mantenimiento **SELECT**.

Los mensajes de error posibles son:

- El código de acción no es igual a A, C o D.
- Adición solicitada pero el registro ya existe en el archivo.
- Cambio solicitado pero el registro no existe.
- Cambio solicitado pero el registro está marcado para supresión.
- Supresión solicitada pero el registro no existe.
- Supresión solicitada pero el registro ya existe.

13.4.5 Mantenimiento del maestro de proyectos - Formato PRJMNT

La Figura 168 muestra la pantalla para la entrada del Mantenimiento del Maestro de Proyectos.

```

+-----+
| PRG01                Sistema de Información de Tiempos          MM/DD/AA
|                    Mantenimiento del Maestro de Proyectos      TT:TT:TT
|
|                    Código proyecto 00000000                    1
| Descripción          BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
|
| Responsabilidad      BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB    2
| Fecha inicio proyecto 999999-                                3
| Fecha estimada final proy 999999-                            4
| Fecha finalización proyec 999999-                            5
| Horas estimadas proyecto 999999999-                          6
|
|                    Historia proyecto
|                    Mes      Año a la      Año
|                    actual   fecha        anterior
|                    66666666 6666666666 6666666666          7
|
|
|                    0000000000000000000000000000000000000000  8
|
| F3-Fin de trabajo F4-Selección mantenimiento F6-Selección código proyecto
+-----+

```

Figura 168. Mantenimiento del maestro de proyectos - Formato PRJMNT

Los valores actuales del maestro de proyectos se visualizan en una petición de cambio o en una petición de supresión. En una petición de adición todos los campos están en blanco. En una petición de supresión, todos los campos están protegidos. Puede teclear la información de descripción de proyecto 1 , de responsabilidad 2 , de fecha de inicio 3 , de fecha de finalización estimada 4 , de fecha de finalización 5 y de horas estimadas 6 . La historia de la información de tiempos 7 no puede mantenerse.

Cuando pulsa Intro, el archivo maestro de proyectos se actualiza y aparece el formato de selección de mantenimiento de proyectos **PRJSEL**. Puede pulsar F3 para finalizar el trabajo, o F4 para volver al formato de selección de mantenimiento **SELECT**, o F6 para volver al formato de selección de mantenimiento de proyectos **PRJSEL**. El programa no actualiza el archivo maestro de proyectos cuando utiliza estas teclas de función. El programa no valida los datos entrados en este formato. Si se solicita una **Adición** y el registro del maestro de proyectos se ha suprimido anteriormente, el programa visualiza un mensaje de aviso de 8 .

Los mensajes de error posibles son:

Aviso - El registro se suprimió anteriormente.

13.4.8 Descripciones de datos de mantenimiento del archivo maestro - PRG01FM

La Figura 171 muestra las DDS para el archivo de pantalla Mantenimiento del Archivo Maestro PRG01FM. Existen siete formatos de registro, que se identifican por una **R** en la posición 17 seguida del nombre de formato en las posiciones 19 a 28. Se han utilizado las palabras clave siguientes:

- ALARM** Activa la alarma audible.
- BLINK** Hace parpadear el cursor.
- CAAnn** Hace que la tecla de función especificada en la palabra clave esté disponible para su utilización.
- DATE** Visualiza la fecha del trabajo actual como una constante.
- DSPATR** Especifica un atributo de pantalla para el campo.
- DSPSIZ** Especifica el tamaño de la pantalla en la que el programa puede abrir este archivo.
- EDTCDE** Especifica la edición en un campo numérico con posibilidad de salida.
- INDARA** Elimina los indicadores de respuesta y de opción del almacenamiento intermedio y los sitúa en un área de indicadores de 99 bytes separada.
- REFFLD** Hace referencia a los atributos de un campo definido anteriormente.
- TIME** Visualiza la hora actual del sistema como una constante.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A* PRG01FM - Descripciones datos mantenimiento archivo maestro
A* DESCRIPCION - Un archivo de pantalla que describe los formatos
A* que el programa PRG01 RPG/400 utiliza para permitir
A* el mantenimiento de la estación de trabajo de los
A* archivos maestros de información de tiempos:
A* EMPMST - Archivo maestro empleados
A* PRJMST - Archivo maestro proyectos
A* RSNMST - Archivo maestro códigos razón
A*****
```

```
A* El código siguiente contiene las palabras clave que describen el
A* archivo de pantalla en su conjunto.
```

```
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++
```

```
A DSPSIZ(24 80 *DS3)
A PRINT
A INDARA
A CA03(03 'fin de trabajo')
A CA04(04 'volver a selección
A de mantenimiento')
```

R SELECT

```
A* El formato SELECT describe los literales y campos que utiliza
A* para entrar el código de selección mantenimiento para determinar
A* qué archivo maestro de información de tiempos desea mantener.
```

```
A BLINK
A 60 ALARM
A 2 5'PRG01'
A 2 30'Sistema información tiempos'
A 2 70DATE
A EDTCDE(Y)
A 3 30'Selección mantenimiento'
A 3 70TIME
A 6 14'Entre una X junto a la
A aplicación que desea mantener'
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++
```

```
A EMPAPL 1A B 9 25
A 9 28'Mantenimiento maestro empleados'
A PRJAPL 1A B 10 25
A 10 28'Mantenimiento maestro proyectos'
A RSNAPL 1A B 11 25
A 11 28'Mantenimiento maestro códigos razón'
A EMESS 50A O 21 16
A 60 DSPATR(HI)
A 23 7'F3-Fin de trabajo'
```

R EMPSEL

```
A* El formato EMPSEL describe los literales y campos que utiliza
A* para entrar el número de empleado y el código de acción de
A* mantenimiento para seleccionar un registro maestro de empleados.
```

```
A BLINK
A 60 ALARM
A 2 5'PRG01'
A 2 30'Sistema información tiempos'
```

```

A          2 70DATE
A          EDTCDE (Y)
A          3 27'Mantenimiento maestro empleados'
A          3 70TIME
A          6 27'Número empleado'
A          EMPNO      R      B 6 44REFFLD (RCEMP/EMPNO *LIBL/EMPMST)
A          8 27'Código acción'
A          ACODE      1A  B 8 44
A          8 48'A-Añadir'
A          9 48'C-Cambiar'
A          10 48'D-Suprimir'
    
```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre++++RLon++TdPBLínPosFunciones+++++*****
    
```

```

A          EMESS      50A  O 21 16
A 60          DSPATR (HI)
A          23 7'F3-Fin de trabajo'
A          23 25'F4-Selección mantenimiento'
A          R EMPMNT
    
```

A* El formato EMPMNT describe los literales y campos que utiliza para entrar el mantenimiento del maestro de empleados.

```

A          BLINK
A 61          ALARM
A          CA05(05 'volver a selección
A          empleado')
A          2 5'PRG01'
A          2 30'Sistema información tiempos'
A          2 70DATE
A          EDTCDE (Y)
A          3 70TIME
A          3 27'Mantenimiento maestro empleados'
A          5 14'Número'
A          EMPNO      R      O 5 22REFFLD (RCEMP/EMPNO *LIBL/EMPMST)
A          5 31'Nombre'
A          ENAME      R      B 5 37REFFLD (RCEMP/ENAME *LIBL/EMPMST)
A 90          DSPATR (PR)
A          7 18'Categoría'
A          EMCAT      R      B 7 37REFFLD (RCEMP/EMCAT *LIBL/EMPMST)
A 90          DSPATR (PR)
A          8 18'Departamento'
A          EDEPT      R      B 8 37REFFLD (RCEMP/EDEPT *LIBL/EMPMST)
A 90          DSPATR (PR)
A          9 18'Ubicación'
A          ELOCN      R      B 9 37REFFLD (RCEMP/ELOCN *LIBL/EMPMST)
    
```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre++++RLon++TdPBLínPosFunciones+++++*****
    
```

```

A 90          DSPATR (PR)
A          10 18'USRID'
A          EUSRI      R      B 10 37REFFLD (RCEMP/EUSRI *LIBL/EMPMST)
A 90          DSPATR (PR)
A          11 18'Horas semana normal'
A          ENHRS      R      B 11 37REFFLD (RCEMP/ENHRS *LIBL/EMPMST)
A 90          DSPATR (PR)
A          13 30'Historia información tiempos'
A          15 40'Mes      Año a la      Año'
A          16 40' actual      fecha      anterior'
A          17 19'Relacionado proyecto'
A          EPHRC      R      O 17 41REFFLD (RCEMP/EPHRC *LIBL/EMPMST)
A          EPHRY      R      O 17 49REFFLD (RCEMP/EPHRY *LIBL/EMPMST)
A          EPHRP      R      O 17 58REFFLD (RCEMP/EPHRP *LIBL/EMPMST)
A          18 19'No relacionado proyecto'
A          EPNRC      R      O 18 41REFFLD (RCEMP/EPNRC *LIBL/EMPMST)
A          EPNRY      R      O 18 49REFFLD (RCEMP/EPNRY *LIBL/EMPMST)
A          EPNRP      R      O 18 58REFFLD (RCEMP/EPNRP *LIBL/EMPMST)
A          EMESS      50  O 21 16
A 61          DSPATR (HI)
A          23 7'F3-Fin de trabajo'
A          23 25'F4-Selección mantenimiento'
A          23 55'F5-Selección empleado'
A          11 44'(ej. 40,0 entre 400)'
    
```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre++++RLon++TdPBLínPosFunciones+++++*****
    
```

A* El formato PRJSEL describe los literales y campos que utiliza para entrar el código de proyecto y el código de acción de mantenimiento para seleccionar un registro maestro de proyectos.

```

A          BLINK
A 60          ALARM
    
```

```

A          2 5'PRG01'
A          2 30'Sistema información tiempos'
A          2 70DATE
A          EDTCDE (Y)
A          3 27'Mantenimiento maestro proyectos'
A          3 70TIME
A          6 27'Código proyecto'
A          PRCDE      R      B 6 44REFFLD (RCPRJ/PRCDE *LIBL/PRJMST)
A          8 27'Código acción'
A          ACODE      1A  B 8 44
A          8 48'A-Añadir'
A          9 48'C-Cambiar'
A          10 48'D-Suprimir'
A          EMESS      50  O 21 16
A 60          DSPATR (HI)
A          23 7'F3-Fin de trabajo'
A          23 25'F4-Selección mantenimiento'
    
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

AAN01N02N03T.Nombre+++RLon++TDpBLínPosFunciones+++++

A R PRJMNT

A*

A* El formato PRJMNT describe los literales y campos que utiliza

A* para entrar el mantenimiento del maestro de proyectos.

A*

```

A          BLINK
A 61          ALARM
A          CA06(06 'volver a selección
A          proyecto')
A          2 5'PRG01'
A          2 30'Sistema información tiempos'
A          2 70DATE
A          EDTCDE (Y)
A          3 70TIME
A          3 27'Mantenimiento maestro proyectos'
A          5 29'Código proyecto'
A          PRCDE      R      O 5 43REFFLD (RCPRJ/PRCDE *LIBL/PRJMST)
A          7 10'Descripción'
A          PRDSC      R      B 7 23REFFLD (RCPRJ/PRDSC *LIBL/PRJMST)
A 90          DSPATR (PR)
A          9 13'Responsabilidad'
A          PRRSP      R      B 9 41REFFLD (RCPRJ/PRRSP *LIBL/PRJMST)
A 90          DSPATR (PR)
A          10 13'Fecha inicio proyecto'
A          PRSTR      R      B 10 41REFFLD (RCPRJ/PRSTR *LIBL/PRJMST)
A 90          DSPATR (PR)
A          10 53'(MMDDAA)'
A          11 13'Fecha estimada final proyecto'
A          PREND      R      B 11 41REFFLD (RCPRJ/PREND *LIBL/PRJMST)
A 90          DSPATR (PR)
A          11 53'(MMDDAA)'
    
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

AAN01N02N03T.Nombre+++RLon++TDpBLínPosFunciones+++++

```

A          12 13'Fecha finalización proyecto'
A          PRCMP      R      B 12 41REFFLD (RCPRJ/PRCMP *LIBL/PRJMST)
A 90          DSPATR (PR)
A          12 53'(MMDDAA)'
A          13 13'Horas estimadas proyecto'
A          PREST      R      B 13 41REFFLD (RCPRJ/PREST *LIBL/PRJMST)
A 90          DSPATR (PR)
A          15 33'Historia proyecto'
A          17 27'Actual'
A          17 37'Año a la Anterior'
A          18 28'Mes'
A          18 39'Fecha Año'
A          PRHRC      R      O 19 27REFFLD (RCPRJ/PRHRC *LIBL/PRJMST)
A          PRHRY      R      O 19 36REFFLD (RCPRJ/PRHRY *LIBL/PRJMST)
A          PRHRP      R      O 19 47REFFLD (RCPRJ/PRHRP *LIBL/PRJMST)
A          EMESS      50  O 21 16
A 61          DSPATR (HI)
A          23 7'F3-Fin de trabajo'
A          23 25'F4-Selección mantenimiento'
A          23 54'F6-Selección código proyecto'
    
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

AAN01N02N03T.Nombre+++RLon++TDpBLínPosFunciones+++++

A R RSNSEL

A* El formato RSNSEL describe los literales y campos que utiliza

A* para entrar el código de razón y el código de acción de

A* mantenimiento para seleccionar un registro maestro de códigos de

A* razón.

A*

```

A          BLINK
A 60          ALARM
    
```

Descripciones de datos de mantenimiento del archivo maestro - PRG01FM

```

A          2 5'PRG01'
A          2 30'Sistema información tiempos'
A          2 70DATE
A          EDTCDE (Y)
A          3 26'Mantenimiento maestro códigos razón'
A          3 70TIME
A          6 27'Código razón'
A          RSCDE      R          B 6 44REFFLD (RCRSN/RSCDE *LIBL/RSNMST)
A          8 27'Código acción'
A          ACODE      1A      B 8 44
A          8 48'A-Añadir'
A          9 48'C-Cambiar'
A          10 48'D-Suprimir'
A          EMESS      50      O 21 16
A 60          DSPATR (HI)
A          23 7'F3-Fin de trabajo'
A          23 25'F4-Selección mantenimiento'
A          R RSNMNT
A* El formato RSNMNT describe los literales y campos que utiliza
A* para entrar el mantenimiento del maestro de códigos de razón.

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre++++RLon++TdPBLínPosFunciones+++++*****
A          BLINK
A 61          ALARM
A          CA07 (07 'volver a selección código
A          razón')
A          2 5'PRG01'
A          2 30'Sistema información tiempos'
A          2 70DATE
A          EDTCDE (Y)
A          3 70TIME
A          3 27'Mantenimiento maestro empleados'
A          5 30'Código razón'
A          RSCDE      R          O 5 43REFFLD (RCRSN/RSCDE *LIBL/RSNMST)
A          7 9'Descripción'
A          RSDSC      R          B 7 22REFFLD (RCRSN/RSDSC *LIBL/RSNMST)
A 90          DSPATR (PR)
A          9 31'Historia código razón'
A          11 26'Actual'
A          11 36'Año a la      Anterior'
A          12 27'Mes'
A          12 38'Fecha      Año'
A          RSHRC      R          O 13 26REFFLD (RCRSN/RSHRC *LIBL/RSNMST)
A          RSHRY      R          O 13 35REFFLD (RCRSN/RSHRY *LIBL/RSNMST)
A          RSHRP      R          O 13 46REFFLD (RCRSN/RSHRP *LIBL/RSNMST)
A          EMESS      50      O 21 16
A 61          DSPATR (HI)
A          23 7'F3-Fin de trabajo'
A          23 25'F4-Selección mantenimiento'
A          23 54'F7-Selección código razón'

```

Figura 171. Descripciones de datos de mantenimiento del archivo maestro - PRG01FM

13.4.9 Programa en RPG/400 de mantenimiento de archivos maestros - PRG01

La Figura 172 muestra el programa PRG01 en RPG/400. El programa contiene comentarios intercalados para explicar el flujo lógico y la utilización de las funciones y de los códigos de operación RPG/400.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*****
F* PRG01 - Programa RPG/400 de Mantenimiento del Archivo Maestro
F* DESCRIPCION - El mantenimiento del archivo maestro de información
F* de tiempos que utiliza un proceso de estación de
F* trabajo descrito externamente.
F*****
F* INDICADORES UTILIZADOS:
F* 50 - No se ha encontrado ningún registro en la operación CHAIN
F* 60 - Condición de error general
F* 90 - Pantalla protegida en solicitud supresión
F* KC - Solicitado fin de trabajo
F* KD - Volver a selección de aplicación
F* KE - Volver a selección de empleado
F* KF - Volver a selección de proyecto
F* KG - Volver a selección de código de razón
F* LR - Ultimo registro
F*****
F* SUBROUTINAS UTILIZADAS:
F* EDITSL - Editar pantalla selección aplicación (SELECT)
F* ACDESR - Editar código acción para todas solicitudes mantenimiento
F*****
F* Este programa utiliza todos los archivos descritos externamente.
F* los archivos utilizados son:
F* PRG01FM - Archivo de pantalla de mantenimiento
F* EMPMST - Archivo maestro de empleados
F* PRJMST - Archivo maestro de proyectos
F* RSNMST - Archivo maestro de códigos de razón
F*****

```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF....LregLCIAIDePcLaEDispos+.....KSalidaEntradA....U1.*
FPRG01FM CF E WORKSTN
FEMPMST UF E K DISK A
FPRJMST UF E K DISK A
FRSNMST UF E K DISK A

```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
E* La matriz en tiempo de compilación que contiene las descripciones
E* de los errores.
E...Arch(de)Arch(a)+NombreN/rN/tbLonPDSNommatLonPDSComentarios+++++
E ERR 1 10 50
E*

```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C* CALCULOS PRINCIPALES
C*****
C* Esta rutina principal controla el proceso y la edición del
C* archivo de pantalla. Mediante la utilización de las teclas de
C* función descritas en cada formato de pantalla, puede realizar
C* transferencias de una aplicación de mantenimiento a otra. El
C* código de acción que selecciona en los formatos de selección
C* determina si el programa añade un nuevo registro al archivo o si
C* actualiza un registro existente en el archivo.
C*****
C* El programa contiene varias operaciones TAG. El programa se bifurcará
C* a estos TAG basándose en la acción que realice o en la tecla de
C* función que pulse en los distintos formatos. El BEGIN TAG cuenta
C* con una etiqueta a la que el programa se bifurca si pulsa F4
C* en cualquiera de los formatos de mantenimiento.
C* El término 'preparación' utilizado en este programa hace
C* referencia a la inicialización de indicadores, campos de trabajo
C* temporales, y campos de pantalla. La preparación asegura que la
C* información de operaciones de entrada o de cálculo anteriores que
C* podría afectar a las operaciones que el programa ejecuta a
C* continuación no se conserva. El indicador 60 (*IN60) está
C* desactivado, y se mueven blancos hacia los campos de la pantalla
C* de formato SELECT como parte de la preparación.

```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C BEGIN TAG
C MOVE '0' *IN60
C MOVE *BLANKS EMESS
C MOVE *BLANKS EMPAPL
C MOVE *BLANKS PRJAPL
C MOVE *BLANKS RSNAPL

```

```

C*
C* El SELTAG TAG proporciona una etiqueta a la que el programa
C* se bifurca si se encuentran errores en el formato de selección de
C* mantenimiento SELECT. El formato SELECT se escribe en la estación
C* de trabajo utilizando EXFMT. El EXFMT permite escribir en la
C* pantalla y leer de la misma. Si pulsa F3 (*INKC = 1), el programa
C* se bifurca al TAG END. Si no pulsa F3 (*INKC = 0), el programa
C* procesa la subrutina EDITSL para editar la entrada de formato
C* SELECT.
C*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          SELTAG      TAG
C          *INKC      CABEQ'1'      END
C          EXFMTSELECT
C          EXSR EDITSL
C*
C* Si el indicador de error general *IN60 está activado (igual a 1),
C* el programa vuelve a bifurcarse al SELTAG.
C*
C          *IN60      CABEQ'1'      SELTAG
C*
C* En este punto, el formato SELECT se ha verificado y el programa
C* visualiza el formato de entrada de mantenimiento para la aplicación
C* seleccionada. Los campos de selección de la aplicación del formato
C* SELECT se comprueban y el programa se bifurca a la selección específica
C* de la aplicación. Si EMPAPL (mantenimiento de empleados) es igual a
C* X, el programa se bifurca a la etiqueta EMPTAG. Si PRJAPL (mantenimiento
C* de proyectos) es igual a X, el programa se bifurca a la etiqueta PRJTAG.
C* Si las dos comprobaciones anteriores no fueron satisfactorias, ha
C* elegido el mantenimiento de códigos de razón. El programa
C* continúa con la operación siguiente.
C*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          EMPAPL      CABEQ'X'      EMPTAG
C          PRJAPL      CABEQ'X'      PRJTAG
C*
C*****
C* Mantenimiento de códigos de razón: El RSNTAG TAG proporciona una
C* etiqueta a la que el programa se bifurca si pulsa F7 desde el formato
C* de entrada de mantenimiento de códigos de razón RSNSEL. El programa
C* se bifurca a este identificador cuando la petición de mantenimiento
C* se completa satisfactoriamente.
C* Preparación: Inicialice el indicador de errores generales 60 y
C* borre los campos de la pantalla en formato RSNSEL.
C*
C          RSNTAG      TAG
C          MOVE '0'      *IN60
C          MOVE *BLANKS  EMESS
C          MOVE *BLANKS  RSCDE
C          MOVE *BLANKS  ACODE
C*
C* El programa se bifurca al RSNERR TAG si se encuentran errores
C* durante la edición de la entrada de formato RSNSEL.
C* Preparación: Inicialice los campos de la pantalla en formato RSNMNT.
C* El formato RSNSEL se escribe utilizando la operación EXFMT. Cuando
C* pulsa Intro o una tecla de función, el programa continúa con la
C* operación siguiente. Si pulsa F4 (*INKD = 1), el programa se
C* bifurca al BEGIN TAG.
C*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          RSNERR      TAG
C          MOVE *BLANKS  RSDSC
C          EXFMTRSNSL
C          *INKD      CABEQ'1'      BEGIN
C*
C* Si pulsa F3 (*INKC = 1), el programa se bifurca al END TAG.
C* Si no pulsa F3, se accede al archivo maestro de códigos de
C* razón utilizando el código de razón (RSCDE) que ha entrado y la
C* operación CHAIN. Si no se encuentra el registro, se activa el
C* indicador 50 resultante (posiciones 54 y 55). La subrutina
C* ACDESR se procesa para editar su petición.
C*
C          *INKC      CABEQ'1'      END
C          RSCDE      CHAINRSNMST      50
C          EXSR ACDESR
C*
C* Si la edición que ha procesado la subrutina ACDESR detecta errores
C* en su petición, el indicador 60 de errores generales se activa

```

```

C* y el programa vuelve a bifurcarse al RSNERR TAG.
C*
C          *IN60      CABEQ'1'      RSNERR
C*
C* El formato RSNMNT se escribe utilizando la operación EXFMT. Si
C* pulse F4 (*INKD), el programa vuelve a bifurcarse al BEGIN TAG.
C* Si pulsa F7 (*INKG), el programa vuelve a bifurcarse al RSNTAG TAG.
C*
C          EXFMTRSNMNT
C          *INKD      CABEQ'1'      BEGIN
C          *INKG      CABEQ'1'      RSNTAG

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* El código siguiente efectúa las operaciones de actualización del
C* archivo de códigos de razón. El código contiene cinco niveles de
C* sentencias IF jerarquizadas. El inicio de cada nivel se indica
C* mediante Bnn, donde nn es el nivel jerarquizado. Los cálculos de
C* cada nivel se indican mediante nn. El final de cada nivel se
C* indica mediante Enn. En cada nivel, el campo ACREC (código de
C* registro activo) se actualiza a A para activo o D para suprimido.
C*
C* Primer nivel - si pulsa F3 para finalizar el trabajo, el programa
C* procesa los cálculo del nivel 01 y se bifurca al END TAG.
C* Segundo nivel - si entra el código de acción A, y el registro no
C* existe todavía (*IN50 = 1), el programa ESCRIBE el registro.
C* Tercer nivel - si entra el código de acción A, y el registro ya
C* existe (*IN50 = 0) con un código de registro activo de D (ACREC = D),
C* entonces el programa actualiza el registro existente.
C* Cuarto nivel - si entra el código de acción D, el registro se actualiza.
C* Quinto nivel - si entra el código de acción C, el registro se actualiza.
C*

```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          *INKC      IFEQ '0'      B01
C          ACODE      IFEQ 'A'      B02
C          *IN50      ANDEQ'1'      02
C          MOVE 'A'      ACREC      02
C          WRITERCRSN      02
C          ELSE      02
C          ACODE      IFEQ 'A'      B03
C          *IN50      ANDEQ'0'      03
C          ACREC      ANDEQ'D'      03
C          MOVE 'A'      ACREC      03
C          UPDATRCRSN      03
C          ELSE      03
C          ACODE      IFEQ 'D'      B04
C          MOVE 'D'      ACREC      04
C          UPDATRCRSN      04
C          ELSE      04
C          ACODE      IFEQ 'C'      B05
C          UPDATRCRSN      05
C          END      E05
C          END      E04
C          END      E03
C          END      E02
C          ELSE      01
C          GOTO END      01
C          END      E01
C*
C* Su petición de mantenimiento se ha completado y el programa vuelve
C* a bifurcarse al RSNTAG TAG.
C*
C          GOTO RSNTAG

```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C* La rutina de mantenimiento del maestro de empleados efectúa
C* los mismos pasos que los realizados en la rutina de código de razón.
C* Consulte el RSNTAG para obtener una explicación más detallada
C* de los siguientes pasos del proceso.
C*****
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          EMPTAG      TAG
C          MOVE '0'      *IN60
C          MOVE *BLANKS      EMESS
C          Z-ADD0      EMPNO
C          MOVE *BLANKS      ACODE
C          EMPERR      TAG
C          MOVE *BLANKS      ENAME
C          MOVE *BLANKS      EMCAT
C          MOVE *BLANKS      EDEPT
C          MOVE *BLANKS      ELOCN

```

```

C          MOVE *BLANKS      EUSRI
C          Z-ADD0          ENHRS
C* Visualice el formato de selección de empleado
C          EXFMTEMPSEL
C          *INKD          CABEQ'1'          BEGIN
C* Acceda al maestro de empleados para validar la petición de código
C* de acción.
C          *INKC          CABEQ'1'          END
C          EMPNO          CHAINEMPST          50
C          EXSR ACDESR
C          *IN60          CABEQ'1'          EMPERR
C* Visualice el formato de mantenimiento de empleados
C          EXFMTEMPMNT
C          *INKD          CABEQ'1'          BEGIN
C          *INKE          CABEQ'1'          EMPTAG

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C          *INKC          IFEQ '0'          B01
C          ACODE          IFEQ 'A'          B02
C          *IN50          ANDEQ'1'          02
C          MOVE 'A'          ACREC          02
C          WRITERCEMP          02
C          ELSE          02
C          ACODE          IFEQ 'A'          B03
C          *IN50          ANDEQ'0'          03
C          ACREC          ANDEQ'D'          03
C          MOVE 'A'          ACREC          03
C          UPDATRCEMP          03
C          ELSE          03
C          ACODE          IFEQ 'D'          B04
C          MOVE 'D'          ACREC          04
C          UPDATRCEMP          04
C          ELSE          04
C          ACODE          IFEQ 'C'          B05
C          MOVE 'A'          ACREC          05
C          UPDATRCEMP          05
C          END          E05
C          END          E04
C          END          E03
C          END          E02
C          ELSE          01
C          GOTO END          01
C          END          E01
C          GOTO EMPTAG

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C* La rutina de mantenimiento de maestro de proyectos efectúa los mismos
C* pasos que los realizados en la rutina de código de razón. Consulte
C* el RSNTAG para obtener una explicación más detallada
C* de los siguientes pasos del proceso.
C*****
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C          PRJTAG          TAG
C          MOVE '0'          *IN60
C          MOVE *BLANKS      EMESS
C          MOVE *BLANKS      PRCDE
C          MOVE *BLANKS      ACODE
C          PRJERR          TAG
C          MOVE *BLANKS      PRDSC
C          MOVE *BLANKS      PRRSP
C          Z-ADD0          PRSTR
C          Z-ADD0          PREND
C          Z-ADD0          PRCMP
C          Z-ADD0          PREST
C* Visualice el formato de selección de proyecto
C          EXFMTPRJSEL
C          *INKD          CABEQ'1'          BEGIN
C* Acceda al maestro de proyectos para validar la petición de
C* código de acción
C          *INKC          CABEQ'1'          END
C          PRCDE          CHAINPRJMST          50
C          EXSR ACDESR
C          *IN60          CABEQ'1'          PRJERR
C* Visualice el formato de mantenimiento de proyectos
C          EXFMTPRJMNT
C          *INKD          CABEQ'1'          BEGIN
C          *INKF          CABEQ'1'          PRJTAG

```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C* Pase a la modalidad de actualización y efectúe la adición
C* o actualización del registro

```

```

C          *INKC      IFEQ '0'                      B01
C          ACODE      IFEQ 'A'                      B02
C          *IN50      ANDEQ'1'                      02
C                      MOVE 'A'          ACREC        02
C                      WRITERCPRJ        02
C                      ELSE              02
C          ACODE      IFEQ 'A'                      B03
C          *IN50      ANDEQ'0'                      03
C          ACREC      ANDEQ'D'                    03
C                      MOVE 'A'          ACREC        03
C                      UPDATRCPRJ        03
C                      ELSE              03
C          ACODE      IFEQ 'D'                      B04
C                      MOVE 'D'          ACREC        04
C                      UPDATRCPRJ        04
C                      ELSE              04
C          ACODE      IFEQ 'C'                      B05
C                      MOVE 'A'          ACREC        05
C                      UPDATRCPRJ        05
C                      END              E05
C                      END              E04
C                      END              E03
C                      END              E02
C                      ELSE              01
C                      GOTO END          01
C                      END              E01
C                      GOTO PRJTAG

```

C* Fin de trabajo solicitado. El control pasa a este punto cuando
C* pulsa F3 (*INKC). El último indicador de registro *INLR se activa
C* y el programa finaliza.

```

C          END        TAG
C                      MOVE '1'          *INLR

```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

C*****

C* La subrutina EDITSL verifica la entrada de pantalla de selección
C* de la aplicación de información de tiempos.

C*****

CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++

```

C          EDITSL     BEGSR

```

C* Preparación: El indicador de errores generales *IN60 se desactiva
C* y el campo de mensajes de error EMESS queda en blanco.

```

C                      MOVE *BLANKS     EMESS 50
C                      MOVE '0'         *IN60

```

C* La combinación IF AND OR siguiente comprueba los campos de
C* selección de la aplicación para asegurarse de que sólo se ha
C* seleccionado una aplicación. Si se ha seleccionado más de una, el
C* indicador de errores generales *IN60 se activa (igual a 1) y el
C* mensaje de error se establece moviendo el elemento de matriz 2
C* (ERR,2) al campo EMESS.

```

C          Z-ADD0     SELCNT 10
C          EMPAPL     IFEQ 'X'
C                      ADD 1             SELCNT
C                      END
C          PRJAPL     IFEQ 'X'
C                      ADD 1             SELCNT
C                      END
C          RSNAPL     IFEQ 'X'
C                      ADD 1             SELCNT
C                      END
C          SELCNT     IFGT 1
C                      MOVE '1'         *IN60
C                      MOVE ERR,2       EMESS
C                      ELSE
C                      MOVE '0'         *IN60
C                      END

```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

C*

C* La combinación IF AND siguiente asegura que al menos se ha
C* seleccionado una aplicación. Los campos de selección de la
C* aplicación se comprueban y si todos ellos son iguales a ' '
C* (blancos), *IN60 se activa y el elemento de matriz 3 se mueve al
C* campo de mensajes de error.

C*

CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++

```

C          EMPAPL     IFEQ ' '
C          PRJAPL     ANDEQ' '
C          RSNAPL     ANDEQ' '
C                      MOVE '1'         *IN60
C                      MOVE ERR,3       EMESS

```

```

C          END
C* El código siguiente comprueba cada campo de selección de la aplicación
C* para asegurarse de que es ' ' (blanco) o igual a 'X'. Si alguno de
C* los tres campos de selección contienen un valor que no sea ' '
C* o 'X', *IN60 se activa y el elemento de matriz 1 se mueve al campo
C* de mensajes de error.
C      EMPAPL  IFNE ' '
C      EMPAPL  ANDNE'X'
C      MOVE '1'      *IN60
C      MOVE ERR,1    EMESS
C      END
C      PRJAPL  IFNE ' '
C      PRJAPL  ANDNE'X'
C      MOVE '1'      *IN60
C      MOVE ERR,1    EMESS
C      END
C      RSNAPL  IFNE ' '
C      RSNAPL  ANDNE'X'
C      MOVE '1'      *IN60
C      MOVE ERR,1    EMESS
C      END
C      ENDSR

```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C* La subrutina ACDESR verifica los códigos de acción de información
C* de tiempos para todas las selecciones de mantenimiento.
C*****
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C      ACDESR  BEGSR

```

C* Preparación: Los indicadores de error *IN60 y *IN61 se desactivan y el campo de mensajes de error EMESS queda en blanco. El indicador *IN90 se define en los formatos de pantalla de mantenimiento para proteger la pantalla en una petición de supresión. Aquí se desactiva (igual a 0) como parte de la preparación.

```

C      MOVE *BLANKS  EMESS
C      MOVE '0'      *IN60
C      MOVE '0'      *IN61
C      MOVE '0'      *IN90

```

C* Las sentencias comparar y bifurcar (CABEQ) siguientes efectúan dos funciones. Determinan el tipo de mantenimiento solicitado y se bifurcan a la etiqueta adecuada, y determinan también si el código de mantenimiento entrado no es correcto. La operación CABEQ comprueba que el campo ACODE (código de acción) tenga un valor de 'A' (añadir) y, si es correcto, se bifurca al ADDCDE TAG. También se comprueba que ACODE tenga 'C' y se envía al CHGDDE TAG y que tenga 'D' y se envía al DELCDE TAG. Si el campo ACODE no es igual a A, C, o D, *IN60 se activa y el elemento de matriz 4 se mueve al campo de mensajes de error. A continuación se bifurca al final de la etiqueta ACDEND de subrutina en la sentencia ENDSR.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C      ACODE    CABEQ'A'    ADDCDE
C      ACODE    CABEQ'C'    CHGCDE
C      ACODE    CABEQ'D'    DELCDE

```

C* Código de acción no válido

```

C      MOVE '1'      *IN60
C      MOVE ERR,4    EMESS
C      GOTO ACDEND

```

C* El código siguiente verifica la solicitud de adición. Si el indicador *IN50 es igual a '0' indica que el registro se ha encontrado en la operación CHAIN. Si el registro ya existe (*IN50 igual a 0) y el campo de estado de registro ACREC del registro es 'A' para activo, *IN60 se activa, y el elemento de matriz 5 se mueve a EMESS. Si el registro ya existe y el campo de estado de registro ACREC es 'D' para suprimido, el elemento de matriz 6 se mueve a EMESS. En el último error, no activamos el indicador de error porque el mensaje de error 6 es un mensaje de aviso que se visualiza en el formato de mantenimiento para indicar que el registro ya se ha suprimido. A continuación debe decidir si desea volver a activa el registro o volver a la pantalla de selección para cambiar la solicitud.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++

```

```

C          ADDCDE      TAG
C          *IN50      IFEQ '0'
C          ACREC      ANDEQ'A'
C                      MOVE '1'          *IN60
C                      MOVE ERR,5      EMESS
C                      ELSE
C          *IN50      IFEQ '0'
C          ACREC      ANDEQ'D'
C                      MOVE ERR,6      EMESS
C                      END
C                      END
C                      GOTO ACDEND
C*
C* El código siguiente verifica la solicitud de cambio. Si la
C* primera comprobación verifica que el registro no se encuentra en
C* la operación CHAIN (*IN50 igual a 1), *IN60 se activa y el elemento
C* de matriz se mueve a EMESS. La segunda comprobación verifica la
C* existencia del registro (*IN50 igual a 0) y que el campo de estado
C* de registro ACREC sea igual a 'D' para suprimido, *IN60 se activa
C* y el elemento de matriz 8 se mueve a EMESS.
C          CHGCDE      TAG
C          *IN50      IFEQ '1'
C                      MOVE '1'          *IN60
C                      MOVE ERR,7      EMESS
C                      ELSE
C          *IN50      IFEQ '0'
C          ACREC      ANDEQ'D'
C                      MOVE '1'          *IN60
C                      MOVE ERR,8      EMESS
C                      END
C                      END
C                      GOTO ACDEND
C*
C* .. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* El código siguiente verifica la solicitud de supresión. Cuando el
C* indicador de protección de campo *IN90 se activa por primera vez
C* (igual a 1), no se permite efectuar cambios en los datos existentes
C* en una solicitud de supresión. Cuando el programa comprueba que el
C* registro no se encuentra en la operación CHAIN (*IN50 igual a 1),
C* *IN60 se activa y el elemento de matriz 9 se mueve a EMESS. Una
C* segunda comprobación determina la existencia del registro (IN50
C* igual a 0) y que el campo de estado de registro ACREC sea igual
C* a 'D' para indicar que ya se ha suprimido, *IN60 se activa y el
C* elemento de matriz 10 se mueve a EMESS.
C*
C* CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          DELCDE      TAG
C                      MOVE '1'          *IN90
C          *IN50      IFEQ '1'
C                      MOVE '1'          *IN60
C                      MOVE ERR,9      EMESS
C                      ELSE
C          *IN50      IFEQ '0'
C          ACREC      ANDEQ'D'
C                      MOVE '1'          *IN60
C                      MOVE ERR,10     EMESS
C                      END
C                      END
C          ACDEND      ENDSR
C*
C* ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
C*
C* O*
C* O* A continuación se ha entrado la matriz en tiempo de compilación
C* O* ERR. La matriz está precedida por '*' para indicar el inicio
C* O* de la matriz, y empieza en la columna 1 de la
C* O* especificación de salida.
C* O*
C* ** Matriz ERR - Descripciones de errores
C* EL CODIGO DE SELECCION DE MANTENIMIENTO NO ES IGUAL A "X"
C* SELECCIONADA MAS DE UNA APLICACION PARA MANTENIMIENTO
C* NO HA SELECCIONADO NINGUNA APLICACION PARA MANTENIMIENTO
C* EL CODIGO DE ACCION NO ES IGUAL A "A", "C" O "S"
C* SOLICITADA ADICION PERO EL REGISTRO YA EXISTE EN EL ARCHIVO
C* AVISO - REGISTRO SUPRIMIDO ANTERIORMENTE
C* SOLICITADO EL CAMBIO PERO EL REGISTRO NO EXISTE
C* SOLICITADO EL CAMBIO PERO EL REGISTRO ESTA MARCADO PARA SUPRESION
C* SOLICITADA LA SUPRESION PERO EL REGISTRO NO EXISTE
C* SOLICITADA LA SUPRESION PERO EL REGISTRO YA SE HA SUPRIMIDO

```

Figura 172. Programa RPG/400 de ejemplo - PRG01

13.5 Mantenimiento del archivo de control

Seleccione la opción **2** (Mantenimiento del archivo de control) en el Menú Principal del Sistema de Información de Tiempo para cambiar la fecha de final de semana, la fecha de final de mes, o indicador de todas las entradas realizadas. Efectúe los cambios antes de ejecutar la actualización semanal y la actualización mensual. La Figura 173 muestra el Menú Principal del Sistema de Información de Tiempos. La opción 2 permite llamar al programa PRG02 utilizando el mandato CALL PGM(PRG02).

```
+-----+
| TMENU                               Sistema de Información de Tiempos |
|                                     Menú Principal                   |
|                                                                              |
| 1. Mantenimiento archivo maestro (PRG01)                               |
| 2. Mantenimiento archivo control (PRG02)                               |
| 3. Entrada transacciones archivo tiempos (PRG03)                       |
| 4. Actualización semanal archivo tiempos (PROC1)                       |
| 5. Actualización e información mensual archivo tiempos (PROC3)         |
|                                                                              |
| 8. Visualizar mensajes (DSPMSG)                                         |
| 9. Fin de sesión (SIGNOFF)                                              |
|                                                                              |
| Selección o mandato                                                     |
| ==>                                                                      |
| F3=Salir F4=Solicit F9=Recuperar F12=Cancelar                          |
| F13=Soporte usuario F16=Menú principal del sistema                     |
+-----+
```

Figura 173. Menú Principal del Sistema de Información de Tiempos

Subtemas

13.5.1 Mantenimiento del archivo de control - PRG02FM

13.5.2 Descripciones de datos del mantenimiento del archivo de control - PRG02FM

13.5.3 Programa en RPG/400 de Mantenimiento del archivo de control - PRG02

13.5.2 Descripciones de datos del mantenimiento del archivo de control - PRG02FM

La Figura 175 muestra las DDS para el archivo de pantalla de Mantenimiento del Archivo de Control PRG02FM. Las descripciones de datos describen la función y el aspecto de los formatos del archivo de pantalla. En el archivo de pantalla se han incluido comentarios para describir el código.

Se utilizan las palabras clave siguientes:

- BLINK** Hace parpadear el cursor.
- CAnn** Especifica la tecla de función, identificada por nn, que está disponible para ser utilizar.
- DSPATR** Especifica los atributos de pantalla para el campo.
- DSPSIZ** Especifica el tamaño de la pantalla en la que el programa puede abrir el archivo.
- EDTCDE** Especifica el código de edición para un campo con posibilidad de salida.
- INDARA** Elimina los indicadores de opción y de respuesta del almacenamiento intermedio y los sitúa en un área de indicadores de 99 bytes separada.
- REFFLD** Hace referencia a los atributos de un campo definido anteriormente.
- TIME** Visualiza la hora actual del sistema como un campo de constantes.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A* Esta pantalla permite el mantenimiento del archivo de control
A* de información de tiempos.
A*****
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++*****
A                                     DSPSIZ(24 80 *DS3)
A                                     PRINT
A                                     INDARA
A                                     CA03(03 'fin de trabajo')
A      R CTLMNT
A*
A* El formato CTLMNT describe el diseño de la entrada de
A* mantenimiento del archivo de control.
A*
A                                     CA12(12 'AVISO FECHA - ACEPTAR')
A                                     BLINK
A      2 5'PRG02'
A      2 30'Sistema información tiempos'
A      2 70DATE
A                                     EDTCDE(Y)
A      3 70TIME
A      3 23'Mantenimiento área datos archivo control'
A      6 17'Fecha final semana'
A      CWKDT      R      B 6 45REFFLD(RCCTL/CWKDT *LIBL/CTLFIL)
A      7 17'Fecha final mes'
A      CMTDT      R      B 7 45REFFLD(RCCTL/CMTDT *LIBL/CTLFIL)
A      8 17'Indicador todas entradas realizadas'
A      CALLE      R      B 8 45REFFLD(RCCTL/CALLE *LIBL/CTLFIL)
A      ERMESS      50A O 21 16
A 60
A                                     DSPATR(HI)
A      23 7'F3-Fin de trabajo'
A      6 55'(MMDDAA)'
A      7 55'(MMDDAA)'
A      8 55'(Y o N)'

```

Figura 175. Descripciones de datos del mantenimiento del archivo de control - PRG02FM

13.5.3 Programa en RPG/400 de Mantenimiento del archivo de control - PRG02

La Figura 176 muestra las especificaciones RPG/400 para el programa de mantenimiento del archivo de control. Como parte del programa se incluyen comentarios para describir las diversas secciones del código.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*****
F* PRG02 - Programa en RPG/400 de Mantenimiento del Archivo de
F* Control
F* DESCRIPCION - El mantenimiento del archivo de control de información
F* de tiempos que utiliza el proceso de estación de
F* trabajo descrito por el programa.
F*****
F* INDICADORES UTILIZADOS:
F* 01 - Entrada de pantalla de mantenimiento del archivo de control
F* 50 - Año bisiesto
F* 51 - Entrada fecha no válida
F* 52 - Indicador de entradas de tiempos no válido
F* 53 - Entrado un año en la fecha que no es igual al año actual
F* 60 - Condición de error general
F* KC - Solicitado fin de trabajo
F* KL - Aceptar aviso de error
F* LR - Ultimo registro
F*****
F* SUBROUTINAS UTILIZADAS:
F* EDITSR - Editar campos de entrada desde el formato CTLMNT
F* DATESR - Editar formato de fecha
F*****
FNombarchIPEAF...LregLC1AIDePcLaEDispos+.....KSalidaEntradA...U1.*
FPRG02FM CP F 100 WORKSTN
E...Arch(de)Arch(a)+NombreN/rN/tbLonPDSNommatLonPDSComentarios++++*
E*
E* La matriz en tiempo de compilación que contiene las descripciones
E* de los errores.
E ERR 1 3 50
E* Las matrices en tiempo compilación que contienen los días por mes
E* para año no bisiesto y para año bisiesto.
E ARM 12 12 2 0
E ARL 12 12 2 0

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I*
I* Entrada del archivo de pantalla descrito por el programa para el
I* mantenimiento del archivo de control. Los campos de entrada son:
I* CWEEK - Fecha final semana
I* CMMTH - Fecha final mes
I* CENTR - Indicador todas entradas tiempos realizadas
I*
INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
I PRG02FM NS 01
I.....PDesdea++DCampo+L1M1FrPlMnZr...*
I 1 60CWEEK
I 7 120CMNTH
I 13 13 CENTR
I*
I* La constante con nombre siguiente define el nombre del formato
I* de registro para el archivo WORKSTN.
I*
I 'CTLMNT' C RECFMT
I*
I* Area de datos del archivo de control descrita externamente
I*
IDSnombre..NODsSal-arch++.....NvecLon+.....*
ICTLFIL EUDS
I*
I* Estructura de datos utilizada para la edición de la fecha: La
I* estructura de datos contiene un campo de fecha de 6 posiciones
I* con tres subcampos de 2 posiciones. Esto hace que el programa
I* pueda disponer de una referencia individual de mes, día, y año.

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
IDSnombre..NODsSal-arch++.....NvecLon+.....*
I DS
I.....PDesdea++DCampo+L1M1FrPlMnZr...*
I 1 60CDATE
I 1 20CDTMM
I 3 40CDTDD
I 5 60CDTTY

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C* CALCULOS PRINCIPALES
    
```

```

C*****
C* Este programa utiliza el ciclo de programa RPG/400 para gestionar
C* las entradas y las salidas del archivo de pantalla y de la
C* estructura de datos del área de datos. La rutina MAINLINE
C* comprueba primero una petición de fin de trabajo indicada por
C* el indicador de tecla de función F3 (*INKC). Si *INKC está
C* desactivado (igual a 0), el indicador de error general *IN60 se
C* desactiva (igual a 0), y el campo de mensaje de error se llena
C* con blancos. La subrutina EDITSR se procesa para validar la
C* entrada del archivo de pantalla.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          *INKC          IFEQ '0'
C                      MOVE '0'          *IN60
C                      MOVE *BLANKS      EMESS
C                      EXSR EDITSR
C*
C* El control vuelve a la sentencia que sigue a la línea EXSR
C* superior y se comprueba el indicador de error general *IN60 para
C* ver si está desactivado (igual a 0), y la entrada del archivo de
C* pantalla se mueve a los campos de la estructura de datos del área
C* de datos. El último indicador de registro *INLR se activa (igual
C* a 1) y el programa finaliza.
C*
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          *IN60          IFEQ '0'
C                      MOVE CWEEK       CWKDT
C                      MOVE CMNTH       CMTDT
C                      MOVE CENTR       CALLE
C                      MOVE '1'        *INLR
C                      END
C*
C* La sentencia END precedente señala el final de la segunda
C* sentencia IF.
C*
C* La sentencia ELSE siguiente se asocia a la comprobación de
C* sentencia IF inicial del indicador *INKC de fin de trabajo
C* solicitada. Las sentencias que siguen a ELSE realizan la
C* activación de condición del indicador *INKC (Igual a 1)
C* activando el último indicador de registro *INLR.
C*
C                      ELSE
C                      MOVE '1'        *INLR
C                      END
C*
C* La sentencia END precedente señala el final de la sentencia
C* IF inicial.
C*
C*****
C* La subrutina EDITSR verifica las fechas de final de semanal y
C* final de mes entradas y todos los indicadores de todas las
C* entradas realizadas.
C*****
C          EDITSR      BEGSR
C*
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* El campo de entrada de pantalla CWEEK se mueve al campo de estructura
C* de datos para proporcionar una referencia por separado del mes, día
C* y año. La subrutina DATESR se procesa para verificar el formato de
C* fecha y el control vuelve a la sentencia que sigue a EXSR. Si no se
C* encuentran errores en la fecha de final de semana (*IN51 y *IN53
C* ambos igual a 0), se realiza el mismo proceso utilizando la fecha
C* de final de mes.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          MOVE CWEEK       CDATE
C          EXSR DATESR
C          *IN51          IFEQ '0'
C          *IN53          ANDEQ'0'
C          MOVE CMNTH       CDATE
C          EXSR DATESR
C          END
C*
C* El indicador de todas las entradas realizadas CENTR se comprueba
C* sólo para asegurar que se hayan entrado los valores Y o N. El
C* indicador de error *IN52 se desactiva (igual a 0). Si el campo es
C* igual a Y o N, ELSE *IN52 se activa (igual a 1).
C*
C          CENTR          IFEQ 'Y'
C          CENTR          OREQ 'N'

```

```

C          MOVE '0'          *IN52
C          ELSE
C          MOVE '1'          *IN52
C          END

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* El código siguiente comprueba los indicadores de errores y mueve el
C* mensaje de error adecuando desde la matriz en tiempo de compilación.
C* El indicador de error general *IN60 se activa (igual a 1) si
C* cualquiera de los tres indicadores de error está activado y se
C* mueve el mensaje de error.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          *IN51      IFEQ '1'
C          *IN52      OREQ '1'
C          *IN53      OREQ '1'
C          MOVE '1'          *IN60
C          END
C*
C  51          MOVE ERR,1      EMESS  50
C  52          MOVE ERR,2      EMESS
C  53          MOVE ERR,3      EMESS
C*
C          ENDSR
C*****
C* La subrutina DATESR verifica el formato de fecha entrado. La fecha
C* se ha movido a la estructura de datos del programa antes de que se
C* procese esta rutina.
C*****
C          DATESR      BEGSR
C*
C* El año entrado se procesa para determinar si se trata de un año
C* bisiesto. El año se divide entre 4 y el resto se mueve a un campo
C* de trabajo separado. El indicador 50 resultante, posiciones 58 y
C* 59 se activa (igual a 1) si el resto es igual a cero. Esto
C* indica un año bisiesto y se utiliza para adaptar los cálculos
C* siguientes.
C*
C          CDTYY      DIV  4          LEAPYR  30
C          MVR          LEAPRM  30      50

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* El indicador de error *IN51 se desactiva (igual a 0) y a continuación
C* se comprueba el mes. Si el mes es superior a 12 o inferior a 1,
C* el indicador de error *IN51 se activa (igual a 1).
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          MOVE '0'          *IN51
C          CDTMM      IFGT 12
C          CDTMM      ORLT 1
C          MOVE '1'          *IN51
C          END
C* El código siguiente verifica el día entrado. El mes entrado se
C* utiliza como índice para ARL y ARM de la matriz en tiempo de
C* compilación. La matriz de ARL contiene el número de días de un mes
C* durante un año bisiesto; la matriz de ARM contiene el número de
C* días de un mes de un año no bisiesto. Si el número de días entrado
C* es superior al elemento de matriz, el indicador 51 se activa.
C          *IN51      IFEQ '0'
C          Z-ADDCDTMM      M          20
C          *IN50      IFEQ '1'
C          CDTDD      COMP ARL,M          51
C          ELSE
C          CDTDD      COMP ARM,M          51
C          END
C          END
C* El año entrado se compara con la palabra reservada para el año del
C* sistema UYEAR. Si no son iguales, *IN53 es igual a 1, se emite un
C* mensaje de aviso. Si pulsa F12 (*INKL) para aceptar el valor
C* entrado, se ignora la verificación.
C          MOVE '0'          *IN53
C          *INKL      IFEQ '0'
C          CDTYY      IFNE UYEAR
C          MOVE '1'          *IN53
C          END
C          END
C          ENDSR

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
O*
O* El formato de pantalla CTLMNT se escribe en el primer ciclo del
O* programa RPG/400 adaptándolo al indicador 1P.

```

```

O*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OPRG02FM D          1P
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O
O                                K6 RECFMT
O                                CWKDT      6
O                                CMTDT     12
O                                CALLE     13
O* El formato de pantalla CTLMNT vuelve a visualizarse sólo si se
O* encuentran errores. Debe corregir (a excepción del aviso de error
O* de la entrada del año) las entradas erróneas o pulsar F3 para
O* finalizar el trabajo.
O      D          01 60
O
O                                K6 RECFMT
O                                CWEEK      6
O                                CMNTH     12
O                                CENTR     13
O                                EMESS     63
O

* ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
O* A continuación se entra la matriz en tiempo de compilación ERR.
O* A la matriz precede "*** " para indicar el inicio de la matriz
O* y empieza en la columna 1 de la especificación de salida.

O*
** Matriz ERR - Descripciones de errores
FORMATO FECHA NO VALIDO DEBE SER MMDDAA
INDICADOR TODAS ENTRADAS NO VALIDO - DEBE SER "Y" O "N"
AVISO - AÑO NO = AÑO ACTUAL - PF12 PARA ACEPTAR
** Matriz ARM - días por mes de año no bisiesto
312831303130313130313031
** Matriz ARL - días por mes de año bisiesto
312931303130313130313031

```

Figura 176. Programa RPG/400 de ejemplo - PRG02

13.6 Entrada de transacciones del archivo de tiempos

Seleccione la opción **3** (Entrada de transacciones del archivo de tiempos) en el Menú Principal del Sistema de Información de Tiempos para entrar transacciones del archivo de tiempos de empleados. Las entradas puede efectuarse en cualquier momento antes de iniciar su actualización semanal. La Figura 177 muestra el Menú Principal del Sistema de Información de Tiempos. La opción **3** llama al programa PRG03 utilizando el mandato CALL PGM(PGM03).

```
+-----+
| TMENU                               Sistema de Información de Tiempos |
|                                     Menú Principal                   |
|                                                                              |
| 1. Mantenimiento archivo maestro (PRG01)                             |
| 2. Mantenimiento archivo control (PRG02)                             |
| 3. Entrada transacciones archivo tiempos (PRG03)                   |
| 4. Actualización semanal archivo tiempos (PROC1)                     |
| 5. Actualización e información mensual archivo tiempos (PROC3)       |
|                                                                              |
| 8. Visualizar mensajes (DSPMSG)                                       |
| 9. Fin de sesión (SIGNOFF)                                            |
|                                                                              |
| Selección o mandato                                                    |
| ==>                                                                      |
| F3=Salir F4=Solicit F9=Recuperar F12=Cancelar                         |
| F13=Soporte usuario F16=Menú principal del sistema                    |
+-----+
```

Figura 177. Menú Principal del Sistema de Información de Tiempos

Subtemas

- 13.6.1 Entrada de transacciones de información de tiempos - PRG03FM
- 13.6.2 Descripciones de datos de entrada de transacciones de información de tiempos - PRG03FM
- 13.6.3 Programa en RPG/400 de entrada de transacciones de información de tiempos - PRG03

13.6.1 Entrada de transacciones de información de tiempos - PRG03FM

La aplicación de la entrada de transacciones del sistema de información de tiempos consta de dos pantallas. En la primera pantalla, se entra un número de empleado y se cambia la fecha de final de semana. En la segunda pantalla, se entran las transacciones de detalle. La segunda pantalla consta de tres formatos: un formato de registro de control de subarchivos, un formato de registro de subarchivos, y un formato que contiene descripciones de teclas de función válidas y mensajes de error.

Subtemas

13.6.1.1 Pantalla de Selección de Empleados

13.6.2 Descripciones de datos de entrada de transacciones de información de tiempos - PRG03FM

La Figura 180 muestra las DDS para el archivo de pantalla de Entrada de Transacciones de Información de Tiempos **PRG03FM**. Los cuatro formatos de registro, identificados por una **R** en la posición 17, preceden al nombre de formato en las posiciones 19 a 28. Se utilizan las palabras clave siguientes:

- ALARM** Activa la alarma audible.
- BLINK** Hace parpadear el cursor.
- CAnn** Hace que la tecla de función especificada en la palabra clave esté disponible para su utilización.
- DATE** Visualiza la fecha del trabajo actual como una constante.
- DSPATR** Especifica un atributo de pantalla para el campo.
- DSPSIZ** Especifica el tamaño de la pantalla en la que el programa puede abrir este archivo.
- EDTCDE** Especifica la edición en un campo numérico con posibilidad de salida.
- EDTWRD** Especifica una palabra de edición en un campo numérico con posibilidad de salida.
- OVERLAY** Especifica que el formato de registro que está definiendo aparece en la pantalla sin tener que borrar primero toda la pantalla.
- REFFLD** Hace referencia a los atributos de un campo definido anteriormente.
- SFL** Palabra clave a nivel de registro que especifica que este formato de registro será un formato de registro de subarchivo.
- SFLCLR** Utilizado en el formato de registro de control de subarchivo para que su programa pueda borrar todos los registros del subarchivo.
- SFLCTL** Palabra clave a nivel de registro que especifica que este formato de registro será un formato de registro de control de subarchivo.
- SFLDSP** Utilizado en el formato de registro de control de subarchivo para que el sistema OS/400 visualice el subarchivo cuando su programa envíe una operación de salida al formato de registro de control de subarchivo.
- SFLDSPSTL** Utilizadas en el formato de registro de control de subarchivo para que el sistema OS/400 visualice campos en el formato de registro de control de subarchivo cuando su programa envíe una operación de salida al formato de registro de control de subarchivo.
- SFLPAG** Utilizada en el formato de registro de control de subarchivo para especificar el número de registros de subarchivo que se visualizarán a la vez.
- SFLSIZ** Utilizada en el formato de registro de control de subarchivo para especificar el número de registros del subarchivo .
- TIME** Visualiza la hora actual del sistema como una constante.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A* PRG03FM - Descripciones de datos de entrada de transacciones
A* DESCRIPCION - Un archivo que contiene formatos de registro para
A* las transacciones de archivo de tiempos de empleados.
A*****
AAN01N02N03T.Nombre++++RLon++TdPBLínPosFunciones+++++*****
A
A                                DSPSIZ(24 80 *DS3)
A                                R EMPSEL
A                                CA03(03 'Fin de trabajo')
A                                BLINK
A 60                               ALARM
A                                2 5'PRG03'
A                                2 30'Sistema de información de tiempos'
A                                2 71DATE
A                                EDTCDE(Y)
A                                3 31'Entrada de tiempos de empleados'
A                                3 71TIME
A                                6 29'Número de empleado'
A                                EMPNO      R      B 46REFFLD(RCEMP/EMPNO *LIBL/EMPMST)
A                                8 28'Fecha final semana'
A                                CWKDTX     R      B 8 46REFFLD(RCCTL/CWKDT *LIBL/CTLFIL)
A                                EMESS      50A  O 21 15
A 60                               DSPATR(HI)
A                                23 6'F3-Fin de trabajo'
A                                R EMPFIL
A                                RECNO      3S 0B 11 2DSPATR(ND)
A                                DSPATR(PR)
A                                STATUS     1A  B 11 10
A                                PRFLAG    2   O 11 14
A 60                               DSPATR(HI)
A 60                               DSPATR(BL)
A                                PRCDEX    R      B 11 17REFFLD(RCWEK/PRCDE *LIBL/TRWEEK)
A                                RSFLAG    2   O 11 27

```

```

A 60 DSPATR(HI)
A 60 DSPATR(BL)
A RSCDEX R B 11 30REFFLD(RCWEK/RSCDE *LIBL/TRWEEK)
A HRFLAG 2 O 11 40

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++*****
A 60 DSPATR(HI)
A 60 DSPATR(BL)
A EHWRKX R B 11 43REFFLD(RCWEK/EHWRK *LIBL/TRWEEK)
A ACDATX R B 11 54REFFLD(RCWEK/ACDAT *LIBL/TRWEEK)
A TFRRN R B 11 67REFFLD(RCWEK/TFRRN *LIBL/TRWEEK)
A DSPATR(PR)
A R EMPCTL SFLCTL(EMPFIL)
A SFLSIZ(0050)
A SFLPAG(0010)
A 30 SFLCLR
A N30 SFLDSP
A N30 SFLDSPCTL
A OVERLAY
A CA03(03 'Fin de trabajo')
A CA05(05 'Reiniciar pantalla empleado
A desde el principio')
A CA01(01 'Pantalla selección
A empleado')
A BLINK
A 60 ALARM
A 2 5'PRG03'
A 2 30'Sistema de información de tiempos'
A 2 71DATE
A EDTCDE(Y)
A 3 31'Entrada de tiempos de empleados'
A 3 71TIME
A 5 9'Núm empleado'
A EMPNO R O 5 16REFFLD(RCEMP/EMPNO *LIBL/EMPMST)
A DSPATR(CS)
A 5 25'Nombre'

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++*****
A ENAME R O 5 30REFFLD(RCEMP/ENAME *LIBL/EMPMST)
A DSPATR(CS)
A 5 63'Departamento'
A EDEPT R O 5 68REFFLD(RCEMP/EDEPT *LIBL/EMPMST)
A DSPATR(CS)
A 7 15'Horas entrada tiempos empleados
A en archivo semanal')
A CURHRS 5 10 7 57EDTWRD(' . ')
A DSPATR(CS)
A 9 8'Horas Fecha real Razón -
A acción proyecto'
A 9 66'Relativo'
A 10 9'Cód Cód Cód T-
A rab trab
A 10 66'Registro núm'
A R EMPERR
A EMESS 50 O 22 17
A 60 DSPATR(HI)
A 23 6'F3-Fin de trabajo'
A 23 24'F1-Selección empleado'
A 23 50'F5-Reiniciar pantalla empleado'

```

Figura 180. Descripciones de datos de entrada de transacciones de información de tiempos - PRG03FM

13.6.3 Programa en RPG/400 de entrada de transacciones de información de tiempos - PRG03

La Figura 181 muestra el programa en RPG/400 para la entrada de transacciones de información de tiempos. Como parte del programa, se incluyen comentarios para describir las diversas secciones del código y la lógica del RPG/400.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*****
F* PRG03 - Programa RPG/400 entrada transac. información de tiempos
F* DESCRIPCION - Entrada de transacciones de tiempos que utiliza el
F* proceso de estación de trabajo de subarchivo.
F*****
F* INDICADORES UTILIZADOS:
F* 31 - La lectura del registro no coincide con el factor 1 de la
F* operación READE
F* 32 - El subarchivo está lleno
F* 35 - No existen más registros cambiados en el subarchivo
F* de la operación READC
F* 41 - El registro encontrado en la operación SETLL es igual al
F* factor 1
F* 45 - Registro no encontrado en el archivo TRWEEK en la operación CHAIN
F* 60 - Condición de error general
F* 64 - Registro no encontrado en el archivo EMPMST en la operación CHAIN
F* 65 - Registro no encontrado en el archivo PRJMST en la operación CHAIN
F* 66 - Registro no encontrado en el archivo RSNMST en la operación CHAIN
F* KA - Volver a selección de empleado
F* KC - Solicitado fin de trabajo
F* KE - Reiniciar la pantalla de selección de empleado
F* LR - Ultimo registro
F*****
F* SUBROUTINAS UTILIZADAS:
F* EMPEDT - Verifica el empleado solicitado en formato EMPSEL
F* SFLEDT - Verifica las entrada de subarchivo en formato EMPFIL
F*****

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*
F* Este programa utiliza todos los archivos descritos externamente.
F* Los archivos utilizados son:
F* PRG03FM - archivo de pantalla de entrada de transacciones
F* EMPMST - archivo maestro de empleados
F* PRJMST - archivo maestro de proyectos
F* RSNMST - archivo maestro de códigos de razón
F* TRWEEKL - vista lógica del archivo de transacciones
F* semanal por número de empleado EMPNO y
F* número de registro relativo TFRRN.
F*****
FNombarchIPEAF...LregLC1AIDePcLaEDispos+.....KSalidaEntrada...U1.*
FPRG03FM CF E WORKSTN
F RECNO KSFIL EMPFIL
F*
F* La línea de continuación para el archivo WORKSTN identifica el formato
F* de registro (EMPFIL) que va a utilizarse como un subarchivo. El campo
F* de número de registro relativo (RECNO) controla a qué registro dentro
F* del subarchivo está accediéndose.
F*
FEMPMST IF E K DISK
FPRJMST IF E K DISK
FRSNMST IF E K DISK
FTRWEEKL UF E K DISK A

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
E*
E* Matriz en tiempo de compilación que contiene las descripciones de error.
E*
E...Arch(de)Arch(a)+NombreN/rN/tbLonPDSNommatLonPDSComentarios++++*
E ERR 1 6 50
E*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I*
I* La especificación de entrada siguiente define un campo de constante
I* con nombre FLAG. Esta constante se utiliza para indicar
I* errores de transacción en la pantalla de subarchivo.
I*
I.....Constante+++++++C.....Campo+.....*
I '->' C FLAG
I*
I* Area de datos del archivo de control descrita externamente
I*
IDsnombre..NODsSal-arch++.....NvecLon+.....*
ICTLFIL EUDS

```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C*  CALCULOS PRINCIPALES
C*****
C*  Esta rutina principal controla el proceso de archivo de pantalla
C*  y la edición. Debe seleccionar en primer lugar un número de empleado
C*  de la pantalla EMPSEL. El número se verifica con el maestro de
C*  empleados. La pantalla de entrada de transacciones consta de tres
C*  formatos: registro de control de subarchivo EMPCTL, registro de
C*  subarchivo EMPFIL, y registro de pantallas y mensajes de error EMPERR.
C*  Puede volver a la pantalla de selección de empleado, finalizar el
C*  trabajo o reanudar la pantalla de subarchivo utilizando las teclas de
C*  función. Las teclas de giro le permite desplazarse a través de las
C*  entradas existentes. Puede añadir nuevos registros, y cambiar o suprimir
C*  registros existentes. El archivo de transacciones semanal se actualiza
C*  con entradas válidas cuando pulsa Intro.
C*****
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C*  La rutina de principal comienza moviendo la fecha de fin de semana
C*  hacia la fecha de fin de semana del archivo de pantalla. Esto protege
C*  la actualización de la fecha de fin de semana del área de datos porque
C*  puede cambiar la fecha del archivo EMPSEL. Como parte de la preparación,
C*  el indicador de error general *IN60 se desactiva, se mueven blancos
C*  al campo de pantalla de mensajes de error, y el número de empleado
C*  se pone a cero antes de grabar el formato EMPSEL. El BEGIN TAG
C*  proporciona una etiqueta a la que el programa puede volver cuando se
C*  solicita desde la pantalla de entrada utilizando la tecla de función 1 *INKA.
C*
```

```
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          MOVE CWKDT      CWKDTX
C          BEGIN          TAG
C          MOVE '0'        *IN60
C          MOVE *BLANKS    EMESS
C          Z-ADD0          EMPNO
```

```
C*  El SELTAG TAG proporciona una etiqueta a la que vuelve el programa
C*  si se encuentra un error en la pantalla de selección de empleado EMPSEL.
C*  El formato EMPSEL se graba en la estación de pantalla utilizando EXFMT.
C*  EXFMT produce una grabación a y una lectura desde la pantalla. La
C*  tecla de función F3 (*INKC) se comprueba para ver si se ha solicitado
C*  un fin de trabajo, y si no es así (*INKC igual a 0), se procesa la
C*  subrutina EMPEDT de edición de empleado. Si el fin de trabajo se ha
C*  solicitado mediante F3, que activa el indicador 03 (vea archivo de
C*  pantalla) el indicador LR de último registro se activa y la
C*  operación RETRN vuelve a iniciarse.
C*
```

```
C          SELTAG        TAG
C          EXFMTEMPSEL
C 03          SETON              LR
C 03          RETRN
C          EXSR EMPEDT
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C*  Si el indicador de error general *IN60 está activado (igual a 1),
C*  el programa vuelve a bifurcarse a SELTAG y vuelve a visualizar el
C*  formato EMPSEL con el mensaje de error adecuado.
C*
```

```
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          *IN60          IFEQ '1'
C          GOTO SELTAG
C          END
```

```
C*  Al llegar a este punto, el formato EMPSEL se ha verificado y el
C*  visualiza los formatos de entrada de tiempos. Los registros de
C*  transacciones de entrada de tiempos existentes para el empleado
C*  se visualizan. El REPEAT TAG proporciona una etiqueta a la que el
C*  programa puede bifurcarse si se ha solicitado F5 (*INKE) o cuando
C*  se hayan verificado todas las entradas de subarchivo y se hayan
C*  aplicado al archivo de transacciones. Los indicadores resultantes
C*  se desactivan (igual a 0) antes de realizarse el proceso.
C*
```

```
C          REPEAT        TAG
C          SETOF              313235
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C*  El código siguiente sitúa al programa en el número de empleado
C*  solicitado dentro del archivo de transacciones. Si se encuentra un
C*  registro que coincida con EMPNO, el indicador 41 resultante se activa.
C*  El indicador 30 se define en el formato de control de subarchivo para
```

C* evitar que se borre el formato de pantalla de subarchivo. El indicador
 C* 30 se activa y el formato EMPCTL se graba. El formato EMPERR también
 C* se graba para visualizar teclas de función válidas en la parte inferior
 C* de la pantalla. El indicador 30 se desactiva y los campos de trabajo se
 C* inicializan. RECNO se define en la línea de continuación de
 C* definición de archivo. LSRRN se utiliza para almacenar el último
 C* número de registro relativo del archivo de transacciones; CURHRS
 C* se utiliza para visualizar las horas totales que se han entrado para
 C* el empleado, y STATUS se utiliza para permitir la supresión de
 C* los registros de subarchivo existentes.

C*
 CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*

```
C      EMPNO      SETLLTRWEEKL      41
C      MOVE '1'      *IN30
C      WRITEEMPCTL
C      WRITEEMPERR
C      MOVE '0'      *IN30
C      Z-ADD0      RECNO      30
C      Z-ADD0      LSRRN      50
C      Z-ADD0      CURHRS      51
C      MOVE *BLANKS      STATUS
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

C*
 C* La operación DOWEQ siguiente se procesa hasta que el indicador 31
 C* (*IN31) se activa mediante la operación READE o mediante el subarchivo
 C* que está llenándose. READE lee todas las entradas existentes en el
 C* archivo de transacciones que sean iguales al número de empleado (EMPNO).
 C* Si se encuentra una entrada (*IN31 igual a 0), los campos del archivo
 C* de transacciones se mueven a los campos de pantalla de subarchivo.
 C* Se utilizan tres distintivos de error en cada registro de subarchivo
 C* para señalar las entradas no válidas; estos distintivos se borran:
 C* Se incrementa en 1 el número de registro relativo RECNO, las horas
 C* del registro de transacciones se añaden al total para el empleado,
 C* y el registro de subarchivo se graba. Cuando el subarchivo está
 C* completo, el indicador 32 resultante en la operación WRITE se activa.

C*
 CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*

```
C      *IN31      DOWEQ'0'
C      EMPNO      READERCWEK      31
C
C      *IN31      IFEQ '0'
C      MOVE PRCDE      PRCDEX
C      MOVE RSCDE      RSCDEX
C      Z-ADDEHWRK      EHWRKX
C      Z-ADDACDAT      ACDATX
C      Z-ADDTFRRN      LSRRN
C      MOVE *BLANKS      PRFLAG
C      MOVE *BLANKS      RSFLAG
C      MOVE *BLANKS      HRFLAG
C      ADD 1      RECNO
C      ADD EHWRK      CURHRS
C      WRITEEMPFIL      32
C      END
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

C*
 C* Si el indicador de subarchivo lleno indicador (*IN32), el indicador
 C* 31 se activa para finalizar la operación DOWEQ. Si 31 está desactivado,
 C* el código anterior vuelve a procesarse.

C*
 CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*

```
C      *IN32      IFEQ '1'
C      MOVE '1'      *IN31
C      END
C      END
```

C* El END anterior señala el final de la operación DOW
 C* (Hacer Mientras).

C*
 C* El código siguiente determina si el subarchivo está lleno. Si el
 C* indicador 30 está desactivado (*IN30 igual a 0), entonces la
 C* operación DOWNQ se procesa hasta que el resto del subarchivo
 C* se llena con registros en blanco.

```
C      *IN32      DOWEQ'0'
C      MOVE *BLANKS      STATUS
C      MOVE *BLANKS      PRCDEX
C      MOVE *BLANKS      RSCDEX
C      Z-ADD0      EHWRKX
C      Z-ADD0      ACDATX
C      Z-ADD0      TFRRN
C      ADD 1      RECNO
C      WRITEEMPFIL      32
C      END
```

C*
 C* El END anterior señala el final de la operación DOW
 C* (Hacer Mientras).
 C*
 .. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..
 C*
 C* SFDISP TAG proporciona una etiqueta a la que el programa puede
 C* bifurcarse si se encuentran errores en los registros de subarchivo.
 C* Si el indicador 60 está activado, el formato EMPERR se graba para
 C* visualizar el mensaje de error seguido de la operación EXFMT para
 C* grabar el formato de subarchivo de control EMPCTL. Si se pulsa F1 (*INKA)
 C* desde la pantalla de entrada de transacciones, el indicador 01 se
 C* activa (vea archivo de pantalla) y el programa vuelve al identificador
 C* BEGIN y visualiza el formato de selección de empleado EMPSEL. Si se
 C* entra F3 (*INKC) para solicitar el fin del trabajo, el indicador 03 se
 C* activa. Esto a su vez activa el indicador LR de último registro y la
 C* operación RETRN. Si se entra (*INKE) para solicitar una revisualización
 C* de las entradas de tiempos de empleados, el programa vuelve a bifurcarse
 C* a REPEAT TAG y el subarchivo vuelve a crearse. No se efectúan
 C* actualizaciones de archivo si se utiliza alguna de estas teclas de
 C* función.

```

CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          SFDISP      TAG
C  60          WRITEEMPERR
C          EXFMTEMPCTL
C  01          GOTO BEGIN
C  03          SETON              LR
C  03          RETRN
C  05          GOTO REPEAT
C*
    
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..
 C*
 C* El código siguiente prepara al programa para procesar el subarchivo.
 C* El número de registro relativo de subarchivo se establece a un
 C* valor de 1 utilizando la operación Z-ADD, y el indicador de error *IN60
 C* se desactiva. La clave del archivo de transacciones se establece
 C* utilizando los códigos de operación KLIST y KFLD. Las operaciones
 C* KLIST y KFLD son operaciones declarativas (no pueden procesarse)
 C* que indican el argumento de búsqueda del archivo. Podrían haber
 C* aparecido en cualquier lugar dentro de los cálculos, pero se
 C* codifican aquí a efectos de documentación.
 C*

```

CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          Z-ADD1      RECNO
C          MOVE '0'    *IN60
C          TRKEY      KLIST
C          KFLD        EMPNO
C          KFLD        TFRRN
C*
    
```

C* La operación DOWEQ siguiente procesa el archivo de transacciones
 C* y todos los registros cambiados del subarchivo. La operación DOW
 C* (Hacer mientras) continúa hasta que el indicador 35 (*IN35) se
 C* activa. El indicador 35 se define como un indicador resultante en la
 C* operación (leer registros cambiados), que se activa cuando se leen
 C* todos los registros de subarchivo cambiados. Si el campo de pantalla
 C* STATUS está en blanco, y el indicador 35 está desactivado (igual a
 C* 0) la subrutina de edición de subarchivo SFLEDT se procesa. El
 C* programa vuelve de la edición y si el indicador de error 60 está
 C* activado, el programa vuelve a bifurcarse a SFDISP TAG, graba el
 C* formato EMPERR, y vuelve a visualizar el formato EMPCTL. Si el
 C* el indicador 35 está activado, el programa vuelve a bifurcarse a
 C* REPEAT TAG y reconstruye el subarchivo.
 C*

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          *IN35      DOWEQ'0'
C          READCEMPFIL          35
C          STATUS    IFEQ ' '
C          *IN35      ANDEQ'0'
C          EXSR SFLEDT
C          END
C  60          GOTO SFDISP
C  35          GOTO REPEAT
C*
    
```

C* El código siguiente sigue formando parte de la operación DOW (Hacer
 C* Mientras). AL utilizar el campo TRKEY construido por la operación
 C* KLIST, se tiene acceso al archivo de transacciones utilizando la
 C* operación CHAIN. Si el registro no existe en el archivo, el indicador
 C* 45 resultante se activa. Si el indicador 45 está activado, el
 C* registro debe añadirse al archivo de transacciones. El último número

C* de registro relativo, que se almacena en el campo LSRRN, se incrementa
 C* en 1 y los campos de pantalla se mueven al registro de transacciones.
 C* A continuación, el nuevo registro se graba utilizando la operación
 C* WRITE y el formato de registro RCWEEK.

```
C*
C          TRKEY          CHAINTRWEEKL          45
C          *IN45          IFEQ '1'
C          ADD 1          LSRRN
C          Z-ADDLSRRN    TFRRN
C          MOVE PRCDEX    PRCDE
C          MOVE RSCDEX    RSCDE
C          Z-ADDEHWRKX    EHWRK
C          Z-ADDACDATX    ACDAT
C          WRITERCWEEK
C          END
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

C* Si el indicador 45 está desactivado (*IN45 igual a 0) y el campo de
 C* archivo de pantalla STATUS es igual a D, el operador solicita la
 C* supresión del registro. El código de operación DELET se procesa y
 C* el registro se mueve del archivo. No puede recuperarse el registro.

```
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          *IN45          IFEQ '0'
C          STATUS        ANDEQ'D'
C          DELETRCWEEK
C          END
```

C* Si el indicador 45 está desactivado y el campo de archivo de pantalla
 C* STATUS no es igual a D, el operador ha cambiado un registro existente.
 C* Los campos de archivo de pantalla se mueven a los campos de archivo
 C* de transacciones y la operación UPDAT se procesa para actualizar el
 C* archivo.

```
C          *IN45          IFEQ '0'
C          STATUS        ANDNE'D'
C          MOVE PRCDEX    PRCDE
C          MOVE RSCDEX    RSCDE
C          Z-ADDEHWRKX    EHWRK
C          Z-ADDACDATX    ACDAT
C          UPDATRCWEEK
C          END
C          END
```

C* El END anterior indica el final de la operación DOW (Hacer Mientras).

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

```
C*****
C* La subrutina EMPEDT verifica el número de empleado solicitado.
C*****
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          EMPEDT        BEGSR
```

C* El indicador de error 60 (*IN60) se desactiva (igual a 0) y el
 C* campo de mensajes de error EMESS queda en blanco.

```
C          MOVE '0'          *IN60
C          MOVE *BLANKS      EMESS
```

C* Mediante la utilización del número de empleado entrado (EMPNO) y la
 C* operación CHAIN se tiene acceso al archivo maestro de empleados. Si
 C* se encuentra el registro, el indicador 64 resultante se activa. Si *IN64
 C* está activado, el indicador de error 60 se activa y el elemento de
 C* matriz de error 1 se mueve a EMESS. La operación ELSE indica que se
 C* ha encontrado el registro (*IN60 igual a 0) y se comprueba que el
 C* estado del registro tenga un valor de D, suprimido. Si así es, el
 C* indicador de error 60 se activa y el elemento de matriz de error
 C* 2 se mueve a EMESS.

```
C          EMPNO          CHAINEMPST          64
C          *IN64          IFEQ '1'
C          MOVE '1'          *IN60
C          MOVE ERR,1        EMESS
C          ELSE
C          ACREC            IFEQ 'D'
C          MOVE '1'          *IN60
C          MOVE ERR,2        EMESS
C          END
C          END
C          ENDSR
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

```

C*****
C* La subrutina SFLEDT verifica las entradas del subarchivo.
C*****
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C
C      SFLEDT      BEGSR
C*
C* Los distintivos de error de subarchivo PRFLAG, RSFLAG, y HRFLAG
C* pasan a estar en blanco, el indicador de error se desactiva y el
C* campo de mensajes de error EMESS se borra. Los distintivos de
C* error del subarchivo contienen -> (campo FLAG de constante con
C* nombre) cuando se encuentra un error. El distintivo parpadea en
C* la pantalla en modalidad de resaltado junto al campo(s)
C* erróneo(s).
C*
C
C      MOVE *BLANKS      PRFLAG
C      MOVE *BLANKS      RSFLAG
C      MOVE *BLANKS      HRFLAG
C      MOVE '0'          *IN60
C      MOVE *BLANKS      EMESS
C* La primera comprobación determina si los campos de subarchivo de
C* código de proyecto y de código de razón están en blanco. Si ambos
C* están en blanco, FLAG se mueve al distintivo de código de proyecto
C* y al distintivo de código de razón, se activa el indicador de
C* error, y el elemento de matriz de error 3 se mueve a EMESS.
C*
C
C      PRCDEX      IFEQ *BLANKS
C      RSCDEX      ANDEQ*BLANKS
C
C      MOVE FLAG      PRFLAG
C      MOVE FLAG      RSFLAG
C      MOVE '1'       *IN60
C      MOVE ERR,3     EMESS
C
C      ELSE
C*
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* La segunda comprobación determina si se ha entrado un código de
C* proyecto (que no sea igual a blancos) y, utilizando el código entrado,
C* accede al maestro de proyectos. El indicador 65 resultante se
C* activa mediante la operación CHAIN si no se encuentra el registro.
C* Si el indicador 65 está activado, o si está desactivado pero el campo
C* de estado de registro ACREC del registro contiene una D, suprimido,
C* FLAG se mueve al distintivo de error de código de proyecto, el indicador
C* de error 60 se activa, y el elemento de matriz de error 4 se
C* mueve a EMESS.
C*
C
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C
C      PRCDEX      IFNE *BLANKS
C      PRCDEX      CHAINPRJMST          65
C      *IN65       IFEQ '1'
C      *IN65       OREQ '0'
C      ACREC       ANDEQ'D'
C
C      MOVE FLAG      PRFLAG
C      MOVE '1'       *IN60
C      MOVE ERR,4     EMESS
C
C      END
C      ELSE
C*
C* La tercera comprobación determina si se ha entrado un código de
C* razón (que no sea igual a blancos) y, utilizando el código entrado,
C* accede al maestro de códigos de razón. Si registro no se encuentra,
C* la operación CHAIN activa el indicador 66 resultante. Si el indicador
C* 66 está activado, o si está desactivado pero si el campo de estado
C* de registro ACREC del registro contiene una D, suprimido, FLAG se
C* mueve al distintivo de error de código de razón, el indicador de
C* error 60 se activa y el elemento de matriz de error se mueve a EMESS.
C*
C*
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C
C      RSCDEX      IFNE *BLANKS
C      RSCDEX      CHAINRSNMST          66
C      *IN66       IFEQ '1'
C      *IN66       OREQ '0'
C      ACREC       ANDEQ'D'
C
C      MOVE FLAG      RSFLAG
C      MOVE '1'       *IN60
C      MOVE ERR,5     EMESS
C
C      END
C      END
C      END
C      END
C* La cuarta comprobación verifica si se han entrada horas en la
C* transacción del subarchivo. Si el campo de horas trabajadas es
C* igual a cero, FLAG se mueve al distintivo de error de horas

```

```

C* trabajadas, el indicador de error 60 se activa, y el elemento de
C* matriz de error 6 se mueve a EMESS.
C*
C          EHRKX      IFEQ *ZEROS
C          MOVE FLAG      HRFLAG
C          MOVE '1'       *IN60
C          MOVE ERR,6     EMESS
C          END
C*
C* Si se encuentran errores en la edición de entrada del subarchivo,
C* el indicador de error 60 se activa, y el registro de subarchivo se
C* actualiza. Esto es necesario porque hemos movido valores a uno o
C* más distintivos de error. Estos distintivos de error se visualizan
C* cuando el formato EMPCTL se graba.
C*
C          *IN60      IFEQ '1'
C          UPDATEMPFIL
C          END
C          ENDSR
C*
* ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
O*
O* La matriz en tiempo de compilación ERR se entra a continuación.
O* La matriz está precedida por "*** " para indicar el inicio de la
O* matriz, y empieza en la columna 1 de la especificación de salida.
O*
** Matriz ERR - Descripciones de errores
NO ENCONTRADO EL REGISTRO MAESTRO DE EMPLEADOS
REGISTRO MAESTRO DE EMPLEADOS NO ACTIVO
SE REQUIERE UN CODIGO DE PROYECTO O CODIGO DE RAZON
CODIGO DE PROYECTO NO VALIDO
CODIGO DE RAZON NO VALIDO
NO SE HAN INTRODUCIDO HORAS EN ESTA TRANSACCION

```

Figura 181. Programa RPG/400 de ejemplo - PRG03

13.7 Actualización semanal del archivo de tiempos

Una vez a la semana, el archivo de transacciones de entrada de tiempos **TRWEEK** se procesa para: determinar si todos los empleados incorporados al sistema de información de tiempos han entrado sus transacciones de tiempos; para actualizar los archivos maestros con las transacciones entradas; y para preparar los archivos de transacciones para el nuevo proceso semanal. La aplicación semanal consta de dos programas RPG/400 y dos procedimientos de control.

La Figura 182 muestra el Menú Principal del Sistema de Información de Tiempos. El primer paso a realizar en la actualización semanal es cambiar la fecha de final de semana en el archivo de control seleccionando la opción **3** (Entrada de transacciones del archivo de tiempos). Tras haber actualizado el archivo de control, llame a la actualización semanal seleccionando la opción **4** (Actualización semanal del archivo de tiempos). La opción **4** llama a PROC1 (Vea la Figura 183) utilizando el mandato CALL PGM(PROC1).

```

-----+-----
|
| TMENU                               Sistema de Información de Tiempos
|                                     Menú Principal
|
| 1. Mantenimiento archivo maestro           (PRG01)
| 2. Mantenimiento archivo control          (PRG02)
| 3. Entrada transacciones archivo tiempos (PRG03)
| 4. Actualización semanal archivo tiempos (PROC1)
| 5. Actualización e información mensual   (PROC3)
|
|
| 8. Visualizar mensajes                    (DSPMSG)
| 9. Fin de sesión                         (SIGNOFF)
|
|
| Selección o mandato
| ==>
| F3=Salir F4=Solicit F9=Recuperar F12=Cancelar
| F13=Soporte usuario F16=Menú principal del sistema
|
|-----+-----

```

Figura 182. Menú Principal del Sistema de Información de Tiempos

La actualización semanal del archivo de tiempos del sistema de información de tiempos consta de tres programas a nivel de control:

1. El primer programa CL, PROC1, se ejecuta de forma interactiva para determinar si todos los empleados han realizado sus entradas de tiempos. Vea la Figura 183.
2. El programa CL, PROC1, llama al programa PRG05 en RPG/400. Dentro de PRG05 se encuentra el código de operación **CALL**, de RPG/400, que llama al segundo programa CL, PROC5. Para los empleados que no hayan entrado una transacción de tiempos, PROC5 envía mensajes a sus colas de mensajes. Vea la Figura 184.
3. El programa CL, PROC1, somete el tercer programa CL, PROC2, al proceso por lotes. PROC2 produce informes semanales de transacciones de empleados, actualiza los archivos maestros y prepara los archivos de transacción para la nueva semana. Vea la Figura 185.

```

/* Actualización semanal del archivo de tiempos:          */
/* Este procedimiento se ejecuta semanalmente para procesar el */
/* archivo semanal de transacciones de entrada de tiempos. Se */
/* revisa que en el archivo no falten entradas. La persona que */
/* solicita este procedimiento y el empleado al que faltan */
/* entradas reciben un comunicado indicando que faltan entradas. */
/* A continuación, el procedimiento de actualización por lotes */
/* se somete para su proceso                                */
/*                                                         */
/* El programa PRG05 lee el archivo maestro de empleados y */
/* comprueba una entrada del archivo semanal de transacciones. */
/* Si no se encuentra ninguna entrada, el programa llama al */

```

Guía del usuario RPG/400
Actualización semanal del archivo de tiempos

```

/* procedimiento PROC5 para enviar un mensaje a los empleados. */
/* Al final del trabajo, se envía un mensaje a la persona que ha */
/* solicitado el procedimiento indicándole si se han efectuado */
/* o no todas las entradas. */
/* */
BEGIN:      PGM
            CHGJOB      SWS('00000000')
            CALL        PGM(PRG05)
            IF          COND(%SWITCH(XXXXXXXX1)) THEN(DO)
                SBMJOB   CMD(CALL PGM(PROC2)) JOB(*JOB)
            ENDDO
ENDIT:      ENDPGM

```

Figura 183. Programa PROC1 CL

```

/* Este procedimiento envía un mensaje informativo a la cola de */
/* mensajes de los empleados indicándoles que faltan sus */
/* entradas. */
/* */
/* */
PGM        PARM(&EUSRI)
DCL        VAR(&EUSRI) TYPE(*CHAR) LEN(8)
SNDUSRMSG  MSG('Faltan sus entradas de +
              la semana anterior') MSGTYPE(*INFO) TOMSGQ(&EUSRI)
MONMSG     MSGID(CPF2559)
ENDPGM

```

Figura 184. Programa PROC5 CL

```

/* Actualización semanal del archivo de tiempos */
/* Este procedimiento se ejecuta semanalmente para */
/* producir el informe semanal de transacciones de */
/* empleados y para actualizar los archivos maestros de */
/* información de tiempos. */
/* */
/* El programa PRG09 lee el archivo semanal de entradas */
/* de transacciones para producir el informe semanal */
/* y para actualizar las horas del mes a la fecha de */
/* los archivos maestros */
/* */
BEGIN:      PGM
            CALL        PGM(PRG09)
/* */
/* STEP2 añade las transacciones de entradas de tiempos */
/* de las semanas al archivo de transacciones mensual. */
/* */
STEP2:      CPYF        FROMFILE(TRWEEK) TOFILE(TRMNT) MBROPT(*ADD) +
              FMTOPT(*MAP *DROP)
/* */
/* STEP3 borra el archivo de transacciones semanal para */
/* preparar la nueva entrada de semanas. */
/* */
STEP3:      CLRPFM     FILE(TRWEEK)
/* */
ENDIT:      ENDPGM

```

Figura 185. Programa PROC2 CL

Subtemas

- 13.7.1 Programa RPG/400 de Edición de entradas del archivo de tiempos - PRG05
- 13.7.2 Diseño del informe semanal de transacciones de empleados - PRG09
- 13.7.3 Actualización del archivo maestro e Informe semanal de transacciones - PRG09

13.7.1 Programa RPG/400 de Edición de entradas del archivo de tiempos - PRG05

Este programa procesa el archivo maestro de empleados y utiliza el número de empleado para acceder al archivo semanal de entradas de transacciones para determinar si al menos existe un registro de transacciones para el empleado. Si no se encuentra ninguna entrada, el programa llama a un programa de lenguaje de control para enviar un mensaje al empleado que no ha realizado entradas.

La Figura 186 muestra el programa PRG05 incluyendo comentarios para explicar el flujo lógico, y la utilización de las funciones y códigos de operación RPG/400.

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*****
F* PRG05 - Edición de entradas del archivo de tiempos de información
F*   de tiempos
F* DESCRIPCION - Este programa edita el archivo semanal de entradas de
F*   transacciones y el maestro de empleados para determinar
F*   si todos los empleados incorporados han entrado sus
F*   transacciones semanales.
F*****
F* Este programa utiliza archivo descritos externamente. Los archivos
F* utilizados son: EMPMST - archivo maestro de empleados
F*   TRWEEKL - vista lógica del archivo semanal de
F*   entradas de transacciones por número
F*   de empleado
F*****
F* INDICADORES UTILIZADOS:
F* 50 - No se ha encontrado ningún registro en SETGT mayor que argumento
F*   búsqueda
F* 51 - Ningún registro que coincida en la operación REDPE
F* 60 - Faltan entradas de tiempos del archivo de transacciones
F* 99 - Proceso del primer ciclo
F* LR - Último registro
F*****
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF...LregLCIAIDePcLaEDispos+.....KSalidaEntrada...U1.*
FEMPMST IP E DISK
F*
F* El archivo semanal de transacciones contiene la entrada UC en las
F* posiciones 71 a 72. Esta entrada permite al programa controlar la
F* apertura y cierre de este archivo (vea el proceso del primer ciclo
F* y el proceso del último registro para obtener más detalles).
F*
```

```
FTRWEEKL IF E K DISK UC
F*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
E*
E* La matriz en tiempo de compilación que contiene los mensajes del
E* solicitador.
E*
E*...Arch(de)Arch(a)+NombreN/rN/tbLonPDSNommatLonPDSComentarios+++++
E MESS 1 2 50
E*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I*
I* La estructura de datos CTLDS procesa el área de datos del archivo
I* de control CTLFIL. El programa controla el proceso de esta área
I* de datos utilizando los códigos de operación del área de datos
I* (vea el proceso del primer ciclo y el proceso del último registro
I* para obtener más detalles).
I*
IDsnombre..NODsSal-arch+...NvecLon+...*
ICTLDS DS
I.....Campoexter.....PDesdeA++DCampo+...*
I 1 6 CTCDE
I 7 120WKEND
I 13 180CMTDT
I 19 19 CALLE
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C* PROCESO DEL PRIMER CICLO: El código siguiente sólo se procesa
C* en el primer ciclo de programa PG/400. El comprueba el indicador
C* 99, IFEQ en '0', el archivo semanal de transacciones se abre, y
C* el área de datos del archivo de control se recupera con la palabra
C* reservada *LOCK para otorgar a este programa utilización exclusiva
C* del mismo hasta que finalice el trabajo. El indicador 99 se activa
C* a continuación (igual a 1) para evitar que esta rutina se procese
```

```
C* en cualquier otro ciclo.
C*****
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          *IN99      IFEQ '0'
C          OPEN TRWEEKL
C          *NAMVAR    DEFN CTLFIL      CTLDS
C          *LOCK      IN      *NAMVAR
C          MOVE '1'      *IN99
C          END
```

C*****
 C* PROCESO PRINCIPAL: El maestro de empleados se procesa utilizando
 C* el ciclo de programa RPG/400. Para cada lectura de registro de
 C* empleados, debe haber al menos una entrada en el archivo semanal
 C* de transacciones. Si no se encuentra ningún registro en el archivo
 C* semanal de transacciones, se envía un mensaje al empleado.
 C*****

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..
 C* Mediante la utilización del número de empleado EMPNO, el archivo
 C* de transacciones se sitúa en el siguiente registro con un número
 C* de empleado mayor que registro de empleado que está procesándose.
 C* A continuación, utilizando el código de operación REDPE, se lee el
 C* registro inmediatamente anterior secuencialmente. Si el número de
 C* empleado del registro leído no coincide con el número de empleado
 C* del maestro de empleados, el indicador 51 se activa (igual a 1).
 C*

```
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          EMPNO      SETGTRCWEEL      50
C          EMPNO      REDPERCWEEL      51
```

C* El código siguiente se procesa si el indicador 51 está activado
 C* mediante la operación REDPE. Se envía un mensaje al empleado,
 C* indicándole que faltan sus entradas de tiempos. Para enviar el
 C* mensaje, el programa transfiere el control a un programa diferente
 C* que a su vez transfiere el ID de usuario al programa llamado. La
 C* operación PLIST contiene el nombre de la lista de parámetros PLIST1
 C* como factor 1 y a ésta sigue la operación PARM que especifica el
 C* campo de ID de usuario EUSRI.
 C*

C* Esta operación pudo haberse codificado colocando simplemente las
 C* entradas PARM inmediatamente después de la operación CALL, pero
 C* se codifica de esta forma para ilustrar el código de operación PLIST.
 C* El programa transfiere el control al programa PROC5, que es un
 C* programa de lenguaje de control para enviar el mensaje de empleado.
 C* Cuando este programa vuelve a recibir el control, el indicador 60 se
 C* se activa (igual a 1) para indicar que no se han efectuado todas las
 C* entradas de tiempos y que el programa PROC5 se elimina de la lista
 C* de programas activados mediante la utilización de la operación FREE.

```
C          *IN51      IFEQ '1'
C          PLIST1     PLIST
C          PARM              EUSRI
C          CALL 'PROC5'    PLIST1
C          MOVE '1'      *IN60
C          FREE 'PROC5'
C          END
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..
 C* El ciclo de programa RPG 400 activa el indicador LR de último registro
 C* una vez leído el último registro maestro de empleados. Cuando LR
 C* está activado, se comprueba el indicador de error 60, y si está
 C* activado (igual a 1), se envía el mensaje 1 a la persona que ha
 C* solicitado este programa desde la matriz de mensaje MESS, utilizando
 C* el código de operación DSPLY. Entonces esta persona entra una respuesta
 C* que se recibe en el campo de resultado MRESP. El campo del indicador de
 C* todas las entradas realizadas se actualiza a N, y el área de datos se
 C* actualiza utilizando la operación OUT. Si *IN60 está desactivado, la
 C* operación ELSE se procesa y el programa envía a esta persona el mensaje
 C* 2 desde la matriz de mensaje MESS, utilizando el código de operación
 C* DSPLY. Esta persona debe pulsar Intro para continuar el trabajo.
 C* El campo del indicador de todas las entradas realizadas se actualiza a Y
 C* y el área de datos se actualiza utilizando la operación OUT. Si el
 C* campo MREP contiene una Y, indicando que el programa continúa, el
 C* indicador externo U8 se activa (igual a 1). La operación UNLCK, en
 C* cuyo factor 2 se especifica *NAMVAR, desbloquea todas las áreas de
 C* datos del programa. A continuación se procesa la operación CLOSE
 C* para cerrar el archivo semanal de transacciones TRWEEKL.
 C*

C* Nota: Si el factor 1 de la operación OUT estuviera en blanco, el área
 C* de datos se desbloquearía como parte de la función de la
 C* operación; no obstante, se codifica la palabra reservada *LOCK
 C* de RPG para ilustrar el código de la operación UNLCK.
 C*

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

```

CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
CLR      *IN60      IFEQ '1'
CLR      MOVE MESS,1      EMESS  50
CLR      EMESS      DSPLY '*EXT'      MRESP  1
CLR      MOVE 'N'      CALLE
CLR      *LOCK      OUT *NAMVAR
CLR      ELSE
CLR      MOVE MESS,2      EMESS  50
CLR      EMESS      DSPLY '*EXT'      MRESP
CLR      MOVE 'Y'      CALLE
CLR      MOVE 'Y'      MRESP
CLR      *LOCK      OUT *NAMVAR
CLR      END
C*
CLR      MRESP      IFEQ 'Y'
CLR      MOVE '1'      *INU8
CLR      END
CLR      UNLCK*NAMVAR
CLR      CLOSETRWEEKL
O*
O*  La matriz en tiempo de compilación MESS se entra a continuación.
O*  La matriz está precedida por "*** " para indicar el inicio de la misma.
O*
** MESS - mensajes de peticionario
Faltan entradas de tiempo. "Y"-continuar "C"-cancelar
No faltan entradas de tiempo. Pulse Intro para continuar.

```

Figura 186. Programa PRG05

13.7.3 Actualización del archivo maestro e Informe semanal de transacciones - PRG09

El programa PRG09 procesa el archivo semanal de transacciones de entrada de tiempos **TRWEEK** para actualizar los archivos del maestro de empleados, del maestro de proyectos, y del maestro de códigos de razón, y para producir el informe de resumen semanal de transacciones de empleados. El programa es compatible con la SAA mediante la utilización de archivos descritos por el programa.

La Figura 188 muestra el programa PRG09 en RPG/400, en el que incluye comentarios para explicar el flujo lógico y la utilización de las diversas funciones y códigos de operación RPG/400.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*****
F* PRG09 - Actualización del archivo maestro de información tiempos
F* DESCRIPCION - Este programa actualiza los archivos maestros con las
F*   entradas semanales de transacciones y produce el
F*   informe de detalle de las transacciones semanales de
F*   empleados. Este programa es compatible con la SAA.
F*****
F* INDICADORES UTILIZADOS:
F* 40 - Encontrada la entrada en búsqueda de la tabla
F* 50 - Falta o no es válido el registro de empleado
F* 51 - Falta o no es válido el registro del código de proyecto o
F*   del código de razón
F* 69 - Salida de excepción - líneas de cabecera
F* 70 - Salida de excepción - línea de cabecera de empleado
F* 71 - Salida de excepción - actualización del maestro de proyectos
F* 72 - Salida de excepción - actualización del maestro de códigos
F*   de razón
F* 73 - Salida de excepción - línea de impresión de detalle
F* 74 - Salida de excepción - línea de totales de empleado
F* 75 - Salida de excepción - líneas de totales del informe
F* 76 - Salida de excepción - actualizar maestro de empleados
F* L1 - Nivel de control o número de empleado
F*****

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*****
F* SUBRUTINAS UTILIZADAS:
F* L1CLR - Borrado de campos de trabajo de tiempo de detalle a
F*   nivel de control
F* UPDSR - Actualización del maestro de proyecto y códigos de razón
F*   e impresión de detalle
F* TOTL1 - Actualización del maestro de empleados de tiempo
F*   total a nivel de control e impresión de tiempo total
F*****
F* Este programa utiliza los archivos descritos por el programa. Los
F*   archivos utilizados son:
F*   TRWEEK - archivo semanal de entradas de transacciones
F*   EMPMST - archivo maestro de empleados
F*   PRJMST - archivo maestro de proyectos
F*   RSNMST - archivo maestro de códigos de razón
F*   QSYSPT - archivo de impresora
F*****
FNombarchIPEAF...LregLC1AIDePcLaEDispos+....KSalidaEntradA...U1.*
FTRWEEK IP F 53 DISK
FEMPMST UF F 103 4PI 2 DISK
FPRJMST UF F 120 8AI 2 DISK
FRSNMST UF F 73 8AI 2 DISK
FQSYSPT O F 132 OF PRINTER

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
E*
E* La especificación adicional siguiente describe la tabla en tiempo
E* de compilación TABMTH. Esta tabla contiene una entrada para cada
E* mes del año con la entrada alternativa TABNAM que contiene el nombre
E* de descripción del mes. A la tabla se accede para proporcionar el
E* nombre del mes en el campo de fecha de la línea de cabecera.
E*
E...Arch(de)Arch(a)+NombreN/rN/tbLonPDSNommatLonPDSComentarios++++*
E TABMTH 1 12 2 0 TABNAM 9

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I*
I* El archivo semanal de entradas de transacciones que contiene todas
I* las entradas de tiempos se procesa por número de empleado con el
I* indicador L1 de nivel de control definido para controlar el
I* proceso en caso de cambio de número de empleado.
I*
INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
ITRWEEK NS 01

```

```

I.....PDesdea++DCampo+L1M1FrPlMnZr...*
I          P  2  50EMPNO L1
I          6  13 EUSRI
I          20 250CWKDT
I          14 190ACDAT
I          32 39 PRCDE
I          40 47 RSCDE
I          P 48 501EHWRK
I          P 51 530TFRRN
I*

```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I*
I* Al maestro de empleados se accede aleatoriamente cuando se produce
I* una ruptura de control. Se actualizan las horas de proyecto del mes
I* actual (EPHRC) y las horas de código de razón del mes actual (ENPRC).
I* El indicador identificador de registro 02 se activa si la lectura del
I* registro de empleados contiene una A en la posición 1, que indica un
I* registro activo. Si la posición 1 no contiene una A, el indicador
I* identificador de registro 03 se activa. El ciclo del programa
I* RPG/400 activa y desactiva estos indicadores.
I*

```

```

INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
IEMPMST NS 02 1 CA
I.....PDesdea++DCampo+L1M1FrPlMnZr...*
I          6  35 ENAME
I          37 41 EDEPT
I          P 82 841EPHRC
I          P 93 951ENHRC
I          NS 03
I*

```

```

I* Al maestro de proyectos se accede aleatoriamente para cada lectura
I* de transacción si el proyecto de la transacción no está en blanco.
I* Se actualizan las horas de proyecto del mes actual (PRHRC). El
I* indicador identificador de registro 04 se activa si la lectura del
I* registro de proyecto contiene una A en la posición 1, para indicar
I* un registro activo. Si en la posición 1 no aparece ninguna A, el
I* indicador identificador de registro 05 se activa. El ciclo del
I* programa RPG/400 activa y desactiva estos indicadores.
I*

```

```

IPRJMST NS 04 1 CA
I          10 59 PRDSC
I          P 107 1101PRHRC
I          NS 05
I*

```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I* Al maestro de códigos de razón se accede aleatoriamente para cada
I* transacción leída. Se actualizan las horas de código de razón del
I* mes actual (RSHRC). El indicador identificador de registro 06 se
I* activa si el registro de proyecto leído contiene una A en la posición
I* 1, indicando que es un registro activo. Si en la posición 1 no
I* aparece una A, el indicador identificador de registro 07 se activa.
I* El ciclo del programa RPG/400 activa y desactiva estos indicadores.
I*

```

```

INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
IRSNMST NS 06 1 CA
I.....PDesdea++DCampo+L1M1FrPlMnZr...*
I          10 59 RSDSC
I          P 60 631RSHRC
I          NS 07
I*

```

```

I* Las constantes con nombre siguientes definen palabras de edición
I* para el informe semanal de entrada de transacciones de empleados.
I*

```

```

I.....Constante+++++++C.....Campo+.....*
I          ' 0. -'          C          EDTHR1
I          ' 0. -'          C          EDTHR2
I*

```

```

I* El área de datos del archivo de control contiene la fecha de final
I* de semana que se utiliza en las cabeceras del informe para acceder
I* al nombre que describe al mes de la tabla TABMTH.
I*

```

```

IDsnombre..NODsSal-arch+.....NvecLon+.....*
ICTLFIL UDS
I.....Campoexter.....PDesdeA++DCampo+.....*
I          1  6 CTCDE
I          7 120WKEND
I          7  80WKMTHT
I          9 100WKDAY
I          11 120WKYR
I          13 180CMTDT
I          19 19 CALLE
I*

```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C* Proceso de primer ciclo. El código siguiente se procesa sólo
C* en el primer ciclo. El indicador 99 (*IN99) está desactivado

```

C* (igual a 0) en el primer ciclo y se procesa el código que sigue
 C* a la operación IFEQ. El programa RPG/400 recupera implícitamente
 C* la estructura de datos del área de datos CTLFIL. Mediante la
 C* utilización del campo de mes WKMTH del área de datos, se realiza
 C* la operación LOKUP para recuperar el nombre que describe el mes del
 C* archivo de tablas TABMTH. La entrada alternativa TABNAM se mueve al
 C* campo de cabecera de informe MNAME si la búsqueda se realiza
 C* satisfactoriamente (40 está activado). Si no es así, el literal UNKOWN
 C* se mueve a MNAME. Los campos de trabajo utilizados para los totales
 C* de los informes se inicializan a cero mediante la operación Z-ADD
 C* y el indicador 99 se activa para evitar que esta rutina se procese en
 C* los ciclos posteriores.
 C*

CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++

C		*IN99	IFEQ '0'		
C		WKMTH	LOKUPTABMTH	TABNAM	9 40
C	40		MOVELTABNAM	MNAME	9
C	N40		MOVELTABNAM	MNAME	
C			Z-ADD0	PRTOT	91
C			Z-ADD0	RSTOT	91
C			Z-ADD0	WKTOT	91
C			Z-ADD0	EMCNT	50
C		EMPNO	CHAINEMPST		50
C		*IN50	IFEQ '0'		
C		*IN02	ANDEQ'0'		
C			MOVE '1'	*IN50	
C			END		
C			MOVE '1'	*IN99	
C			END		

C* El ciclo del programa RPG/400 controla la lectura del archivo de
 C* transacciones y la activación del indicador de último registro LR.
 C* Se controla definiendo el archivo TRWEEK como P (primario)
 C* en la posición 16 de la especificación del archivo.

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...

C* PRINCIPAL: La rutina principal consta de tres operaciones EXSR
 C* y del proceso del último registro (LR). La primera rutina se procesa
 C* en tiempo de detalle de nivel de control. El tiempo de detalle de
 C* nivel de control tiene lugar en el ciclo inicial del programa RPG/400
 C* y en el primer registro de cada grupo de control. Es decir, cuando el
 C* número de empleado cambia, la rutina L1CLR se procesa antes de efectuarse
 C* el proceso del nuevo grupo de empleados. La segunda rutina se procesa
 C* en cada ciclo de detalle RPG/400 para acumular totales de empleado
 C* y actualizar los archivos maestros de proyectos y de códigos de razón.
 C* La tercera rutina se procesa en tiempo de totales de nivel de control.
 C* El tiempo de totales de nivel de control tiene lugar cuando se ha leído
 C* el último registro del grupo de control y en el último registro.
 C*

C* Las tres últimas líneas del código de la rutina principal se procesan
 C* sólo en el último registro. El indicador 69 se activa para saltar
 C* a la página nueva e imprimir las cabeceras, y el indicador 75 se activa
 C* para imprimir los totales del informe. La operación EXCPT se utiliza
 C* para imprimir la salida de excepción. Los indicadores se desactivan
 C* después de efectuarse la operación EXCPT para evitar que vuelva a
 C* realizarse la misma salida en operaciones EXCPT posteriores.
 C*

CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++

C	L1	EXSR L1CLR		
C	01	EXSR UPDSR		
CL1		EXSR TOTL1		
CLR		SETON		6975
CLR		EXCPT		
CLR		SETOF		6975

C*

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...

C* La subrutina L1CLR se procesa en tiempo de detalle de nivel de control.
 C* El programa borra los campos de trabajo de totales de empleado utilizando
 C* la operación Z-ADD. El programa recupera el registro maestro de empleados
 C* utilizando el número de empleado y la operación CHAIN. Si el registro
 C* no se encuentra, el indicador 50 resultante se activa. Si se encuentra
 C* el registro (*IN50 igual a 0), pero el indicador identificador de
 C* registro 02 no está activado (*IN02 igual a 0), el indicador 50 se
 C* activa. El indicador 50 controla la impresión del nombre de empleado,
 C* departamento de empleado, y la actualización del maestro de empleados
 C* en tiempo de totales. El indicador 69 se activa para saltar a la nueva
 C* página e imprimir las líneas de cabecera. El indicador 70 se activa
 C* para imprimir la línea de cabecera de empleado. La operación EXCPT se
 C* ejecuta para procesar la salida de excepción, y los indicadores
 C* 69 y 70 se desactivan.
 C*

CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++

```

C      L1CLR      BEGSR
C              Z-ADD0      PRHRS      51
C              Z-ADD0      RSHRS      51
C              Z-ADD0      WKHRS      51
C      EMPNO      CHAINEMPST      50
C      *IN50      IFEQ '0'
C      *IN02      ANDEQ'0'
C              MOVE '1'      *IN50
C              END
C              SETON      6970
C              EXCPT
C              SETOF      6970
C              ENDSR
C*
    
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

C*
 C* La subrutina UPDSR se procesa para cada registro de detalle. Primero
 C* se borra el campo de trabajo DESC moviendo blancos al mismo. Esta
 C* acción evita que se imprima la descripción de un registro anterior
 C* cuando no pueden encontrarse los registros de código de proyecto
 C* y de código de razón en los archivos maestros. Si el código de proyecto
 C* no está en blanco, se accede al maestro de proyectos utilizando las
 C* operaciones PRCDE y CHAIN. Si el registro no se encuentra, el indicador
 C* 51 resultante se activa. Si el registro se encuentra (*IN50 igual a 0)
 C* y el indicador identificador de registro 04 está activado (*IN04 igual
 C* 1), las horas de transacción se añaden a las horas de proyecto del mes
 C* actual PRHRC y al campo de trabajo de horas de proyecto de empleado PRHRS
 C* y la descripción de proyecto se mueve al campo de trabajo DESC. El
 C* indicador 71 se activa para actualizar el maestro de proyectos y el
 C* indicador 73 se activa para imprimir la línea de detalle. La operación
 C* EXCPT se ejecuta para procesar la salida de excepción y los
 C* indicadores 71 y 73 se desactivan.
 C*

CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*

```

C      UPDSR      BEGSR
C              MOVE *BLANKS      DESC
C      PRCDE      IFNE *BLANKS
C      PRCDE      CHAINPRJMST      51
C      *IN51      IFEQ '0'
C      *IN04      ANDEQ'1'
C      EHWRK      ADD PRHRC      PRHRC
C      EHWRK      ADD PRHRS      PRHRS
C              MOVE PRDSC      DESC      50
C              SETON      7173
C              EXCPT
C              SETOF      7173
C              END
C              ENDSR
C*
    
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

C*
 C* Si el código de razón no está en blanco, se accede al maestro de
 C* códigos de razón utilizando RSCDE y la operación CHAIN. Si no se
 C* encuentra el registros, el indicador 51 resultante se activa. Si el
 C* registro se encuentra (*IN50 igual a 0) y el indicador identificador
 C* de registro 06 está activado (*IN06 igual a 1), las horas de
 C* transacción se añaden a las horas de código de razón del mes actual RSHRC
 C* y al campo de trabajo de horas de código de razón de empleado RSHRS y
 C* la descripción del código de razón se mueve al campo de trabajo DESC.
 C* El indicador 72 se activa para actualizar el maestro de códigos de razón
 C* y el indicador 73 se activa para imprimir la línea de detalle. La
 C* operación EXCPT se ejecuta para procesar la salida de excepción y
 C* los indicadores 72 y 73 se desactivan.
 C*

CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*

```

C      RSCDE      IFNE *BLANKS
C      RSCDE      CHAINRSNMST      52
C      *IN52      IFEQ '0'
C      *IN06      ANDEQ'1'
C      EHWRK      ADD RSHRC      RSHRC
C      EHWRK      ADD RSHRS      RSHRS
C              MOVE RSDSC      DESC
C              SETON      7273
C              EXCPT
C              SETOF      7273
C              END
C              ENDSR
C*
    
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

C*

C* La subrutina TOTL1 se procesa en tiempo de totales de nivel de control.
 C* Los totales de campo de trabajo de empleado se añaden a los totales
 C* de campo de trabajo del informe. El indicador 74 se activa y se ejecuta
 C* la operación EXCPT para imprimir los totales de empleado, y el indicador
 C* 74 se desactiva. Si el indicador 50 está desactivado (*IN50 igual a 0,
 C* encontrado el registro de empleado), el total de horas semanales de
 C* proyecto de empleado PRHRS se añade al campo del registro maestro de
 C* empleado EPHRC, y el total de horas semanales de código de razón
 C* RSHRS se añade a ENHRC. El indicador 76 se activa y se ejecuta la
 C* operación EXCPT para actualizar el maestro de empleados y a
 C* continuación el indicador 76 se desactiva.

C*
 CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgcComentarios+++*

C	TOTL1	BEGSR		
C	PRHRS	ADD	PRTOT	PRTOT
C	RSHRS	ADD	RSTOT	RSTOT
C	PRHRS	ADD	RSHRS	WKHRS
C	WKHRS	ADD	WKTOT	WKTOT
C	EMCNT	ADD	1	EMCNT
C		SETON		74
C		EXCPT		
C		SETOF		74
C	*IN50	IFEQ	'0'	
C	PRHRS	ADD	EPHRC	EPHRC
C	RSHRS	ADD	ENHRC	ENHRC
C		SETON		76
C		EXCPT		
C		SETOF		76
C		END		
C		ENDSR		

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

O* Este programa utiliza la salida de excepción para todas sus operaciones
 O* de salida. El código siguiente describe el contenido del archivo de
 O* impresora QSYSPRT, el espaciado y el salto. Los dos primeros grupos de
 O* excepción se imprimen cuando el indicador 69 está activo. La primera
 O* excepción hace que se produzca un salto a la línea 03 de una página
 O* nueva, y la segunda excepción produce el espaciado de una línea antes
 O* de la impresión y una línea después de la impresión. La palabra
 O* reservada PAGE de RPG se utiliza para gestionar la numeración de las
 O* páginas y la palabra reservada UPDATE para imprimir la fecha del
 O* sistema.

ONombre++DFBASbSaN01N02N03Nomexc.....*

O	QSYSPRT E	03	69	
O				N01N02N03Campo+YBUposPConstante/palabraedición+...*
O				6 'PRG09'
O				61 'ENTRADA DE TRANSACCIONES'
O				67 'DEL EMPLEADO'
O				105 'PAGINA'
O				PAGE Z 110
O	E 11		69	
O				55 'PARA LA SEMANA QUE FINALIZA EL'
O				MNAME 65
O				WKDAY 68
O				72 ', 19'
O				WKYR 74
O				UPDATE Y 110

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

O* Las tres líneas de excepción siguientes las controla el indicador
 O* 70 e imprimen información de cabecera adicional. La primera línea
 O* de excepción imprime la información del empleado. Si el registro de
 O* empleado no se encuentra (el indicador 50 está activado), el nombre
 O* del empleado se sustituye por el mensaje de error. Las dos líneas de
 O* excepción siguientes imprimen cabeceras para las líneas de detalle.

ONombre++DFBASbSaN01N02N03Nomexc.....*

O	E 11		70	
O				N01N02N03Campo+YBUposPConstante/palabraedición+...*
O				20 'NUMERO DE EMPLEADO'
O				EMPNO Z 28
O				39 'NOMBRE'
O		N50		ENAME 71
O		50		66 'NUMERO DE EMPLEADO NO VALIDO'
O				88 'DEPARTAMENTO'
O		N50		EDEPT 95
O	E 1		70	
O				22 'CODIGO DE CODIGO DE'
O				40 'DESCRIPCION'
O				108 'FECHA REAL HORAS'

```
O      E 1      70
O
O      21 'PROYECTO      RAZON'
O      108 'TRABAJADA    TRABAJADAS'
O*
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

O* La línea de excepción siguiente se controla mediante el indicador 73
O* e imprime cada detalle de la transacción.

```
O*
ONombre++DFBASbSaN01N02N03Nomexc.....*
O      E 1      73
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O      PRCDE      9
O      RSCDE      23
O      DESC       79
O      ACDAT Y    94
O      EHWRK     108 EDTHR1
O*
```

O* Las tres líneas de excepción siguientes las controla en indicador
O* 74 e imprimen al producirse un cambio de número de empleado o una
O* ruptura de control.

```
O      E 2      74
O      78 'TOTALES EMPLEADO:'
O      93 'HORAS PROYECTO'
O      PRHRS     108 EDTHR1
O      E 1      74
O      91 'HORAS NO PROYECTO'
O      RSHRS     108 EDTHR1
O      E 1      74
O      98 'HORAS TOTAL SEMANA'
O      WKHRS     108 EDTHR1
O*
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

O* Las cuatro líneas de excepción siguientes las controla el indicador
O* 75 y imprimen al final del archivo o en el último registro.

```
O*
ONombre++DFBASbSaN01N02N03Nomexc.....*
O      E 2      75
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O      71 'TOTAL INFORMES:'
O      87 'HORAS PROYECTO'
O      PRTOT     108 EDTHR2
O      E 1      75
O      91 'HORAS NO PROYECTO'
O      RSTOT     108 EDTHR2
O      E 1      75
O      92 'TOTAL HORAS SEMANA'
O      WKTOT     108 EDTHR2
O      E 1      75
O      88 'CUENTA EMPLEADO'
O      EMCNT Z   108
O*
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

O* La línea de excepción siguiente la controla el indicador 71 y
O* actualiza el registro de archivos del maestro de proyectos.

```
O*
ONombre++DFBASbSaN01N02N03Nomexc.....*
OPRJMST E      71
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O      PRHRC     110P
O* La línea de excepción siguiente la controla el indicador 72 y
O* actualiza el registro de archivos del maestro de códigos de razón.
ORSNMST E      72
O      RSHRC     63P
```

O* La línea de excepción siguiente la controla el indicador 76 y
O* actualiza el registro de archivos del maestro de empleados.

```
OEMP MST E      76
O      EPHRC     84P
O      ENHRC     95P
```

* ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7

O* La tabla siguiente contiene las descripciones para cada mes
O* del año. El número de mes se utiliza como búsqueda para recuperar
O* la descripción del mes - el elemento de la tabla alternativa.
O* La tabla empieza en la columna 1 de la especificación de
O* salida.

O*

**** TABMTH - Tabla de descripción del mes**
01ENERO

02FEBRERO
03MARZO
04ABRIL
05MAYO
06JUNIO
07JULIO
08AGOSTO
09SEPTIEMBRE
10OCTUBRE
11NOVIEMBRE
12DICIEMBRE

Figura 188. Programa RPG/400 de ejemplo - PRG09

13.8 Proceso mensual

Todos los archivos maestro se procesan tras la última actualización semanal para el mes para poder producir informes mensuales, añadir valores del mes actual a los valores del año a la fecha y preparar archivos de transacciones para los nuevos procesos mensuales.

El diseño técnico de cada paso del proceso manual contiene todo o parte de lo siguiente:

- Diseño de formato de pantalla
- Descripciones de datos del archivo de pantalla
- Código del programa y comentarios
- Diagrama de espaciado de la impresora.

Subtemas

- 13.8.1 Actualización e información mensual del archivo de tiempos
- 13.8.2 Diseño del informe de resumen de la información de tiempos de empleados - PRG06RP
- 13.8.3 Descripciones de datos del informe de resumen de empleados -PRG06RP
- 13.8.4 Programa RPG/400 de informe de resumen de empleados - PRG06
- 13.8.5 Diseño del informe de resumen del proyecto de información de tiempos - PRG07RP
- 13.8.6 Descripciones de los datos del informe de resumen del proyecto - PRG07RP
- 13.8.7 Programa RPG/400 de informe de resumen del proyecto - PRG07
- 13.8.8 Diseño del informe de resumen de información de tiempos de códigos de razón - PRG08RP
- 13.8.9 Descripciones de datos del informe de resumen de códigos de razón - PRG08RP
- 13.8.10 Programa RPG/400 de informe de resumen de códigos de razón - PRG08
- 13.8.11 Programa RPG/400 de actualización y borrado mensual del archivo maestro - PRG04

13.8.1 Actualización e información mensual del archivo de tiempos

La Figura 189 muestra el Menú Principal del Sistema de Información de Tiempos. El primer paso para la actualización mensual es cambiar la fecha de final de mes en el archivo de control utilizando la opción 3. Después de haber actualizado el archivo de control, debe llamar a la actualización mensual entrando la opción 5 (Actualización e información mensual del archivo de tiempos). La opción 5 llama a PROC3, solicitando que se determine si la ejecución es para el final del año. Debe entrar una **Y** o una **N**. A continuación, PROC3 somete a PROC4 al proceso por lotes. Vea la Figura 190.

□ **CALL PGM(PROC3)**

```

-----
TMENU                               Sistema de Información de Tiempos
                                     Menú Principal

1.  Mantenimiento archivo maestro           (PRG01)
2.  Mantenimiento archivo control          (PRG02)
3.  Entrada transacciones archivo tiempos  (PRG03)
4.  Actualización semanal archivo tiempos  (PROC1)
5.  Actualización e información mensual del archivo de tiempos (PROC3)

8.  Visualizar mensajes                    (DSPMSG)
9.  Fin de sesión                          (SIGNOFF)

Selección o mandato
===>
F3=Salir  F4=Solicit  F9=Recuperar  F12=Cancelar
F13=Soporte usuario  F16=Menú principal del sistema
-----

```

Figura 189. Menú de la información de tiempos

La actualización mensual del archivo de tiempos consta de dos programa de nivel de control:

- El programa en CL, PROC3, es un programa interactivo que se solicita que especifique una **Y** o una **N** como respuesta al proceso de final de año. El programa acepta que la respuesta sea en mayúsculas o en minúsculas. A continuación, el programa somete a PROC4 al proceso por lotes.
- El programa en CL, PROC4, es un trabajo por lotes que produce informes de resumen mensuales de empleados, proyectos, y códigos de razón. También prepara los archivos maestros y los archivos de transacciones para el nuevo proceso mensual (y para el nuevo año, si se ha solicitado el proceso de final de año).

```

/* Actualización e información mensual del archivo de tiempos: */
/* Este procedimiento es el primer paso para la actualización */
/* mensual de la información de tiempos. El programa envía un */
/* mensaje solicitando que se determine si esta ejecución es */
/* el final del año. Si la ejecución es para el final del año, */
/* el mandato CHGJOB activa el conmutador de trabajo 4. El */
/* programa de actualización PROC4 se somete entonces al proceso */
/* por lotes. */
/* */
BEGIN:      PGM
            DCL                &REPLY *CHAR LEN(1)
            SNDUSRMSG MSG('Actualizar para final del año Y o N') +
                MSGRPY(&REPLY)
            IF COND(&REPLY *EQ Y) THEN(DO)
            SBMJOB CMD(CALL PGM(PROC4)) JOB(PROC4) SWS(00010000)
            ENDDO
            ELSE
            IF COND(&REPLY *NE Y) THEN(DO)
            SBMJOB CMD(CALL PGM(PROC4)) JOB(PROC4) SWS(00000000)

```

```

                ENDDO
/*
ENDIT:         ENDPGM
                */

```

Figura 190. Programa PROC3 CL

```

/* Actualización e información mensual del archivo de tiempos: */
/* Este procedimiento se ejecuta semanalmente para producir los */
/* informes de empleados, proyectos y códigos de razón y para */
/* preparar los archivos maestros y archivos de transacciones */
/* para el proceso del nuevo mes. */
/*
/* El programa PRG06 lee el archivo de transacciones mensual */
/* para producir el informe de entrada de tiempos de empleados. */
/*
BEGIN:         PGM
              RTVJOBA
              CALL          PGM (PRG06)
/*
/* El programa PRG07 lee el archivo de transacciones mensual */
/* para producir el informe de entrada de tiempos de proyectos. */
/*
PRG07:        CALL          PGM (PRG07)
/*
/* El programa PRG08 lee el archivo de transacciones mensual */
/* producir el informe de entrada de tiempos de códigos de */
/* razón. */
/*
PRG08:        CALL          PGM (PRG08)
/*
/* En este paso se añaden las horas del mes actual a las horas */
/* del año a la fecha y se borra el campo del mes a la fecha. Si */
/* ésta es una ejecución de final de año, la fecha de año a la */
/* fecha se incorpora al año a la fecha anterior y el año a la */
/* se borra. El paso se repite en bucle tres veces. Cada vez que */
/* se llama al programa PRG04, el archivo abierto y actualizado */
/* se controla mediante el indicador externo activado por el */
/* mandato CHGJOB. */
/* U1 - Maestro de empleados */
/* U2 - Maestro de proyectos */
/* U3 - Maestro de códigos de razón */
/*
              CHGJOB        SWS (100X0000)
PRG04:        CALL          PGM (PRG04)
              IF            COND (%SWITCH (100X0000)) THEN (DO)
                  CHGJOB SWS ('010X0000')
                  GOTO      CMDLBL (PRG04)
                  ENDDO
              ELSE
              IF            COND (%SWITCH (010X0000)) THEN (DO)
                  CHGJOB SWS ('001X0000')
                  GOTO      CMDLBL (PRG04)
                  ENDDO
/*
/* El paso CLEAR borra el archivo mensual de transacciones para */
/* preparar la actividad del nuevo mes. */
/*
CLEAR:        CLRPFM        FILE (TRMNTH)
/*
ENDIT:        ENDPGM

```

Figura 191. Programa PROC4 CL

13.8.3 Descripciones de datos del informe de resumen de empleados -PRG06RP

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A* Este archivo de impresora describe el formato para el informe de
A* resumen de información de tiempos de empleados. En este archivo de
A* impresora existen cuatro formatos de registro, identificados por
A* una R en la posición 17 seguida de el nombre de formato en las
A* posiciones 19 a 20. Se utilizan las palabras clave siguientes:
A* EDTCDE(a) - Edita los campos numéricos con posibilidad de salida.
A* PAGNBR - Especifica un campo decimal con zona, de cuatro
A* dígitos, que contiene el número de página.
A* REF(REFMST) - Las líneas que contienen una R en la posición 29
A* utilizan los atributos de un campo definido
A* anteriormente en este campo de referencia.
A* SKIPB(n) - Especifica que el dispositivo de impresora va a
A* saltar a una línea específica antes de imprimir
A* la línea siguiente.
A* SPACEA(n) - Especifica que el dispositivo de impresora va a dejar
A* espacios de (n) líneas tras imprimir una o más líneas.
A* SPACEB(n) - Especifica que el dispositivo de impresora va a dejar
A* espacios de (n) líneas antes de imprimir la línea o
A* líneas siguientes.
```

```
A*****
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++++
A
```

REF(REFMST)

```
A*
A* El primer formato, TITLE1, contiene la definición de las
A* líneas de cabecera del informe. El formato se graba en el
A* primer ciclo, cuando se produce un cambio de número de empleado,
A* o cuando se produce un desbordamiento durante la impresión de
A* los detalles de un empleado.
A*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++++
```

```
A          R TITLE1                                SKIPB(3)
A          2'PRG06RP'
A          37'RESUMEN DEL INFORME DE TIEMPOS DE EMPLEADOS'
A          90'PAGINA'
A          95PAGNBR
A          SPACEA(1)
A          38'PARA EL PERIODO QUE FINALIZA'
A          CMTDT          R          59EDTCDE(Y)
A          RDATE          6S 0      91EDTCDE(Y)
A          SPACEA(2)
A          2'NUMERO DE EMPLEADO'
A          EMPNO          R          19EDTCDE(Z)
A          31'NOMBRE DEL EMPLEADO'
A N60          ENAME          R          46
A 60          46'NUMERO DE EMPLEADO NO VALIDO'
A          82'DEPARTAMENTO'
A N60          EDEPT          R          94
A          SPACEA(2)
A          2'CODIGO DE CODIGO DE'
A          26'DESCRIPCION'
A          78'FECHA REAL          HORAS'
A          SPACEA(1)
A          3'PROYECTO          RAZON'
A          81'TRABAJADA          TRABAJADAS'
A          SPACEA(1)
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*
A* El segundo formato, DETAIL, contiene la definición de las líneas de
A* impresión de detalle. El formato se graba para cada registro de
A* detalle del archivo de transacciones mensual.
A*
```

```
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++++
A
```

```
A          R DETAIL
A          PRCDE          R          2
A          RSCDE          R          14
A N61N62          RDESC          50      26
A 61          26'CODIGO PROYECTO NO VALIDO'
A 62          26'CODIGO RAZON NO VALIDO'
A          CWKDTX          R          79REFFLD(CWKDT)
A          EDTCDE(Y)
A          EHWRK          R          93EDTCDE(L)
A          SPACEA(1)
```

```
A*
A* El tercer formato, TOTL1, contiene la definición de la interrupción
A* de nivel L1 de los tiempos totales. El formato se graba cuando se
A* cambia la fecha de final de semana o cuando se cambia el número
```

```

A* de empleado.
A*
A      R TOTL1                SPACEB(1)
A      71'TOTAL HORAS SEMANA'
A      WKTOT                5S 1  93EDTCDE(L)
A      SPACEA(2)

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*
A* El cuarto formato, TOTL2, contiene la definición de la interrupción
A* de nivel L2 de tiempos totales. El formato se graba cuando se cambia
A* el número de empleado.
A*
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++*****
A      R TOTL2
A      44'MES                % DE '
A      67'AÑO A              % DE '
A      SPACEA(1)
A      45'ACTUAL            TOTAL'
A      69'LA FECHA          TOTAL'
A      SPACEA(1)
A      2'RESUMEN EMPLEADO'
A      23'HORAS DEL PROYECTO'
A      PRMTH                5S 1  44EDTCDE(L)
A      PCMTH                5S 1  55EDTCDE(L)
A      PRYER                7S 1  66EDTCDE(L)
A      PCYER                5S 1  79EDTCDE(L)
A      SPACEA(1)
A      23'HORAS NO DE PROYECTO'
A      NPMTH                5S 1  44EDTCDE(L)
A      NCMTH                5S 1  55EDTCDE(L)
A      NPYER                7S 1  66EDTCDE(L)
A      NCYER                5S 1  79EDTCDE(L)
A      SPACEA(1)
A      23'HORAS TOTALES'
A      TOMTH                5S 1  44EDTCDE(L)
A      TCMTH                5S 1  55EDTCDE(L)
A      TOYER                7S 1  66EDTCDE(L)
A      TCYER                5S 1  79EDTCDE(L)
A      SPACEA(1)

```

Figura 193. Descripciones de datos del informe de resumen de empleados - PRG06RP

13.8.4 Programa RPG/400 de informe de resumen de empleados - PRG06

La Figura 194 muestra el programa PRG06 en RPG/400 incluyendo comentarios para explicar el flujo lógico y la utilización de las diversas funciones y códigos de operación RPG.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*****
F* PRG06 - Informe de resumen de información de tiempos de empleados
F* DESCRIPCION - Este programa produce el informe de resumen de
F* información de tiempos de empleados. Todas las entradas
F* de tiempo del mes se imprimen por fecha de final de
F* semana con subtotales por semana y un resumen de empleados
F* que muestra los totales de mes y de año a la fecha.
F*****
F* Este programa utiliza archivos descritos externamente. Los
F* archivos utilizados son:
F* TRMNTHL - Visión lógica de TRMNTH, archivo de transacciones mensual
F* por número de empleado y fecha de final de semana.
F* EMPMST - Archivo maestro de empleados
F* PRJMST - Archivo maestro de proyectos
F* RSNMST - Archivo maestro de códigos de razón
F* PRG06RP - Archivo del informe de resumen de empleados
F*****
F* INDICADORES UTILIZADOS:
F* 60 - No encontrado el registro maestro de empleados
F* 61 - No encontrado el registro maestro de proyectos
F* 62 - No encontrado el registro maestro de códigos de razón
F* 99 - Proceso del primer ciclo
F* L1 - Nivel de control en fecha de final de semana
F* L2 - Nivel de control en número de empleado
F*****
F* SUBROUTINAS UTILIZADAS:
F* DTLR - Rutina de cálculos de detalle
F* LCHK - Rutina de comprobación de recuento de líneas
F* L2CLR - Borrar campos de trabajo en tiempo de detalle L2
F* SUBRL1 - Cálculos de tiempos totales - cambio de fecha de final
F* de semana
F* SUBRL2 - Cálculos de tiempos totales - cambio de número de empleado

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF....LregLC1AIDePcLaEDispos+.....KSalidaEntradA....U1.*
FTRMNTHL IP E K DISK
FEMPMST IF E K DISK
FPRJMST IF E K DISK
FRSNMST IF E K DISK
FPRG06RP O E PRINTER

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
E* Las matrices siguientes se utilizan para almacenar las horas de
E* semanales de proyecto y de código de razón del empleado. Cada matriz
E* contiene hasta cinco totales semanales.
E*
E...Arch(de)Arch(a)+NombreN/rN/tbLonPDSNommatLonPDSComentarios++++*
E ARR 5 5 1
E ARR 5 5 1

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I* El código siguiente cambia el nombre de los nombres de campo de
I* entrada del archivo de transacciones mensual. Estos campos aparecen
I* en otras definiciones de datos y se recubren cuando se leen dichos
I* archivos; estos nombres evitan el recubrimiento.
I*
INomreg++...In.....*
IRCMNTH
I.....Campoexter.....Campo+L1C1..PoNeCe...*
I EMPNO EMPNOXL2
I CWKDT CWKDTXL1
I CMTDT CMTDTX
I*
I* Area de datos del archivo de control descrita externamente
I*
IDsnombre..NODsSal-arch++.....NvecLon+.....*
ICTLFIL EUDS
I*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* PROCESO DEL PRIMER CICLO: El indicador 99 se desactiva (igual a 0)
C* en el primer ciclo del programa RPG/400 y la rutina se procesa.
C* La operación TIME recupera la hora del día y la fecha del sistema
C* y las sitúa en el campo de resultado TDATE. La hora del día ocupa
C* las seis primeras posiciones y la fecha del sistema las seis últimas

```

C* posiciones de TDATE. La operación MOVE mueve las últimas seis posiciones
 C* al campo de resultado RDATE para proporcionar la fecha de ejecución
 C* del informe. Para obtener el mismo resultado, pudo haberse especificado
 C* la palabra reservada UDATE de RPG en las especificaciones de salida.
 C* El indicador 99 se activa entonces (igual a 1) para evitar que esta
 C* rutina se procese en los ciclos siguientes.

C*

CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++

```
C      *IN99      IFEQ '0'
C
C      TIME      TDATE 120
C      MOVE TDATE RDATE 60
C      MOVE '1'   *IN99
C      END
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

C*

C* RUTINA PRINCIPAL: La rutina principal consta de cuatro subrutinas
 C* EXSR. La primera subrutina se procesa en tiempo de detalle cuando
 C* el indicador de nivel de control L2 está activado. Esto tiene lugar
 C* en el primer ciclo del programa RPG/400 y en el ciclo del programa
 C* RPG/400 que sigue a los cálculos de tiempos totales. La subrutina
 C* L2CLR borra los campos de trabajo y graba las cabeceras del informe.
 C* La segunda subrutina, DTLSR, se procesa en cada ciclo de tiempos de
 C* detalle RPG/400. La rutina graba líneas de informe de detalle y
 C* acumula datos para la impresión de los tiempos totales. Las subrutinas
 C* tercera y cuarta se procesan en tiempo de totales. La subrutina SUBRL1
 C* se procesa cuando se produce un cambio de fecha de final de semana y
 C* también cuando se produce un cambio de número de empleado (la lógica
 C* del RPG/400 activa todos los indicadores de control de nivel inferior
 C* cuando se produce una interrupción de control, es decir, cuando
 C* L2 se activa, y L1 también lo hace). La subrutina SUBRL2 se
 C* procesa cuando se produce un cambio de número de empleado.

C*

CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++

```
C      L2      EXSR L2CLR
C      EXSR DTLSR
C      CL1     EXSR SUBRL1
C      CL2     EXSR SUBRL2
```

C*****

C* SUBRUTINA DTLSR: Esta rutina realiza operaciones en tiempo de detalle.
 C* Los indicadores de error *IN61 y *IN62 se desactivan (igual a 0) como
 C* parte la preparación. Si el código de proyecto PRCDE no está en blanco,
 C* las horas trabajadas se añaden al elemento actual de la matriz de
 C* proyecto. La matriz incrementa cada vez que cambia la fecha de final de
 C* semana, y se restaura a 1 (el primer elemento) cuando cambia el número
 C* de empleado. Al archivo maestro de empleados se accede utilizando
 C* la operación CHAIN. Si no se encuentra el registro, el indicador 61
 C* se activa. Si el registro se encuentra, la descripción del proyecto
 C* se mueve al campo de trabajo RDESC.

C*

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++

```
C      DTLSR      BEGSR
C      MOVE '0'   *IN61
C      MOVE '0'   *IN62
C      PRCDE      IFNE *BLANKS
C      EHWRK      ADD  ARRP,P   ARRP,P
C      PRCDE      CHAINPRJMST   61
C      *IN61      IFEQ '0'
C      MOVE PRDSC RDESC
C      END
C      ELSE
```

C* La sentencia ELSE anterior indica el final de las operaciones de código
 C* del proyecto. Si el código de proyecto está en blanco, debe existir
 C* un código de razón. Las horas trabajadas se añaden al elemento actual
 C* de la matriz de horas no del proyecto y al archivo maestro de códigos
 C* de razón se accede utilizando la operación CHAIN. Si no se encuentra
 C* el registro, el indicador 62 se activa. Si se encuentra el registro,
 C* la descripción del código de razón se mueve al campo de trabajo RDESC.

C*

```
C      EHWRK      ADD  ARRN,N   ARRN,N
C      RSCDE      CHAINRSNMST   62
C      *IN62      IFEQ '0'
C      MOVE RSDSC RDESC
C      END
C      END
```

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

C*

C* La sentencia END anterior indica el final del IF original de esta
 C* rutina. El registro de detalle se ha procesado hora y el programa
 C* está preparado para grabar la línea de informe de detalle. La

C* sentencia WRITE graba el formato de registro DETAIL en el archivo
 C* de impresora PRG06RP descrito externamente. El formato contiene una
 C* línea y el recuento de líneas se incrementa en uno. Cada vez que se
 C* realiza una operación de salida al archivo de impresora, se procesa
 C* la subrutina LICHK. Esta rutina determina si debe efectuarse el
 C* proceso de desbordamiento de página.

C*
 CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++

C **WRITEDETAIL**
 C **ADD 1 LICNT**
 C **EXSR LICHK**
 C **ENDSR**

C*
 C*****
 C* SUBROUTINA SUBRL1: Esta rutina realiza operaciones de tiempos totales.
 C* Se añaden las horas de proyecto y las horas no de proyecto para
 C* proporcionar las horas totales de la semana utilizando el elemento
 C* actual para cada matriz. Se comprueba el valor del recuento de líneas.
 C* Si es mayor o igual a 59, se establece en 60 y se procesa la rutina de
 C* desbordamiento LICHK. El programa realiza estas operaciones
 C* para asegurar que existen suficientes líneas de impresión en la
 C* página para imprimir la línea de totales de la semana. Entonces se graba
 C* el formato de impresión TOTL1. Los índices de la matriz de proyecto y
 C* no de proyecto incrementan en uno para las horas de la semana siguiente,
 C* y el recuento de líneas incrementa en dos para las líneas de
 C* impresión grabadas en formato TOTL1.
 C*

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..
 CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++

C **SUBRL1 BEGSR**
 C **ARRP,P ADD ARRN,N WKTOT**
 C **LICNT IFGE 59**
 C **Z-ADD60 LICNT**
 C **EXSR LICHK**
 C **END**
 C **WRITETOTL1**
 C **ADD 1 P**
 C **ADD 1 N**
 C **ADD 2 LICNT**
 C **EXSR LICHK**
 C **ENDSR**

C*
 C*****
 C* SUBROUTINA SUBRL2: Esta subrutina realiza operaciones de tiempos
 C* totales. Se comprueba el valor del recuento de líneas. Si es mayor o
 C* igual a 55, se establece en 60 y se procesa la rutina de desbordamiento
 C* LICHK. El programa realiza estas operaciones para asegurar que
 C* existen suficientes línea de impresión en la página para imprimir las
 C* líneas totales del empleado. Las matrices de proyecto y no de proyecto
 C* se suman utilizando la operación XFOOT. Esta operación junta todos
 C* los elementos de la matriz y sitúa la suma en el campo de resultado.
 C* La serie de operaciones aritméticas que siguen preparan la línea de
 C* totales del empleado para la impresión. Las horas de proyecto totales
 C* se añaden a las horas no de proyecto totales, el resultado es igual
 C* a las horas totales del mes. El porcentaje de horas de proyecto del
 C* total se calcula dividiendo las horas de proyecto totales por las horas
 C* totales y multiplicando el resultado (WRK1) por 100. El porcentaje de
 C* horas no de proyecto del total se calcula dividiendo las horas no de
 C* proyecto totales por las horas totales y multiplicando el resultado
 C* (WRK1) por 100. El campo de totales de porcentaje se establece a 100
 C* utilizando la operación Z-ADD.
 C*

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..
 CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++

C **SUBRL2 BEGSR**
 C **LICNT IFGE 55**
 C **Z-ADD60 LICNT**
 C **EXSR LICHK**
 C **END**
 C **XFOOTARRP PRMTH**
 C **XFOOTARRN NPMTH**
 C **PRMTH ADD NPMTH TOMTH**
 C **PRMTH DIV TOMTH WRK1 53H**
 C **WRK1 MULT 100 PCMTH**
 C **NPMTH DIV TOMTH WRK1 H**
 C **WRK1 MULT 100 NCMTH**
 C **Z-ADD100 TCMTH**

C* Los cálculos siguientes añaden el mes actual a los totales del año
 C* a la fecha y realizan las mismas expresiones que para el mes actual.

C **PRMTH ADD EPHRY PRYER**

```

C          NPMTH      ADD  EPNRY      NPYER
C          PRYER      ADD  NPYER      TOYER
C          PRYER      DIV  TOYER      WRK1      H
C          WRK1       MULT 100        PCYER
C          NPYER     DIV  TOYER      WRK1      H
C          WRK1       MULT 100        NCYER
C          Z-ADD100   TCYER
C*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* La línea de totales del empleado está ahora preparada para la impresión.
C* Se graba el formato TOTL2. Dado que el programa realizará operaciones
C* L2 de tiempo de detalle en el siguiente ciclo para prepararse para el
C* siguiente empleado, el campo de recuento de líneas no incrementa y
C* el desbordamiento no se comprueba después de grabarse el formato.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgcComentarios+++
C          WRITETOTL2
C          ENDSR
C*
C*****
C* SUBROUTINA LICHK: Esta rutina controla el desbordamiento de página.
C* Si el recuento de líneas es mayor o igual a 60, se graba el formato
C* de cabecera TITLE1 y el recuento de líneas se establece en 9.
C*
C          LICHK      BEGSR
C          LICNT      IFGE 60
C          WRITETITLE1
C          Z-ADD9          LICNT
C          END
C          ENDSR

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* SUBROUTINA L2CLR: Esta subrutina prepara los campos de trabajo e
C* imprime las cabeceras de línea antes de procesar el primer registro
C* de detalle del empleado. La matriz de horas de proyecto ARRP y la
C* matriz de horas no de proyecto ARRn se inicializan a 0. Los elementos
C* de matriz se establecen a continuación en 1 para la primera aparición.
C* Al archivo maestro de empleados se accede utilizando el número de
C* empleado del registro de transacciones. Si el registro de empleados
C* no se encuentra, el indicador 60 se activa. Las cabeceras del informe se
C* imprimen grabando el formato de impresión TITLE1, y el recuento
C* de líneas se establece en 9.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgcComentarios+++
C          L2CLR      BEGSR
C          Z-ADD0          ARRP
C          Z-ADD0          ARRn
C          Z-ADD1          P          10
C          Z-ADD1          N          10
C          EMPNOX      CHAINEMPMST          60
C          WRITETITLE1
C          Z-ADD9          LICNT      30
C          ENDSR

```

Figura 194. Programa RPG/400 de ejemplo - PRG06

13.8.5 Diseño del informe de resumen del proyecto de información de tiempos - PRG07RP

El informe Resumen del proyecto de información de tiempos lista todas las transacciones de entrada cargadas para un código de proyecto. El informe está organizado por número de empleado dentro de la fecha de final de semana, dentro del código de proyecto. Los subtotales se imprimen cuando se produce un cambio en la fecha de final de semana y un resumen de proyecto se imprime cuando se produce un cambio de código de proyecto.

La Figura 195 muestra el Informe de resumen del proyecto de información de tiempos. Los campos alfanuméricos que se definen en el informe se representan por una serie de letras A, los campos numéricos ser representan por una serie de números 9, y las fechas se representan mediante **MM/DD/AA**. Vea la Figura 196 en el tema 13.8.6 para obtener las especificaciones de descripción de datos para el informe PRG07RP.

```

-----
|0000000001111111122222222333333333344444444455555556666666777777788888889999999|
|12345678901234567890123456789012345678901234567890123456789012345678901234567890|
-----
|001|
|002|
|003| PRG07RP                RESUMEN DE LA INFORMACION DE TIEMPOS DEL PROYECTO                PAGINA 9999
|004|                               PARA EL PERIODO QUE FINALIZA EL MM/DD/AA                               MM/DD/AA
|005|
|006| CODIGO DEL PROYECTO AAAAAAAA  DESCRIPCION  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
|007|
|008| RESPONSABILIDAD                FECHA DE          FECHA DE FINAL   FECHA DE          TOTAL HORAS
|009|                               COMIENZO          ESTIMADA         FINAL            ESTIMADAS
|010| AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  MM/DD/AA        MM/DD/AA        MM/DD/AA        99999999,9-
|011|
|012| NUMERO DE          NOMBRE DEL EMPLEADO                FECHA DE FINAL   HORAS
|013| EMPLEADO                DE LA SEMANA          TRABAJADAS
|014| 999999  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  MM/DD/AA        9999,9-
|015| 999999  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  MM/DD/AA        9999,9-
|016| 999999  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  MM/DD/AA        9999,9-
|017|
|018|                               TOTAL SEMANAL  999999,9-
|019|
|020| 999999  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  MM/DD/AA        9999,9-
|021| 999999  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  MM/DD/AA        9999,9-
|022|
|023|                               TOTAL SEMANAL  999999,9-
|024|
|025| RESUMEN DEL PROYECTO:  MES          AÑO ACTUAL          TOTAL DEL          TOTAL HORAS          % VARIACION SOB
|026|                   ACTUAL          A LA FECHA          AÑO ANTERIOR        DEL PROYECTO        LAS HORAS ESTIMA
|027|                   999999,9-  99999999,9-  99999999,9-  99999999,9-  9999.9-
|028|
-----

```

Figura 195. Diseño del informe de resumen del proyecto de información de tiempos - PRG07RP

13.8.6 Descripciones de los datos del informe de resumen del proyecto - PRG07RP

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A* Este archivo de impresión describe el formato del informe de resumen
A* mensual de la información de tiempos. En este archivo de impresora
A* hay cuatro formatos de registro, identificados por una R en la posición
A* 17 seguida del nombre de formato en las posiciones 19 a 20. Se utilizan
A* la palabras clave siguientes:
A* DATE - Especifica la fecha del sistema
A* EDTCDE(a) - Edita los campos numéricos con posibilidad de salida
A* PAGNBR - Especifica un campo decimal con zona, de cuatro dígitos
A* para contener el número de página.
A* REF(REFMST) - Cualquier línea que contenga una R en la posición 29
A* utiliza los atributos de un campo definido anteriormente
A* en este archivo de referencia.
A* REFFLD - Para un campo que hace referencia a un campo definido
A* anteriormente.
A* SKIPB(n) - Especifica que el dispositivo de impresora va a
A* saltar a una línea específica antes de imprimir
A* la línea siguiente.
A* SPACEA(n) - Especifica que el dispositivo de impresora va a dejar
A* espacios de n líneas tras imprimir una o más líneas.
A* SPACEB(n) - Especifica que el dispositivo de impresora va a dejar
A* espacios de n líneas antes de imprimir la línea o
A* líneas siguientes.
A*****
AAN01N02N03T.Nombre+++RLon++TDpBLínPosFunciones+++++++*
A REF(REFMST)
A*
A* El primer formato, TITLE1, contiene la definición de las
A* líneas de cabecera del informe. El formato se graba en el
A* primer ciclo, cuando se produce un cambio de número de proyecto,
A* o cuando se produce un desbordamiento durante la impresión de
A* los detalles de un código de proyecto.
A*
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre+++RLon++TDpBLínPosFunciones+++++++*
A R TITLE1 SKIPB(3)
A 2'PRG07RP'
A 34'RESUMEN DE LA INFORMACION DE
A TIEMPOS DEL PROYECTO'
A 88'PAGINA'
A 93PAGNBR
A SPACEA(1)
A 35'PARA EL PERIODO FINALIZADO'
A CMTDT R 56EDTCDE(Y)
A 89DATE EDTCDE(Y)
A SPACEA(2)
A 4'CODIGO DEL PROYECTO'
A PRCDEX R 18REFFLD(PRCDE)
A 35'DESCRIPCION'
A PRDSC R 48
A SPACEA(2)
A 4'RESPONSABILIDAD'
A 41'FECHA DE ESTIMADA'
A 69'FECHA TOTAL HORAS'
A SPACEA(1)
A 42'COMIENZO ESTIMADA'
A 72'DE FINAL ESTIMADAS'
A SPACEA(1)
A N60 PRRSP R 4
A N60 PRSTR R 40EDTCDE(Y)
A N60 PREND R 55EDTCDE(Y)
A N60 PRCMP R 70EDTCDE(Y)
A N60 PREST R 86EDTCDE(L)
A SPACEA(2)
A 4'NUMERO DE NOMBRE DE EMPLEADO'
A 54'FECHA DE FINAL HORAS'
A SPACEA(1)
A 5'EMPLEADO'
A 57'DE LA SEMANA TRABAJADAS'
A SPACEA(1)
```

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*
A* El segundo formato, DETAIL, contiene la definición de las líneas de
A* impresión de detalle. El formato se graba para cada registro de
A* detalle en el archivo de transacciones mensual.
A*
AAN01N02N03T.Nombre+++RLon++TDpBLínPosFunciones+++++++*
A R DETAIL
```

Descripciones de los datos del informe de resumen del proyecto - PRG07RP

```

A      EMPNO      R      5EDTCDE (Z)
A N61  ENAME      R      18
A 61   CWKDTX     R      18'NUMERO DE EMPLEADO NO VALIDO'
A      CWKDTX     R      55REFFLD (CWKDT)
A      EDTCDE (Y)
A      EHWRK     R      71EDTCDE (L)
A      SPACEA (1)
A*
A* El tercer formato, TOTL1, contiene la definición de la interrupción
A* de nivel L1 de los tiempos totales. El formato se graba cuando se
A* produce un cambio de fecha de final de semana o un cambio de
A* código de proyecto.
A*
A      R TOTL1    SPACEB (1)
A      54'TOTAL SEMANAL'
A      WKTOT      7S 1  69EDTCDE (L)
A      SPACEA (2)

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*
A* El cuarto formato, TOTL2, contiene la definición de la interrupción
A* de nivel L2 de tiempos totales. El formato se graba cuando se produce
A* un cambio de código de proyecto.
A*
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones++++*****
A      R TOTL2    SPACEB (1)
A      2'RESUMEN DEL PROYECTO:'
A      20'MES      AÑO ACTUAL'
A      49'TOTAL DEL  TOTAL HORAS'
A      82'% VARIACION SOBRE'
A      SPACEA (1)
A      21'ACTUAL   A LA FECHA'
A      52'AÑO ANTERIOR DEL PROYECTO'
A      81'A LAS HORAS ESTIMADAS'
A      SPACEA (1)
A      PRMTH      7S 1  19EDTCDE (L)
A      PRYER      9S 1  32EDTCDE (L)
A      PRHRP     R      49EDTCDE (L)
A      PRTOT      9S 1  65EDTCDE (L)
A      PRVAR      5S 1  86EDTCDE (L)

```

Figura 196. Descripciones de los datos del informe de resumen del proyecto - PRG07RP

13.8.7 Programa RPG/400 de informe de resumen del proyecto - PRG07

La Figura 197 muestra el programa RPG/400 incluyendo comentarios para explicar el flujo lógico y la utilización de las diversas funciones y códigos de operación RPG/400.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*****
F* PRG07 - Informe de resumen de la información de tiempos del proyecto
F* DESCRIPCION - Este programa produce el informe de resumen mensual de
F* la información de tiempos del proyecto. Todas las entradas
F* de tiempo del mes se imprimen por número de empleado,
F* dentro de la fecha de final de semana, dentro del código
F* de proyecto. Los subtotales se imprimen por semana y
F* si se produce un cambio de código de proyecto, se
F* imprime un resumen del proyecto.
F*****
F* Este programa utiliza archivo descritos externamente. Los archivos
F* utilizados son:
F* TRMNTHL - visión lógica de TRMNTH, archivo de transacciones
F* mensual por código de proyecto, número de empleado
F* y fecha de final de semana.
F* EMPMST - archivo maestro de empleados
F* PRJMST - archivo maestro de proyectos
F* PRG07RP - archivo del informe resumen del proyecto
F*****
F* INDICADORES UTILIZADOS:
F* 60 - No encontrado el registro maestro de proyectos
F* 61 - No encontrado el registro maestro de empleados
F* L1 - Nivel de control en fecha de final de semana
F* L2 - Nivel de control en código de proyecto
F*****
F* SUBROUTINAS UTILIZADAS:
F* DTLSR - Rutina de cálculos de detalle
F* LICHK - Rutina de comprobación de recuento de líneas
F* L2CLR - Borrar campos de trabajo en tiempo de detalle L2
F* SUBRL1 - Cálculos de tiempos totales - cambio de fecha de final
F* de semana
F* SUBRL2 - Cálculos de tiempos totales - cambio de código de proyecto
F*****

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF...LregLCIAIDePcLaEDispos+.....KSalidaEntradA...U1.*
FTRMNTHN IP E K DISK
FEMPMST IF E K DISK
FPRJMST IF E K DISK
FPRG07RP O E PRINTER

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
E*
E* La matriz siguiente se utiliza para almacenar las horas semanales
E* del código de proyecto. La matriz contiene hasta cinco totales
E* semanales.
E*
E...Arch(de)Arch(a)+NombreN/rN/tbLonPDSNommatLonPDSComentarios+++++
E ARR 5 5 1

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I*
I* El código siguiente vuelve a asignar un nombre a los nombres de
I* campo de entrada del archivo de transacciones mensual. Estos campos
I* aparecen en otras definiciones de datos, y se recubren cuando se leen
I* dichos archivos, estas nuevos nombres de códigos evitan el recubrimiento.
INomreg++...In.....*
IRCMNTH
I.....Campoexter.....Campo+L1C1..PoNeCe...*
I PRCDE PRCDEXL2
I CWKDT CWKDTXL1
I CMTDT CMTDTX
I* Area de datos del archivo de control descrita externamente
I*
IDSnombre..NODsSal-arch++.....NvecLon+.....*
ICTLFIL EUDS
I*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C* RUTINA PRINCIPAL: La rutina principal consta de cuatro subrutinas
C* EXSR. La primera subrutina se procesa en tiempo de detalle cuando
C* el indicador de nivel de control L2 está activado. Esto tiene lugar
C* en el primer ciclo del programa RPG/400 y en el ciclo del programa
C* RPG/400 que sigue a los cálculos de tiempos totales. La subrutina L2CLR
C* borran los campos de trabajo y graba las cabeceras del informe. La

```

C* segunda subrutina, DTLSR, se procesa en cada ciclo de tiempo de detalle
 C* RPG/400. La rutina graba las líneas del informe de detalle y acumula
 C* datos para la impresión de tiempos totales. Las subrutinas tercera y
 C* cuarta se procesa en el tiempo de totales. La subrutina SUBRL1 se procesa
 C* cuando se produce un cambio de fecha de final de semana y también
 C* cuando se produce un cambio de código de proyecto (la lógica RPG
 C* activa todos los indicadores de nivel de control de nivel inferior
 C* cuando se produce una interrupción de control, es decir, cuando
 C* L2 se activa, y L1 también lo hace). La subrutina SUBRL2 se
 C* procesa cuando se produce un cambio de código de proyecto.

CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*

C L2 EXSR L2CLR
 C EXSR DTLSR
 CL1 EXSR SUBRL1
 CL2 EXSR SUBRL2

C*****
 C* SUBROUTINA DTLSR: Esta rutina realiza operaciones en tiempo de detalle.
 C* Las horas trabajadas EHWRK del registro de transacción se añaden al
 C* elemento actual de la matriz de horas de proyecto. Al archivo maestro
 C* de empleados se accede utilizando la operación CHAIN y al número de
 C* empleado se accede desde el registro de transacción. El registro de
 C* detalle se graba a continuación al archivo de impresora PRG07RP
 C* utilizando el formato de registro DETAIL. El recuento de líneas se
 C* incrementa en uno, y la rutina de desbordamiento LICHK se procesa
 C* para determinar si es necesario un salto a una nueva página y
 C* salida de línea de cabecera.

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*

C DTLSR BEGSR
 C EHWRK ADD ARRP,P ARRP,P
 C EMPNO CHAINEMPMST 61
 C WRITEDETAIL
 C ADD 1 LICNT
 C EXSR LICHK
 C ENDSR

C*****
 C* SUBROUTINA SUBRL1: Esta rutina realiza operaciones de tiempos totales.
 C* El total de horas de la semana actual de la matriz de horas de proyecto
 C* se mueve al campo de impresión WKTOT utilizando la operación Z-ADD y
 C* la aparición actual de la matriz de proyecto (valor de índice de matriz
 C* en P). La línea de totales semanal se graba en el archivo de impresora
 C* PRG07RP utilizando el formato de impresión TOTL1. El índice P de la
 C* matriz de horas de proyecto se incrementa en uno para acumular las
 C* horas de la semana siguiente, y el recuento de líneas se incrementa en
 C* dos. La rutina de desbordamiento LICHK se procesa para determinar si es
 C* necesario un salto a una nueva página y salida de línea de cabecera.

C SUBRL1 BEGSR
 C Z-ADDARRP,P WKTOT
 C WRITETOTL1
 C ADD 1 P
 C ADD 2 LICNT
 C EXSR LICHK
 C ENDSR

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..

C*****
 C* SUBROUTINA SUBRL2: Esta rutina realiza operaciones de tiempos totales.
 C* El valor de recuento de líneas se comprueba. Si el valor es mayor o
 C* igual a 55, se establece en 60 y la rutina de desbordamiento LICHK se
 C* procesa. El programa realiza estas operaciones para asegurar que
 C* existen suficientes línea de impresión en la página para imprimir
 C* las líneas de totales del proyecto. Se suma la matriz del proyecto
 C* utilizando la operación XFOOT. Esta operación une todos los elementos
 C* de la matriz y sitúa la suma en el campo de resultado. Las siguientes
 C* series de operaciones aritméticas preparan las líneas de resumen del
 C* proyecto para impresión. Las horas totales del proyecto del mes se
 C* añaden a las horas totales del proyecto del año a la fecha desde el
 C* archivo de proyectos para determinar las horas actuales del año a la
 C* fecha. Las horas actuales del año a la fecha se añaden a total del
 C* año anterior para determinar las horas totales del proyecto. Las horas
 C* totales del proyecto se toman de las horas totales estimadas, y se
 C* cambia el signo del resultado utilizando la operación Z-SUB para obtener
 C* las horas de variación. Las horas de variación se dividen por las horas
 C* totales estimadas y luego se multiplican por 100 para obtener el
 C* porcentaje de variación. Este porcentaje indica si el porcentaje de
 C* las horas reales es superior (% positivo) o inferior (%
 C* negativo) respecto a las horas estimadas.

CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*

```

C          SUBRL2  BEGSR
C          LICNT  IFGE 55
C                    Z-ADD60          LICNT
C                    EXSR LICHK
C                    END
C                    XFOOTARRP        PRMTH
C          PRMTH  ADD  EPHRY          PRYER
C          PRYER  ADD  PRHRP          PRTOT   91
C          PREST  SUB  PRTOT          PRDIF   91
C                    Z-SUBPRDIF        PRDIF
C          PRDIF  DIV  PRTOT          WRK1   53H
C          WRK1  MULT 100          PRVAR   51

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* La línea de totales del empleado está ahora preparada para la impresión.
C* Se graba el formato TOTL2. Dado que el programa realizará operaciones
C* L2 de tiempo de detalle en el siguiente ciclo para prepararse para el
C* código de proyecto siguiente, no se incrementa el campo de recuento de
C* líneas y no se comprueba el desbordamiento después de grabarse el
C* programa.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          WRITETOTL2
C          ENDSR
C*
C*****
C* SUBROUTINA LICHK: Esta rutina controla el desbordamiento de página.
C* Si el recuento de líneas es mayor o igual a 60, se graba el formato
C* de cabecera TITL1 y el recuento de líneas se establece en 9.
C*
C          LICHK  BEGSR
C          LICNT  IFGE 60
C                    WRITETITL1
C                    Z-ADD9          LICNT
C                    END
C          ENDSR

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C* SUBROUTINA L2CLR: Esta subrutina prepara los campos de trabajo e
C* imprime las cabeceras de línea antes de procesar el primer registro
C* de detalle del proyecto. La matriz de horas de proyecto ARRP se
C* inicializa a 0 y el índice de matriz se establece en 1. Al archivo
C* maestro de proyectos se accede utilizando el código de proyecto del
C* registro de transacciones. Si el registro de proyecto no se encuentra,
C* se activa el indicador 60. Las cabeceras del informe se imprimen
C* grabando el formato de impresión TITL1, y el recuento de líneas
C* se establece en 9.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          L2CLR  BEGSR
C                    Z-ADD0          ARRP
C                    Z-ADD1          P          10
C          PRCDEX  CHAINPRJMST          60
C          *IN60  IFEQ '1'
C                    MOVE *BLANKS  PRDSC
C                    MOVE 'NO VALIDO' PRDSC
C                    END
C                    WRITETITL1
C                    Z-ADD9          LICNT   30
C          ENDSR

```

Figura 197. Programa RPG/400 de ejemplo - PRG07

13.8.9 Descripciones de datos del informe de resumen de códigos de razón - PRG08RP

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
A*****
A* Este archivo de impresión describe el formato del informe de resumen
A* mensual de información de tiempos de códigos de razón. En este
A* archivo de impresora hay cuatro formatos de registro, que se identifican
A* por una R en la posición 17 seguida del nombre del formato en las
A* posiciones 10 a 20. Se utilizan las palabras clave siguientes:
A* DATE - Especifica la fecha del sistema
A* EDTCDE(a) - Edita los campos numéricos con posibilidad de salida
A* PAGNBR - Especifica un campo decimal con zona, de cuatro dígitos
A* para contener el número de página.
A* REF(REFMST) - Las líneas que contienen una R en la posición 29
A* utilizan los atributos de un campo definido
A* anteriormente en este archivo de referencia.
A* REFFLD - Para un campo que hace referencia a un campo definido
A* anteriormente.
A* SKIPB(n) - Especifica que el dispositivo de impresora va a
A* saltar a una línea específica antes de imprimir
A* la línea siguiente.
A* SPACEA(n) - Especifica que el dispositivo de impresora va a dejar
A* espacios de n líneas tras imprimir una o más líneas.
A* SPACEB(n) - Especifica que el dispositivo de impresora va a dejar
A* espacios de n líneas antes de imprimir la línea o
A* líneas siguientes.
A*****
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++*****
A REF(REFMST)
A*
A* El primer formato, TITLE1, contiene la definición para las líneas
A* líneas de cabecera del informe. El formato se graba en el
A* primer ciclo, cuando se produce un cambio de código de razón o cuando
A* se produce un desbordamiento durante la impresión de los
A* detalles de un código de razón.
A*

```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++*****
A R TITLE1 SKIPB(3)
A 2'PRG08RP'
A 29'RESUMEN DE LA INFORMACION DE
A CODIGOS DE RAZON'
A 83'PAGINA'
A 88PAGNBR
A SPACEA(1)
A 31'PARA EL PERIODO FINALIZADO'
A CMTDT R 52EDTCDE(Y)
A 84DATE EDTCDE(Y)
A SPACEA(2)
A 2'CODIGO DE RAZON'
A RSCDEX R 15REFFLD(RSCDE)
A 29'DESCRIPCION'
A RSDSC R 42
A SPACEA(2)
A 2'NUMERO DE NOMBRE DE EMPLEADO'
A 52'FECHA DE FINAL HORAS'
A SPACEA(1)
A 3'EMPLEADO'
A 55'DE LA SEMANA TRABAJADAS'
A SPACEA(1)
A*
A* El segundo formato, DETAIL, contiene la definición de las líneas de
A* impresión de detalle. El formato se graba para cada registro de
A* detalle en el archivo de transacciones mensual.
A*

```

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
AAN01N02N03T.Nombre++++RLon++TDpBLínPosFunciones+++++*****
A R DETAIL
A EMPNO R 3EDTCDE(Z)
A N61 EMPNAM 30 16
A 61 16'NUMERO DE EMPLEADO NO VALIDO'
A CWKDTX R 53REFFLD(CWKDT)
A EDTCDE(Y)
A EHWRK R 69EDTCDE(L)
A SPACEA(1)
A*
A* El tercer formato, TOTL1, contiene la definición de la interrupción
A* de nivel L1 de los tiempos totales. El formato se graba cuando se
A* produce un cambio de fecha de final de semana o un cambio de
A* código de razón.
A*

```

Descripciones de datos del informe de resumen de códigos de razón - PRG08RP

```

A          R TOTL1                SPACEB (1)
A          52 'TOTAL SEMANAL'
A          WKTOT                7S 1    67EDTCDE (L)
A          SPACEA (2)
A*
A* El cuarto formato, TOTL2, contiene la definición de la interrupción
A* de nivel L2 de tiempos totales. El formato se graba cuando se produce
A un cambio de código de razón.
A*
A          R TOTL2                SPACEB (1)
A          2 'RESUMEN DEL CODIGO DE RAZON'
A          29 'MES                AÑO ACTUAL'
A          60 'TOTAL DEL'
A          SPACEA (1)
A          30 'ACTUAL            A LA FECHA'
A          63 'AÑO ANTERIOR'
A          SPACEA (1)
A          RSMTH                7S 1    28EDTCDE (L)
A          RSYER                9S 1    41EDTCDE (L)
A          RSHRP                R      60EDTCDE (L)

```

Figura 199. Descripciones de datos del informe de resumen de códigos de razón - PRG08RP

13.8.10 Programa RPG/400 de informe de resumen de códigos de razón - PRG08

La Figura 200 muestra el programa PRG08 de RPG/400, incluyendo comentarios para explicar el flujo lógico y la utilización de las diversas funciones y códigos de operación RPG/400.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*****
F* PRG08 - Informe de resumen de información de tiempos de códigos
F* de razón
F* DESCRIPCION - Este programa produce el informe de resumen mensual de
F* información de tiempos de códigos de razón. Todas las
F* entradas de tiempos del mes se imprimen por número de
F* empleado, dentro de la fecha de final de semana, dentro
F* del código de razón. Los subtotales se imprimen por semana
F* y se imprime un resumen de códigos de razón cuando
F* se produce un cambio de código de razón.
F*****
F* Este programa utiliza archivo descritos externamente. Los archivos
F* utilizados son:
F* TRMNTHN - visión lógica de TRMNTH, archivo de transacciones
F* mensual por código de razón, número de empleado,
F* y fecha de final de semana.
F* EMPMST - archivo maestro de empleados
F* RSNMST - archivo maestro de códigos de razón
F* PRG08RP - archivo del informe de resumen de códigos de razón
F*****
F* INDICADORES UTILIZADOS:
F* 60 - No encontrado el registro maestro de códigos de razón
F* 61 - No encontrado el registro maestro de empleados
F* 90 - Se ha encontrado una serie de caracteres en la operación SCAN
F* L1 - Nivel de control en fecha de final de semana
F* L2 - Nivel de control en código de razón
F*****
F* SUBROUTINAS UTILIZADAS:
F* DTLSR - Rutina de cálculos de detalle
F* *INZSR - Subrutina de inicialización
F* L1CHK - Rutina de comprobación de recuento de líneas
F* L2CLR - Borrar campos de trabajo en tiempo de detalle L2
F* SUBRL1 - Cálculos de tiempos totales - cambio de fecha de final
F* de semana
F* SUBRL2 - Cálculos de tiempos totales - cambio de código de razón
F*****

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
FNombarchIPEAF...LregLCIAIDePcLaEDispos+.....KSalidaEntradA...U1.*
FTRMNTHR IP E K DISK
FEMPMST IF E K DISK
FRSNMST IF E K DISK
FPRG08RP O E PRINTER

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
E*
E* La matriz siguiente se utiliza para almacenar las horas semanales
E* del código de razón. La matriz contiene hasta cinco totales
E* semanales.
E*
E*...Arch(de)Arch(a)+NombreN/rN/tbLonPDSNommatLonPDSComentarios++++*
E ARR 5 5 1

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I*
I* El código siguiente vuelve a asignar un nombre a los nombres de
I* campo de entrada del archivo de transacciones mensual. Estos campos
I* aparecen en otras definiciones de datos, y se recubren cuando se leen
I* dichos archivos, estos nuevos nombres de códigos evitan el recubrimiento.
INomreg++...In.....*
IRCMNTH
I.....Campoexter.....Campo+L1C1..PoNeCe...*
I RSCDE RSCDEXL2
I CWKDT CWKDTXL1
I CMTDT CMTDTX
I* Area de datos del archivo de control descrita externamente
I*
IDsnombre..NODsSal-arch+...NvecLon+.....*
ICTLFIL EUDS
I*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C* RUTINA PRINCIPAL: La rutina principal consta de cuatro subrutinas
C* EXSR. La subrutina de inicialización *INZSR se procesa en primer lugar,
C* durante el paso para la inicialización del ciclo del programa. La

```

C* subrutina *INZSR inicializa los campos que se utilizan para los cálculos.
 C* La subrutina L2CLR se procesa en tiempo de detalle cuando el indicador
 C* de nivel de control L2 está activado. Esto se produce en el primer ciclo
 C* del programa RPG/400 y en el ciclo del programa RPG/400 que sigue a los
 C* cálculos de tiempos totales. La subrutina L2CLR borra los campos de
 C* trabajo y graba las cabeceras del informe. La segunda subrutina, DTLR,
 C* se procesa en cada ciclo en tiempo de detalle RPG/400. Esta rutina graba
 C* las líneas del informe de detalle y acumula datos para la impresión de
 C* tiempos totales. Las subrutinas tercera y cuarta se procesan en tiempo
 C* de totales. La subrutina SUBRL1 se procesa cuando se produce un cambio
 C* de fecha de final de semana y un cambio de código de razón (la lógica
 C* RPG/400 activa todos los indicadores de control de nivel inferior
 C* cuando se produce una interrupción de control, es decir, cuando L2 se
 C* activa, también lo hace L1). La subrutina SUBRL2 se procesa
 C* cuando se produce un cambio de código de razón.

C*
 CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*

C L2 EXSR L2CLR
 C EXSR DTLR
 CL1 EXSR SUBRL1
 CL2 EXSR SUBRL2

C*
 C*****
 C* SUBROUTINA DTLR: Esta rutina realiza operaciones en tiempo de detalle.
 C* Las horas trabajadas EHWRK del registro de transacciones se añaden al
 C* elemento actual de la matriz de horas del código de razón. Al archivo
 C* maestro de empleados se accede utilizando la operación CHAIN y el
 C* número de empleado desde el registro de transacciones. Si se encuentra
 C* un registro EMPST, el indicador 61 se desactiva y las operaciones
 C* SCAN, SUBST, y CAT dan formato al nombre de empleado para el informe.
 C* El nombre de empleado del registro de entrada está en formato de
 C* primer nombre, seguido de un espacio en blanco, seguido del primer
 C* apellido. El nombre de empleado se imprime en el informe en formato
 C* inverso: primer apellido, seguido de un blanco, seguido del nombre.
 C* La operación SCAN determina la posición del blanco en ENAME. Si SCAN
 C* se procesa satisfactoriamente, se realiza un cálculo de las longitudes
 C* de nombre y apellido; las operaciones SUBST extraen el nombre y apellido
 C* del campo ENAME; la operación CAT concatena los nombres con un espacio
 C* entre ellos. Si SCAN no se procesa satisfactoriamente, el nombre de
 C* empleado se imprime en el informe tal como aparece en el registro de
 C* entrada. El registro de detalle se graba entonces al archivo de
 C* impresora PRG08PR utilizando el formato de registro DETAIL. El
 C* recuento de líneas se incrementa en uno y la rutina de desbordamiento
 C* LICHK se procesa para determinar si se requiere un salto a una
 C* nueva página y salida de línea de cabecera.
 C*

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..
 CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*

C DTLR BEGSR
 C EHWRK ADD ARR,N ARR,N
 C EMPNO CHAINEMPST 61
 C *IN61 IFEQ '0'
 C BLK1 SCAN ENAME BLKPOS 20 90
 C *IN90 IFEQ '1'
 C 1 ADD BLKPOS SPOS 20
 C LENENM SUB BLKPOS LENLNM 20
 C LENLNM SUBSTENAME:SPOSLNAME 30
 C BLKPOS SUB 1 EPOS 20
 C EPOS SUBSTENAME FNAME 30
 C LNAME CAT FNAME:1 EMPNAM
 C CLEARFNAME
 C CLEARLNAME
 C ELSE
 C MOVE ENAME EMPNAM
 C END
 C END
 C WRITEDetail
 C ADD 1 LICNT
 C EXSR LICHK
 C ENDSR

C*****

.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..
 C*****

C* SUBROUTINA *INZSR: Esta rutina inicializa los campos utilizando las
 C* operaciones MOVE y Z-ADD. Los campos BLK1 y LENENM se utilizan para
 C* dar formato al nombre de empleado en la subrutina DTLR. El campo N
 C* es el índice para la matriz ARR.N.
 C*

CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*

C *INZSR BEGSR
 C MOVE *BLANK BLK1 1
 C Z-ADD30 LENENM 20

```

C          Z-ADD1          N          10
C          ENDSR
C*
C*****
C* SUBROUTINA SUBRL1: Esta rutina realiza operaciones de tiempos totales.
C* Las horas totales de la semana actual de la matriz de horas del código
C* de razón se mueve al campo de impresión WKTOT utilizando la operación
C* Z-ADD y la aparición actual de la matriz del código de razón (valor de
C* índice de matriz en N). La línea del total semanal se graba al archivo
C* de impresora PRG08RP utilizando el formato de impresión TOTL1. El
C* índice N de la matriz de horas del código de razón se incrementa en uno
C* para acumular las horas de la semana siguiente, y el recuento de líneas
C* se incrementa en dos. La rutina de desbordamiento LICHK se procesa para
C* determinar si se requieren un salto a una nueva página y salida
C* de línea de cabecera.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          SUBRL1          BEGSR
C          Z-ADDARRN,N      WKTOT
C          WRITETOTL1
C          ADD 1            N
C          ADD 2            LICNT
C          EXSR LICHK
C          ENDSR

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C* SUBROUTINA SUBRL2: Esta subrutina realiza operaciones de tiempos
C* totales. Se comprueba el valor del recuento de líneas. Si es mayor o
C* igual a 55, se establece en 60 y se procesa la subrutina de
C* desbordamiento LICHK. El programa realiza estas operaciones para
C* asegurar que existen suficientes líneas de impresión en la página para
C* imprimir las líneas de totales del código de razón. La matriz del
C* código de razón se suma utilizando la operación XFOOT. Esta operación
C* une todos los elementos de la matriz y sitúa la suma en el campo de
C* resultado. Las horas actuales totales del mes se añaden a las horas del
C* año a la fecha desde el archivo de códigos de razón para determinar
C* las horas actuales del año a la fecha.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          SUBRL2          BEGSR
C          LICNT           IFGE 55
C          Z-ADD60          LICNT
C          EXSR LICHK
C          END
C          XFOOTARRN        RSMTH
C          RSMTH           ADD  RSHRY  RSYER
C*
C* La línea de totales del código de razón está ahora preparada para la
C* impresión. Se graba el formato TOTL2. Dado que el programa efectuará
C* operaciones L2 en tiempo de detalle en el siguiente ciclo para preparar
C* el código de razón siguiente, el campo de recuento de líneas no se
C* incrementa, y no se realizan comprobaciones del desbordamiento tras
C* grabar el formato.
C*
C          WRITETOTL2
C          ENDSR

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C* SUBROUTINA LICHK: Esta rutina controla el desbordamiento de página.
C* Si el recuento de líneas es mayor o igual a 60, se graba el formato
C* de cabecera TITLE1 y el recuento de líneas se establece en 9.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++
C          LICHK           BEGSR
C          LICNT           IFGE 60
C          WRITETITLE1
C          Z-ADD9          LICNT
C          END
C          ENDSR
C*
C*****
C* SUBROUTINA L2CLR: Esta subrutina prepara los campos de trabajo e
C* imprime las cabeceras de línea antes de procesar el primer registro
C* de detalle del código de razón. La matriz de horas del código de
C* razón ARRP se establece en 0 utilizando la operación CLEAR y el índice
C* de la matriz se establece en 1 utilizando la operación RESET. El índice
C* de la matriz, N, se inicializa a 1 en la subrutina de inicialización
C* y se restablece a dicho valor. Al archivo maestro de códigos de razón
C* se accede utilizando el código de razón del registro de transacciones.
C* Si el registro de códigos de razón no se encuentra, el indicador 60
C* se activa. Las cabeceras del informe se imprimen grabando el
C* formato de impresión TITLE1, y el recuento de líneas se establece

```

C* en 9.

C*

C	L2CLR	BEGSR		
C		CLEARARRN		
C		RESETN		
C	RSCDEX	CHAINRSNMST		60
C		WRITETITLE1		
C		Z-ADD9	LICNT	30
C		ENDSR		

Figura 200. Programa RPG/400 de ejemplo - PRG08

13.8.11 Programa RPG/400 de actualización y borrado mensual del archivo maestro - PRG04

La Figura 201 muestra el programa PRG04 de RPG/400, incluyendo comentarios para explicar el flujo lógico y la utilización de las diversas funciones y códigos de operación RPG/400.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
F*****
F* PRG04 - Actualización del archivo maestro de información tiempos
F* DESCRIPCION - Este programa realiza la totalización mensual y de
F* las horas de información de tiempos. Los archivos y
F* el tipo de actualización se controlan mediante
F* indicadores externos.
F*****
F* Este programa utiliza archivo descritos externamente. Los archivos
F* utilizados son: EMPMST - archivo maestro de empleados
F* PRJMST - Archivo maestro de proyectos
F* RSNMST - Archivo maestro de códigos de razón
F*****
F* INDICADORES UTILIZADOS:
F* 50 - Final del archivo
F* U1 - Actualización del maestro de empleados
F* U2 - Actualización del maestro de proyectos
F* U3 - Actualización del maestro de códigos de proyecto
F* U4 - Proceso de final de año
F*****
F* SUBRUTINAS UTILIZADAS:
F* EMPSR - Actualizar maestro de empleados
F* PRJSR - Actualizar maestro de proyectos
F* RSNSR - Actualizar maestro de códigos de razón
F*****
FNombarchIPEAF...LregLC1AIDePcLaEDispos+.....KSalidaEntradA...U1.*
FEMPMST UF E K DISK U1
FPRJMST UF E K DISK U2
FRSNMST UF E K DISK U3

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C* RUTINA PRINCIPAL: La rutina principal determina el archivo a procesar
C* y ejecuta la subrutina adecuada. Todos los archivos maestros se
C* describen como de proceso completo (F en la posición 16 de la
C* especificación de descripción del archivo), lo que permite al
C* programa efectuar operaciones de lectura/grabación en los archivos.
C* Todos los archivos se controlan mediante un indicador de condición de
C* archivo (posición 71 a 71 de la especificación de descripción de
C* archivo) que controla el archivo que se procesa.
C*****
C* El código siguiente determina qué subrutina se evoca.
C* SI *INU1 está activado (igual a 1), se selecciona EMPSR; si *INU2 está
C* activado, se selecciona PRJSR; si *INU3 está activado, se selecciona
C* RSNSR.
C*
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C *INU1 CASEQ'1' EMPSR
C *INU2 CASEQ'1' PRJSR
C *INU3 CASEQ'1' RSNSR
C END
C*****
C* SUBRUTINA EMPSR: La línea de código siguiente realiza una condición
C* Hacer Hasta (Do Until). El programa repite en bucle entre la
C* sentencia DOUEQ y la sentencia END hasta que se haya alcanzado el
C* final del archivo (*IN50 igual a 1).
C*
C EMPSR BEGSR
C*
C *IN50 DOUEQ'1'
C READ RCEMP 50

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C* Las siguientes líneas de código añaden las horas del mes actual a las
C* horas del año a la fecha para el archivo maestro de empleados. Dado
C* que en las mismas sentencias no se especifica el factor 1, el factor 2
C* se añade a los campos de resultado y el resultado se sitúa en el campo
C* de resultado. Si *INU4 está activado, esta se sesión se ejecuta para
C* el final de año, y las horas del año actual se mueven a las horas
C* del año anterior.
C*
CLON01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C ADD EPHRC EPHRY
C ADD EPNRC EPNRY
C U4 MOVE EPHRY EPHRP
C U4 MOVE EPNRY EPNRP
C* El código siguiente borra los campos de las horas del mes actual

```

```

C* poniéndolas a cero y añadiéndoles 0. Si *INU4 está activado, esta
C* sesión se ejecuta para el final del año, y las horas del año actual
C* también deben ponerse a cero.
C          Z-ADD0          EPHRC
C          Z-ADD0          EPNRC
C  U4          Z-ADD0          EPHRY
C  U4          Z-ADD0          EPNRY
C* El código siguiente actualiza el archivo maestro de empleados
C* utilizando el formato RCEMP.
C          UPDATRCEMP
C          END
C* La sentencia END anterior se asocia a la sentencia
C* DOUEQ.
C*
C* EL indicador del último registro *INLR se activa (igual a 1) y
C* el programa finaliza.
C          MOVE '1'          *INLR
C          ENDSR

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*****
C* SUBROUTINA PRJSR: Esta subrutina realiza las mismas funciones que la
C* subrutina EMPSR; sólo se actualiza el maestro de proyectos.
C* Consulte EMPSR para obtener información específica.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C          PRJSR          BEGSR
C          *IN50          DOUEQ'1'
C          READ RCPRJ          50
C*
C* Añada el mes actual al año a la fecha, y mueva el año actual al
C* año anterior si U4 está activado.
C          ADD PRHRC          PRHRY
C  U4          MOVE PRHRY          PRHRP
C*
C* Ponga a cero el mes actual, y el año a la fecha si U4 está activado.
C          Z-ADD0          PRHRC
C  U4          Z-ADD0          PRHRY
C*
C* Actualice el archivo maestro de proyectos.
C          UPDATRCPRJ
C          END
C*
C* Active el último indicador de registro.
C          MOVE '1'          *INLR
C          ENDSR
C*****
C* SUBROUTINA RSNSR: Esta subrutina realiza las mismas funciones que la
C* subrutina EMPSR; sólo se actualiza el maestro de códigos de razón.
C* Consulte EMPSR para obtener información específica.
C*
C          RSNSR          BEGSR
C          *IN50          DOUEQ'1'
C          READ RCRSN          50

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* Añada el mes actual al año a la fecha, y mueva el año actual al
C* año anterior si U4 está activado.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C          ADD RSHRC          RSHRY
C  U4          MOVE RSHRY          RSHRP
C*
C* Ponga a cero el mes actual, y el año a la fecha si U4 está activado.
C          Z-ADD0          RSHRC
C  U4          Z-ADD0          RSHRY
C*
C* Actualice el archivo maestro de códigos de razón
C          UPDATRCRSN
C          END
C*
C* Active el último indicador de registro.
C          MOVE '1'          *INLR
C          ENDSR
C*

```

Figura 201. Programa RPG/400 de ejemplo - PRG04

13.9 Proceso de final de año

Todos los archivos maestros se procesan como parte de la última actualización mensual para preparar los archivos para el nuevo año. Cada archivo maestro contiene las horas totales del año a la fecha actual y del año anterior. El programa PRG04 realiza la revisión de final de año y mensual de las horas de información de tiempos. Se utilizan conmutadores externos para controlar qué archivo se procesa, y si la sesión es para un final de mes normal o para un final de mes y final de año combinados. Consulte la información detallada sobre el programa PRG04 que se facilita para obtener más detalles.

Apéndice A. Información de servicio del programa generador automático de informes y del compilador RPG

A.0 Apéndice A. Información de servicio del programa generador automático de informes y del compilador RPG

Este apéndice va dirigido al personal de servicio del compilador RPG/400 para su utilización durante la investigación de problemas con el compilador RPG/400, y facilita la información siguiente:

- Visión general del compilador
- Opciones de depuración del compilador
- Diseño de la representación intermedia del programa (IRP)
- Visión general del generador automático de informes.

Los programadores del compilador RPG/400 también puede utilizar esta información para investigar los problemas del compilador RPG/400 por su cuenta antes o en lugar de recurrir al servicio.

Subtemas

- A.1 Visión general del compilador
- A.2 Opciones de depuración del compilador
- A.3 Ejemplos de utilización de opciones de depuración del compilador
- A.4 Diseño de la IRP
- A.5 Programa generador automático de informes

A.1 Visión general del compilador

Esta sección proporciona la siguiente información sobre el compilador:

- Cómo funciona el compilador
- Descripciones de las fases del compilador
- Descripciones de las principales áreas de datos del compilador
- Organización de los mensajes de error del compilador.

La Figura 202 en el tema A.1.1 resumen cómo se compila un programa fuente RPG/400 a un objeto del programa (encapsulado).

El texto intermedio, que es la salida del paso 1 en la Figura 202 en el tema A.1.1, es una representación de las sentencias fuente RPG/400 que se crea mediante fases de compilador y sólo existe durante la ejecución de éstas. Este texto puede listarse dinámicamente mediante el parámetro ITDUMP del mandato CL, CRTRPGPGM (Crear programa RPG/400), o bien puede listarse al terminar cualquier fase del compilador mediante el parámetro SNPDUMP del mandatos CRTRPGPGM CL. Consulte "Opciones de depuración del compilador" en el tema A.2 para obtener las explicaciones sobre estos parámetros y ejemplos de texto intermedio.

Cuando la compilación finaliza, el texto intermedio se ha procesado y convertido a la IRP adecuada (representación intermedia de un programa). La IRP, que es la salida del paso 2 Figura 202 en el tema A.1.1, puede listarse dinámicamente mediante el parámetro CODELIST del mandato CRTRPGPGM CL, o puede listarse al finalizar la compilación con un valor ***LIST** para el parámetro GENOPT del mandato CRTRPGPGM CL. Consulte "Opciones de depuración del compilador" en el tema A.2 para obtener explicaciones de estos parámetros y ejemplos de las sentencias IRP.

Una plantilla de programa es la salida del paso 3 de la Figura 202 en el tema A.1.1. Una plantilla es la forma final de un programa antes de convertirse en un programa ejecutable, que se denomina entonces programa encapsulado. Una plantilla puede listarse al final de una compilación mediante el valor ***DUMP** en el parámetro GENOPT del mandato CRTRPGPGM CL. Consulte "Opciones de depuración del compilador" en el tema A.2 para obtener las explicaciones sobre estos parámetros y un ejemplo de listado de plantilla de programa.

Subtemas

- A.1.1 Fases del compilador
- A.1.2 Principales áreas de datos del compilador
- A.1.3 Organización de los mensajes de error del compilador
- A.1.4 Subrutinas en tiempo de ejecución

A.1.1 Fases del compilador

El compilador se compone de las fases que se listan en la Tabla 20. Estas fases se muestran en el orden en que se ejecutan.

Si se ha especificado ***NOGEN** en el parámetro **OPTION** del mandato **CRTRPGPGM CL**, la compilación finaliza a continuación de la fase **QRGCR**.

Durante la compilación, estas fases, que tienen una U en la tercera columna de la Tabla 20 se ejecutan incondicionalmente, y las fases que tienen una C en la tercera columna se ejecutan sólo si el programa que está compilándose las necesita.

La primera fase del compilador se denomina **QRG1**. Todas las fases siguen a **QRG1** tienen nombres que empiezan por **QRG** y terminan con dos caracteres de identificación. A estas fases puede hacerse referencia mediante sus caracteres de identificación. Por ejemplo, esos caracteres pueden utilizarse como valores para los parámetros de depuración en el mandato **CRTRPGPGM CL**. Consulte "Opciones de depuración del compilador" en el tema A.2 para obtener más información.

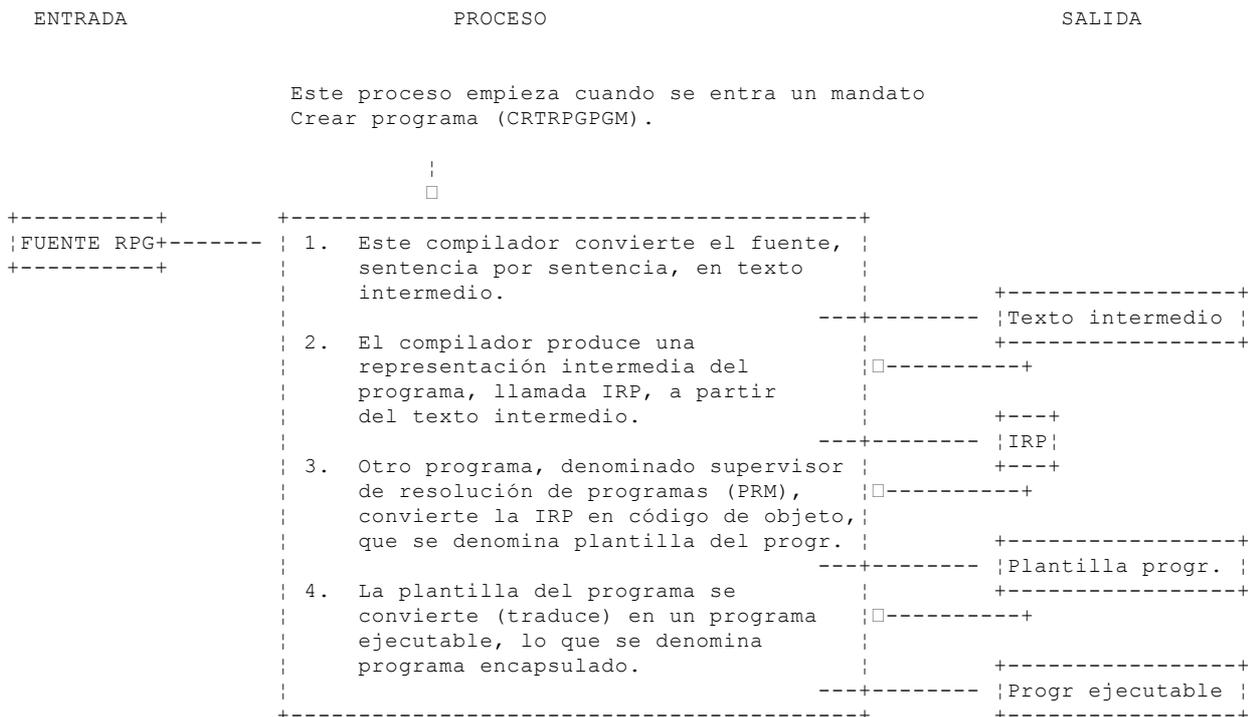


Figura 202. Visión general del compilador RPG/400

Tabla 20. Fases del compilador		
Nombre fase	Descripción de la fase	Llamada: Incondicionalmente (U) Condicionamente (C)
QRG1	Interfaz de mandato que recibe el control cuando se entra el mandato CRTRPGPGM , asigna valores por omisión a la lista de parámetros del mandato y transfiere esta lista de parámetros del mandato a QRGRT .	U
QRGRT	Fase raíz que controla la llamada de todas las demás fases del compilador y contiene todas las interfaces del sistema, como de lectura e impresión de registros.	U
QRGSF	Fase que diagnostica las especificaciones de descripción del archivo y crea una tabla de archivos.	C
QRGSE	Fase que diagnostica las especificaciones de ampliación y crea	C

	texto intermedio.	
QRGSI	Fase que diagnostica las especificaciones de entrada y crea texto intermedio.	C
QRGSC	Fase que diagnostica las especificaciones de cálculo y crea texto intermedio.	C
QRGSO	Fase que diagnostica las especificaciones de salida y crea texto intermedio.	C
QRGAE	Fase que genera sentencias declarativas para campos y crea máscaras de edición.	U
QRGD1	Fase que diagnostica errores de relación entre especificaciones fuente. Estos errores son combinaciones de entradas ilógicas o incorrectas.	U
QRGCR	Fase que produce un listado de referencias cruzadas, genera el código para procesar las tablas en tiempo de ejecución, y produce una lista de mensajes en tiempo de compilación sobre la terminación de una compilación.	U
QRGGB	Fase que genera bloques de control de archivos de usuario (los UFCB).	U
QRGFB	Fase llamada por QRGGB que sirve para generar bloques de información de archivos (FIB).	U
QRGPL	Fase llamada por QRGGB que sirve para generar las PLIST.	U
QRGGV	Fase llamada por QRGGB que sirve para generar controladores de E/S de archivos.	C
QRGGC	Fase principal que controla el proceso de las operaciones de cálculo.	C
QRGAC	Fase llamada por QRGGC para procesar las operaciones de cálculos aritméticos.	C
QRGBC	Fase llamada por QRGGC para procesar las operaciones de cálculos por lotes.	C
QRGCC	Fase llamada por QRGGC para procesar las operaciones de cálculos de comparación.	C
QRGIC	Fase llamada por QRGGC que sirve para generar los enlaces de entrada/salida de las operaciones de cálculo.	C
QRGMC	Fase llamada por QRGGC que sirve para generar códigos para operaciones de cálculo MOVE .	C
QRGRC	Fase llamada por QRGGC para procesar operaciones de cálculo CLEAR y RESET .	C
QRGTC	Fase llamada por QRGGC para procesar operaciones de cálculo en serie.	C
QRGYC	Fase llamada por QRGGC para procesar operaciones de cálculo de varios.	C
QRGGI	Fase que genera códigos que extraen datos de registros y rellenan campos de entrada.	C
QRGGO	Fase que genera códigos que crean registros de salida.	C
QRGGS	Fase que genera la gestión de datos para DISK y para archivos SEQ .	C

Guía del usuario RPG/400
Fases del compilador

QRGGW	Fase llamada por QRGGS para generar la gestión de datos para archivos WORKSTN .	C
QRGGR	Fase llamada por QRGGS que sirve para generar la gestión de datos para archivos RAF .	C
QRGC1	Fase que genera códigos para obtener la salida y procesar múltiples archivos.	U
QRGEC	Fase que genera las subrutinas necesarias para el programa, que genera el código para procesar las tablas en tiempo de ejecución, y que genera el código inicial y final del programa.	U

A.1.2 Principales áreas de datos del compilador

Las principales áreas de datos del compilador son: el área común (**VCOMMON**), la tabla de nombres de campo (**XFDTAB**), la tabla de nombres archivo (**XFLTAB**), la tabla de nombres de registro (**XRCTAB**), y la tabla de indicadores (**XINTAB**).

A.1.3 Organización de los mensajes de error del compilador

Los mensajes de error del compilador están organizados de acuerdo con las fases que los emiten. Por ejemplo, cualquier mensaje del compilador que empiece por 2 lo emite la fase **QRGSF**. La tabla siguiente lista las fases de compilador y los mensajes que estas emiten:

Tabla 21. Fases del programa generador automático de informes	
Mensajes de error	Fase
0000 a 1999	QRGRT
2000 a 2999	QRGSF
3000 a 3999	QRGSE
4000 a 4999	QRGSI
5000 a 5999	QRGSC
6000 a 6999	QRGSO
7000 a 7999	QRGDI y QRGCR
8000 a 8999	QRGAE y QRGCR

A.1.4 Subrutinas en tiempo de ejecución

La Tabla 22 lista las subrutinas en tiempo de ejecución que utiliza el compilador.

Tabla 22. Subrutinas en tiempo de ejecución		
Nombre de la subrutina	Descripción de la subrutina	Llamada: Incondicionalmente (U) Condicionalemente (C)
QRGXINIT	Inicializa el programa RPG/400.	U
QRGXDUMP	Proporciona un vuelco con formato del programa RPG/400.	C
QRGXERR	Se llama cuando se recibe un mensaje de error.	C
QRGXPRF	Llamada por QRGXDUMP para imprimir el vuelco.	C
QRGXSTAT	Llamada para la operación POST con un dispositivo especificado en el factor 1.	C
QRGXTIME	Llamada para el código de operación TIME .	C
QRGXIOU	Llamada para los códigos de operación IN/OUT/UNLCK . El compilador también la utiliza para establecer el área de datos RETURNCODE y para recuperar el área de datos de la especificación de control RPG/400 durante la compilación.	U
QRGXMSG	Envía mensajes RPG/400 en tiempo de ejecución al solicitante y proporciona el vuelco del sistema cuando se solicita.	C
QRGXCLRF	Borra el archivo antes de que se produzca un vuelco de tabla al finalizar el programa.	C
QRGXDSP	Llamada para el código de operación DSPLY .	C
QRGXSIGE	Indica una condición de excepción para un error de terminal en tiempo de ejecución.	C
QRGXGDDM	Llamada para la operación GDDM CALL .	C
QRGINVX	Desbloquea las áreas de datos cuando la finalización de un programa se debe a la existencia de errores.	C
SUBR23R3	Recuperación de mensajes	C
SUBR40R3	Movimiento de datos de doble byte y supresión de caracteres de control.	C
SUBR41R3	Movimiento de datos de doble byte y adición de caracteres de control.	C

A.2 Opciones de depuración del compilador

Esta sección explica cada uno de los parámetros de depuración. Consulte "Ejemplos de utilización de opciones de depuración del compilador" en el tema A.3 para obtener ejemplos sobre la información de depuración que estos parámetros pueden solicitar.

Subtemas

- A.2.1 Valor *SOURCE para el parámetro OPTION
- A.2.2 Valor *XREF para el parámetro OPTION
- A.2.3 Valor *DUMP para el parámetro OPTION
- A.2.4 Valor *LIST para el parámetro GENOPT
- A.2.5 Valor *ATR para el parámetro GENOPT
- A.2.6 Valor *XREF para el parámetro GENOPT
- A.2.7 Valor *DUMP para el parámetro GENOPT
- A.2.8 Valor *PATCH para el parámetro GENOPT
- A.2.9 Valor *OPTIMIZE para el parámetro GENOPT
- A.2.10 Parámetro ITDUMP
- A.2.11 Parámetro SNPDUMP
- A.2.12 Parámetro CODELIST
- A.2.13 Parámetro PHSTRC

A.2.1 Valor *SOURCE para el parámetro OPTION

Un valor de ***SOURCE** para el parámetro OPTION solicita un listado del programa fuente RPG/400. El valor por omisión es ***SOURCE**.

A.2.2 Valor *XREF para el parámetro OPTION

Un valor de ***XREF** para el parámetro OPTION solicita un listado de referencias cruzadas y una tabla de información de campos clave (cuando proceda). Consulte el Capítulo 3, "Compilación de un programa en RPG/400" para obtener una descripción de este listado. El valor por omisión es ***XREF**.

A.2.3 Valor *DUMP para el parámetro OPTION

Un valor de ***DUMP** para el parámetro OPTION hace que se imprima el contenido de las principales áreas de datos, como **VCOMMON**, la tabla de nombres de archivo, la tabla de nombres de campo, e **IT** (texto intermedio). Esta impresión sólo se produce si la compilación finaliza anómalamente. Por lo tanto, generalmente ***DUMP** se especifica cuando vuelve a intentarse una compilación que finalizó anómalamente. El valor por omisión es ***NODUMP**.

A.2.4 Valor *LIST para el parámetro GENOPT

Un valor de ***LIST** para el parámetro GENOPT hace que se listen la IRP, su código hexadecimal asociado, y cualquier mensaje de error. El valor por omisión es ***NOLIST**.

A.2.5 Valor *ATR para el parámetro GENOPT

Un valor de ***ATR** para el parámetro GENOPT hace que se listen los para el parámetro IRP. El listado incluye las descripciones de campo y los números de sentencia en los que se definen. El valor por omisión es ***NOATR**.

A.2.6 Valor *XREF para el parámetro GENOPT

Un valor de ***XREF** para el parámetro GENOPT hace que se imprima un listado de referencias cruzadas de todos los objetos definidos en la IRP cuando la compilación finaliza.

A.2.7 Valor *DUMP para el parámetro GENOPT

Un valor de ***DUMP** para el parámetro GENOPT hace que se liste la plantilla del programa. El valor por omisión es ***NODUMP**.

A.2.8 Valor *PATCH para el parámetro GENOPT

Un valor de ***PATCH** para el parámetro GENOPT reserva espacio en el programa compilado para un área de parches del programa. El área de parches del programa puede utilizarse para la depuración. El tamaño del área de parches se basa en el tamaño del programa generado. El valor por omisión es ***NOPATCH**.

*A.2.9 Valor *OPTIMIZE para el parámetro GENOPT*

Un valor de ***OPTIMIZE** para el parámetro GENOPT hace que el compilador genere un programa que se ejecuta con más eficacia y que requiere menos almacenamiento. Sin embargo, el hecho de especificar ***OPTIMIZE** puede incrementar sustancialmente el tiempo necesario para crear un programa. Los programas existentes pueden optimizarse con el mandato CHGPGM CL.

A.2.10 Parámetro ITDUMP

En el mandato CRTRPGPGM, el parámetro ITDUMP origina un listado dinámico de texto intermedio producido por una fase especificada. Un listado dinámico significa que el texto intermedio se imprime durante la compilación mientras el texto intermedio se crea y almacena. En el mandato CRTRPTPGM, el parámetro ITDUMP hace que se imprima un flujo de las principales rutinas que se ejecutan en una o más fases especificadas.

En el parámetro ITDUMP pueden especificarse hasta 25 fases, cada una de ellas identificada por los dos últimos caracteres de su nombre. La lista debe ir entre paréntesis. Por ejemplo, el parámetro ITDUMP siguiente origina un listado dinámico del texto intermedio producido por **Q**RGSE, **Q**RGSO, y **Q**RGSC: **ITDUMP(S**ESOSC).

A.2.11 Parámetro SNPDUMP

El parámetro SNPDUMP produce un listado de las principales áreas de datos y el texto intermedio a continuación de la ejecución de una o más fases especificadas.

En el parámetro SNPDUMP pueden especificarse hasta 25 fases, cada una de ellas identificada por los dos últimos caracteres de su nombre. La lista debe ir entre paréntesis. Por ejemplo, el parámetro SNPDUMP siguiente origina el listado de texto intermedio producido por **QRGSI**, **QRGSC**, y **QRGSO** y también hace que se imprima el contenido de las principales áreas de datos: **SNPDUMP(SISCO)**.

A.2.12 Parámetro CODELIST

El parámetro CODELIST origina el listado dinámico de la IRP producido por una fase especificada. El listado dinámico significa que la IRP se imprime durante la compilación mientras se procesa la fase especificada.

En el parámetro CODELIST pueden especificarse hasta 25 fases, cada una de ellas identificada por los dos últimos caracteres de su nombre. La lista debe ir entre paréntesis. Por ejemplo, el parámetro CODELIST siguiente origina un listado dinámico de la IRP producida por **QRGGC**, **QRGGO**, y **QRGEC**:
CODELIST(GCGOEC) .

A.2.13 Parámetro PHSTRC

El parámetro PHSTRC especifica si va a producirse o no un rastreo de fase durante la compilación. Un rastreo de fase consiste en la impresión en el listado del compilador de los nombres de las fases del compilador, en el orden en que se ejecutan estas fases. También se listan los números de los mensajes RXT (como las cabeceras del compilador) a medida que van recuperándose.

Los valores que pueden codificarse para el parámetro PHSTRC son ***YES** y ***NO**. ***NO** es el valor por omisión.

A.3 Ejemplos de utilización de opciones de depuración del compilador

La Figura 203 muestra ejemplos de la información de depuración que pueden solicitarse en el mandato CRTRPGPGM mediante las opciones de depuración del compilador. El listado del compilador de la Figura 203 se imprimió con un mandato CRTRPGPGM que especificaba los parámetros de depuración siguientes:

```
GENOPT(*LIST *DUMP) ITDUMP(SC) SNPDUMP(GO)
CODELIST(GO) PHSTRC(*YES)
```

El parámetro PHSTRC(*YES) hace que se imprima el nombre de una fase cuando se procesa esta fase. Por ejemplo, A en la Figura 203 muestra que la fase **QRGSF** procesó las especificaciones de descripción de archivos, la fase **QRGSE** procesó las especificaciones ampliadas, la fase **QRGSI** procesó las especificaciones de entrada y la fase **QRGSC** procesó las especificaciones de cálculo.

El parámetro ITDUMP(SC) hace que se imprima el texto intermedio que la fase **QRGSC** creó y almacenó. (Vea la B de la Figura 203.)

El parámetro CODELIST(GO) hace que se imprima la IRP producida por la fase **QRGGO** cuando finaliza dicha fase. (Vea la C de la Figura 203.)

El parámetro SNPDUMP(GO) hace que se imprima el contenido de las principales áreas de datos al finalizar la fase **QRGGO** y hace que se imprima el texto intermedio producido por **QRGGO**. (Vea la D de la Figura 203.)

El valor *LIST del parámetro GENOPT hace que se imprima la IRP y las instrucciones de máquina al finalizar la compilación. (Vea la E de la Figura 203.) Las cabeceras de este listado IRP indican la información siguiente:

SEQ: Una numeración secuencial de las sentencias IRP. Los mensajes de error, como los errores de sintaxis de la IRP emitidos por el supervisor de resolución del programa, utilizan este número para hacer referencia a las sentencias IRP erróneas.

INST: Una numeración secuencial de las instrucciones de máquina generada a partir de las sentencias IRP. No todas las sentencias IRP hace que se generen instrucciones de máquina. El número de instrucción puede utilizarse como punto de interrupción para las funciones de depuración OS/400. Consulte el Capítulo 4, "Mensajes de error, pruebas y depuración" o la publicación *CL Guía del Programador* para obtener más información sobre los puntos de interrupción.

GENERATED CODE: Instrucciones de máquina que se han generado para las sentencias IRP.

GENERATED OUTPUT: Sentencias IRP.

BREAK: Puntos de interrupción en la IRP que pueden utilizarse para los puntos de parada en las funciones de depuración OS/400. Consulte el Capítulo 4, "Mensajes de error, pruebas y depuración" o el manual *CL Guía del Programador* para obtener más información sobre los puntos de interrupción. Si el punto de interrupción es un número, éste indica la sentencia fuente RPG/400 desde la que se generó la sentencia IRP.

El valor *DUMP para el parámetro GENOPT hace que se imprima la plantilla del programa cuando finaliza la compilación. (Vea la F de la Figura 203.)

Puede codificarse un valor de *DUMP para el parámetro OPTION para hacer que se imprima el contenido de las principales áreas de datos del compilador, si el compilador finaliza anómalamente. La Figura 204 muestra un ejemplo de la información impresa. Para este ejemplo, el mandato es:

```
CRTRPGPGM OPTION(*DUMP)
```

```
RXT0001
RXT0002
RXT0003
RXT0004
RXT0028
  5763RG1 V3R0M5 940125          IBM SAA RPG/400          QGPL/DATAE          01/25/94  14:40

RXT0005
  Compilador . . . . . :  IBM SAA RPG/400
RXT0020
  Opciones del mandato:
RXT0023
RXT0070
```


RXT9132
5763RG1 V3R0M5 940125 IBM SAA RPG/400 QGPL/DATAE 01/25/94 14:

RXT9123
R e s u m e n d e m e n s a j e s

RXT9131
* QRG6103 Gravedad: 00 Número: 1
Mensaje : No se ha especificado ningún indicador de
desbordamiento pero se ha asignado un indicador a un archivo
y se ha generado un salto automático a 6.

RXT9131
* QRG7031 Gravedad: 00 Número: 1
Mensaje : No se hace referencia al nombre ni al indicador.

RXT9124
* * * * * F I N D E R E S U M E N D E M E N S A J E S * * * * *

C
;/* FASE - QRGGO FECHA - 09/25/92 *//SVP*/

/***** GEN TIME - O *****/ZCOMMENT*/

;ENTRY .OFL INT/*ZLABX*/

;DCL INSPTR .OFLWRTN/*OVERFLOW RETURN POINTER*//ZDCLRTN*/

;BRK '1201 '/*BRK POINT*//ZBRKPT*/

;CMPBLA(B) *IN1A,*OFF/ EQ(.OOR0001)/*COND IND TST*//ZCONID*/

;SPACE 3
;/* QSYSPRT FILE OUTPUT *//ZFLCOM*/

;SETSPP .FIBPTR,.F01FIB/*SET FIB PTR*//ZFIBPT*/

;CPYBLA .CURROP, C'WRITE'/* SET OP *//ZCURROP*/

;CPYBLA .PRTCTL,X'0000000000060000'/*SET OPERATION*//ZPRC*/

;CPYBWP .BUFPTR,.U01BUFO/*LOCATE BUFFER*//ZBUFADD*/

;CPYBLA .BUFFER(1:0132),.BLANKS /*ZCLRBUF*/

;CPYBLA .F01XSET,'00000WRITEF*OFL 1201 '/*FILL FEEDBACK*/
;CPYBLA .F01EIND,*OFF/*NO ERROR IND*//ZFIBPT0*/

;CMPBLA(B) .F01OPEN,*OFF/EQ(.DMEXL2)/*FILE OPEN?*/ZFIBPT1*/

;CALLI .XRVPW01,*,.DRIVRTN/*ZPUT*/

;.OOR0001:/*LABEL*//ZLAB*/

;CPYBLA .OFL2A(2:2),'00'/*SET INTERNAL IN D OFF*//ZOF2OFF*/

;B .OFLWRTN/*RETURN*//ZOFRTN*/

5763RG1 V3R0M5 940125 IBM SAA RPG/400 QGPL/DATAE 01/25/94 14:

;ENTRY .EXCPT INT/*ZLABX*/

;DCL INSPTR .LINERTN/*EXCEPTION RETURN POINTER*//ZDCLLRT*/

;BRK '1100 '/*BRK POINT*//ZBRKPT*/

;SPACE 3
;/* QSYSPRT FILE OUTPUT *//ZFLCOM*/

;SETSPP .FIBPTR,.F01FIB/*SET FIB PTR*//ZFIBPT*/

;CPYBLA .CURROP, C'WRITE'/* SET OP *//ZCURROP*/

;CMPBLA(B) *IN1A,*ON/NEQ(.OEC0001)/*OVERFLOW ?*/
;CALLI .FETCH01,*,.OFETCHR/*FETCH OVERFLOW*/
;.OEC0001:/*RET/BR TGT*//ZPRTOF*/

;CPYBLA .PRTCTL,X'0000000100000000'/*SET OPERATION*//ZPRC*/

;CPYBWP .BUFPTR,.U01BUFO/*LOCATE BUFFER*//ZBUFADD*/

;CPYBLA .BUFFER(1:0132),.BLANKS /*ZCLRBUF*/

Ejemplos de utilización de opciones de depuración del compilador

```

00008      /* FASE - QRGD1   FECHA - 11/26/87 */
           /*SVP*/

           /* FASE - QRGAE   FECHA - 11/26/87 */
           /*SVP*/

00009
00010  0001 000004 0252 0009 0257      SETIEXIT .RPGXIEX,.RPXIEXP      /*SET UP I
GM*/

00011  0002 00000A 1011 0234          B .START      /*BEGIN OF
00012      .STOP:

00013  0003 00000E 30B2 0007 000D    CPYBLA .INVOC SW,*OFF      /*ALLOW CA
00014  0004 000014 02A1 0000        RTX *      /* RETURN
/*ZSTAR*/

           /*START OF THE PROGRAM*/

           /* STATIC AREA FOR INDIC/FIELDS */

           /* START OF STRUCTURE POINTED BY .DMPTRLT*/

00017
00018      DCL DD .RPGPGM CHAR(1) BDRY(16) INIT
00019      DCL DD .FIRSTSW CHAR(1) INIT('0')      /* PROGRAM
/
/*PROGRAM
00020      DCL DD .EOJSW CHAR(1) INIT('0')
00021      DCL DD .DUMPSW CHAR(1) INIT('0')
00022      DCL DD .ERRTERM CHAR(1) INIT('0')
00023      DCL DD .INVOC SW CHAR(1) INIT('0')
00024      DCL DD .INVOCER CHAR(4) INIT('8888')
00025      DCL SYSPTR .RPGXIEX INIT('QRGXIN VX', TYPE(PGM,1))
/* INVOCATI
/
/*INTERFAC
00026      DCL DD .INVXLVL BIN(2) INIT(1)
INVX*/
/*END OF S
TURE*/

00027      DCL DD .BLANKS CHAR(140) INIT((140)' ')
00028      DCL CON *ON CHAR(1) INIT('1')      /* SET/CHE
*/

```

```

5763SS1 01/25/94 14:40:34          SALIDA GENERADA
MSGID          DIAGNOSTICOS SEMANTICOS DE CORRIENTES DE INSTRUCCIONES MI
5763SS1 01/25/94 14:40:34          SALIDA GENERADA
DESPLAZAMIENTO          I TEMPLATE DISPLA
00000000 00003272 00000000 0201C4C1 E3C1C540 40404040 40404040 40404040 40404040
00000020 40404040 40404040 C0000000 00000000 00000000 00010000 00000000 00000000
00000040 00000000 00000000 000502FD DD000400 00000000 00000000 00000000 00000000
00000060 208000FC 00000000 00000000 019602B1 00000100 00000C14 0000171C 00000008
00000080 0000015E 00003114 00000000 00000000 00000A4B 000026C8 00000000 00000196 00000000
000000A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
000000C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
000000E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000100 00000AD2 02520009 02571011 023430B2 0007000D 02A10000 21320050 020D10B2
00000120 021F000C 1042020B 20021022 022F0059 0283004C 022D0000 1C464000 021623CA
00000140 026A1C46 C0000216 20000059 10B2021F 000D1011 005C30B2 021F000D 10B2002E
00000160 002D10B2 002D005A 10B20231 20D710B2 0232002D 1CC24000 0005000C 02691CC2
00000180 4000005B 000C0269 10110061 30B20030 005E10B2 00120014 11930013 00141CC2
000001A0 40000016 000D005F 10BE0015 000C1011 006430BE 0015000D 1CC24000 0060000C
000001C0 006410B2 0060000C 10110068 30B20231 20D710B2 0232002D 10B2003A 00AB10B2
000001E0 003B00E9 1CC24000 005B000C 006210B2 005B000C 10110269 30B20232 00631011
00000200 026930B2 00300065 30B20030 006730B2 00300069 1CC24000 0016000C 02660293
00000220 00740000 006A30B2 0030006C 10B20016 000C1042 006D2001 3C461000 006D006F
00000240 00700293 007A0000 007B3143 006D2001 1011006E 3143001B 00711011 007230B2
00000260 00300073 1011005D 3CC24000 0013000D 00780082 00840154 10B20022 007510B2
00000280 01C80076 01320087 01A010B2 608F2001 2084000B 10B20190 007710B2 016C000D
000002A0 1CC24000 0174000D 01FF0293 01ED0000 009030B2 60112002 20020079 1011006A
000002C0 20820084 015410B2 0022007C 1CC2C000 0013000C 007D0293 00830000 008130B2
000002E0 01C8007E 01320087 01A010B2 608F2001 2084000B 11430019 200110A3 608F2001
00000300 20040019 007F1197 608F2004 200120F0 10A2608F 20012004 204010B2 01900080
00000320 10B2016C 000D1CC2 40000174 000D01FF 029301ED 00000090 3011007B 22930074
00000340 0000006A 30110081 21320085 00AC0132 0086011B 1042008E 409D2001 013201CD
00000360 0489008E 013201CF 00AE0132 01CE00AF 104201D4 01C91042 01D601CA 104201D3
00000380 01CB1042 01D501CC 10B20220 000C0283 01CD01D0 000010B2 0220000D 10B20141
000003A0 01420083 0088011F 014410B2 01100135 1CC24000 00C4000C 01D10132 01D900B2
000003C0 10B201DB 000C3042 01D701C9 104201D8 01CB0293 01E00000 01C73042 01D701CA
000003E0 104201D8 01CC0293 01E00000 01C73011 01D13C46 400001D8 200001E7 1C462000
00000400 01D800B8 01E41C46 100001D8 00B801E1 1C461000 00B800B9 01E31011 01E43C46
00000420 900001D8 00B901E3 1C461000 00B800B9 01E310B2 81DA01D9 01E23042 00B801D8
00000440 101101E7 3CC24000 00DE01E5 01E310B2 81DA01D9 01E61011 01E33C46 400001D7
00000460 200001C7 1C461000 00B800B9 01EA1143 00B801D7 1C469000 00B800B9 01C710B2
00000480 81DA01D9 01E81011 01C73011 01C73143 00B801D7 1C469000 00B800BA 01C71147
000004A0 00B800BA 101101C7 3CC24000 0174000D 01FF0293 01DD0000 01D13011 009030B2
000004C0 0220000D 10B200E1 022110B2 00E301F1 1CC24000 01EF000C 01FB1CC2 400000DD
000004E0 01F201FD 1CC24000 00DD01F3 01FD0132 008500AC 1CE2C000 011B0000 01F410B2
00000500 00CB000D 10B200C1 000D10B2 00C2000C 10B200DB 000C10B2 00D2000C 10B200DA
00000520 000D1011 01F730B2 01410142 00830088 011F0105 10B20110 01351011 01F730B2

```

Ejemplos de utilización de opciones de depuración del compilador

```

00000540 00DC01F6 30B20225 000D10B2 0220000D 10B2003A 00AB10B2 003B00E9 10B20231
00000560 20C610B2 023300DC 01320213 00841CC2 400000C3 000C0090 10114202 00B33CC2
00000580 200000DC 01F90268 1CC21000 00DC01FA 02681011 009030B2 00DC01FC 10B20231
000005A0 20C610B2 023300DC 01320213 008410B2 003A00AB 10B2003B 00E91011 01F830B2
000005C0 00DC01FE 101101F7 30B200DC 02001011 01F730B2 01410142 00830088 011F0105
000005E0 10B20110 01351011 01F73CC2 C0000007 000C0235 1042022B 200210B2 02400008
00000600 1011026B 30B20007 000C10B2 00300236 10B2002F 02371042 022B2000 1CC24000
00000620 0003000C 023C0547 020E0000 20010132 00270000 1042020B 20020283 0230022C
00000640 00001C46 40000216 23CA0282 03EF0298 029C2001 10B20028 003F0082 02520245
00000660 1042020B 20010283 02550254 000010B2 0232002D 1C46C000 02162000 026A10B2
00000680 01C5002C 10B201C6 00361042 00232001 10220090 023A2132 00844204 002310B2
000006A0 00E90239 10B200C3 000D1011 025E3D47 C0000023 20010238 10B2021E 000B10B2
000006C0 002F023B 10B201EF 000D1042 022B2000 23EF0298 029C2001 1CC24000 0003000C
000006E0 005D10B2 0003000C 10110072 3042020B 20020283 023D0243 00001C46 40000216
El programa DATAE se coloca en la biblioteca QGPL. 00 es el código de gravedad de error más alto.
RXT0048
* * * * * F I N D E C O M P I L A C I O N * * * * *

```

Figura 203. Ejemplos de la información de depuración del compilador

IMAGEN 40

Figura 204. Ejemplo de la información impresa en un vuelco del compilador

A.4 Diseño de la IRP

El compilador RPG/400 genera un código que se ejecuta en un ciclo predeterminado del programa. El programador puede modificar este ciclo, hasta cierto punto, mediante operaciones de bifurcación, subrutinas de proceso y subrutinas de excepción y de gestión de errores. Dado que el ciclo es similar para todos los programas del usuario, los segmentos en común del código aparecen en la IRP generada. Saber qué hacen estos segmentos y cómo localizarlos en un listado de la IRP puede ayudarle a relacionar un programa de usuario con la IRP que el compilador genera.

La Figura 205 identifica las etiquetas que aparecen en la mayoría de los listados de la IRP, el orden en que aparecen normalmente y la función que realiza la IRP en la etiqueta especificada. Un listado de la IRP normalmente está compuesto de muchas páginas, y contiene más etiquetas y más IRP de las que se indican en la Figura 205. Esta figura está pensada para ser utilizada como un directorio general dentro de un listado IRP.

IMAGEN 41

IMAGEN 42

Figura 205. Diseño de la IRP

A.5 Programa generador automático de informes

El programa generador automático de informes consta de las fases que se listan en la Tabla 23. Las fases que tienen una U en la tercera columna de la Tabla 23 se procesa incondicionalmente, y las fases que tienen una C en la tercera columna sólo se procesan si el programa que está compilándose las requiere. La Figura 206 muestra el orden en el que se ejecutan las fases.

Las principales áreas de datos del programa generador automático de informes son un área común (**XREGN**), una tabla de nombres de campo (**FLDTBL**), y dos almacenamientos intermedios (**BUFFER1** y **BUFFER2**).

Tabla 23. Fases del programa generador automático de informes		
Nombre fase	Descripción de la fase	Llamada: Incondicionalmente (U) Condicionamente (C)
QRPT0000	Fase de interfaz del mandato	U
QRPT0001	Fase de control de raíz	U
QRPT0002	Fase de control de E/S y función /COPY	U
QRPT0003	Fase de diagnóstico para *AUTO	C
QRPT0004	Fase de diagnóstico para *AUTO	C
QRPT0005	Fase de generación *AUTO	C
QRPT0008	Fase de reiniciación cíclica	U

IMAGEN 43

Figura 206. Orden de proceso de las fases del programa generador automático de informes

Este apéndice contiene información sobre las funciones adicionales del RPG/400 que no están disponibles utilizando el compilador RPG II del AS/400 compatible con el Sistema/36.

Subtemas

B.1 Mejoras en el lenguaje

B.1 Mejoras en el lenguaje

Las mejoras en el lenguaje del RPG/400 respecto al compilador del RPG II del AS/400 compatible con el Sistema/36 son :

- *Datos descritos externamente:* Los campos de un archivo se describen al sistema OS/400 mediante las especificaciones de descripción de datos (DDS). La ventaja de datos descritos externamente es que los campos del archivo necesitan describirse sólo una vez al sistema OS/400 y no necesitan describirse para cada programa que utilice el archivo. Si se modifica la descripción del archivo, el programador puede modificar la descripción en un lugar y a continuación volver a compilar los programas que la utilicen.

Los siguientes códigos de operación para el proceso de archivos están disponibles para el lenguaje de programación del RPG/400:

- **DELET** (Suprimir registro)
- **SETGT** (Establecer mayor que)
- **UPDAT** (Modificar registro existente)
- **WRITE** (Crear nuevos registros)
- **REDPE** (Recuperar registro anterior igual).

Los siguientes códigos de operación de control están disponibles en el lenguaje de programación del RPG/400:

- **CLOSE** (Cerrar archivos)
- **FEOD** (Forzar fin de datos)
- **OPEN** (Abrir archivo para proceso).

- *Soporte de estación de trabajo:* Permite la especificación del nombre de dispositivo del RPG/400, **WORKSTN**, utilizado para entrada y salida desde la estación de trabajo de pantalla. Los códigos de operación que dan soporte al control directo sobre formatos específicos de estación de trabajo incluyen:

- **EXFMT** (Ejecutar formato)
- **WRITE** (Crear nuevos registros)

Además el lenguaje de programación del RPG/400 puede dar soporte a un subarchivo en soporte **WORKSTN** si utiliza los códigos de operación siguientes:

- **CHAIN** (Recuperación al azar desde un subarchivo)
- **READC** (Leer siguiente registro modificado)
- **UPDAT** (Modificar registro existente)
- **WRITE** (Crear nuevos registros)

Consulte el Capítulo 8, Utilización de archivos WORKSTN para obtener más información sobre el dispositivo **WORKSTN**.

- *Operaciones ANDxx/ORxx:* Le permiten especificar una condición de decisión más compleja que una simple comparación de A con B, con las operaciones **IFxx**, **DOUxx**, y **DOWxx**.
- *Operación CABxx (comparar y bifurcar):* Le permite realizar una comparación y bifurcar en una operación eliminando la necesidad de establecer y luego comprobar indicadores resultantes.
- *Código de operación CLEAR:* Le permite establecer los elementos de una estructura de datos de una variable en cero, en blancos o 0 (para indicadores), dependiendo del tipo de campo.
- *Control de compromiso:* Le permite agrupar las operaciones de archivo utilizando los códigos de operación **COMIT** y **ROLBK**. Consulte el Capítulo 6, "Control de compromiso" para obtener más información.
- *Función /COPY del compilador:* Permite la fusión de miembros de más de un archivo fuente durante la compilación. No puede efectuarse ninguna clasificación o modificación de registros.
- *Operaciones de área de datos.* Las operaciones **IN** (Recuperar un área de datos) y **OUT** (Grabar un área de datos) le permiten acceder a un área de datos y, opcionalmente, bloquear y desbloquear un área de datos. La operación **UNLCK** (Desbloquear un área de datos) desbloquea una o todas las áreas de datos de un programa.
- *Estructuras de datos:* El lenguaje de programación del RPG/400 da soporte a dos estructuras de datos adicionales:
 - La estructura de datos de estado del programa, la cual proporciona información al programa sobre las condiciones de excepción/error del programa

- Estructuras de datos descritos externamente.

- Inicialización de estructura de datos:* Le permite inicializar estructuras de datos y subcampos en blanco, cero o un valor específico.
- Se da soporte a los *tipos de datos SAA de fecha, hora e indicación de la hora.*
- Operación DSPLY (Función de pantalla):* Permite acceder a un dispositivo de pantalla sin utilizar un archivo de dispositivo de pantalla y permite acceder al manejador de mensajes.
- DSPPGMREF:* La información del objeto al que se hace referencia proporcionada mediante el mandato de CL **DSPPGMREF** incluye ahora los programas llamados además de los archivos y las áreas de datos (no las estructuras de datos).
- Operación ENDyy:* Proporciona una mejor lectura del programa, permitiéndole indicar el tipo de estructura (**CASxx**, **DO**, **DOUxx**, **DOWxx**, **IFxx** o **SELEC**) que la operación **END** está cerrando.

Nota: Cuando el campo del resultado es demasiado pequeño para contener el valor del límite (factor 2) más el incremento (END factor 2), un desbordamiento de datos numéricos puede producir que el program en RPG/400 repita un bucle constantemente. Sin embargo, un programa en Sistema/36 RPG II repetirá el bucle el número requerido de veces que especifiquen los valores de arranque, límite e incremento.

- Manejo de la condición de excepción/error:* Le permite controlar la lógica del programa utilizando la subrutina ***PSSR** de excepción/error de programa si se produce una condición de excepción/error mientras se está ejecutando el programa. Consulte "Manejo de excepciones/errores" en el tema 4.8 para obtener información detallada sobre el manejo de la condición de excepción/error.
- Constantes figurativas:* Le permiten utilizar palabras reservadas del RPG/400 adicionales que se pueden especificar, sin especificar la longitud y las posiciones decimales porque la longitud y posiciones decimales de una constante figurativa son las mismas que las del campo asociado. Estas son:

- ***ALL'a..'**
- ***ALLX'x1x2..'**
- ***HIVAL**
- ***LOVAL**
- ***ON**
- ***OFF**

- Códigos de edición de signo menos flotantes:* Se proporcionan cuatro códigos de edición nuevos (**N**, **O**, **P**, y **Q**) para editar números negativos con un signo menos flotante (-). El signo menos, si está especificado, se imprime a la izquierda del dígito más significativo o del símbolo de moneda flotante.
- Se da soporte a *tipos de datos SAA gráficos.*
- Se da soporte a *literales hexadecimales.*
- Indicadores referenciados como datos (*IN, *INxx):* Le permiten un método alternativo de hacer referencia a y de manipular indicadores. La matriz de indicadores para los indicadores 01 a 99 (***IN**) y el campo de indicador (***INxx**) reduce la codificación y proporciona un método simplificado a muchas necesidades del proceso del programa.
- Barras de sangrado:* Le permiten especificar que las sentencias **DO** y **SELEC**, y las cláusulas **IF-ELSE** se sangren en los listados de los programas para mejorar la lectura del programa.
- Subrutina de inicialización:* Le permite especificar una subrutina en particular para ejecutarla en la inicialización del programa.
- Operaciones ITER y LEAVE:* La operación **ITER** le permite finalizar la iteración actual de un grupo **DO** y comenzar la iteración siguiente. La operación **LEAVE** le permite transferir el control desde un grupo **DO** a la sentencia siguiente a la operación **ENDyy** correspondiente.
- Operaciones KLIST (Definir Clave Compuesta)/KFLD (Definir Partes de una Clave):* Le permiten indicar el nombre por el que puede especificarse una clave compuesta y los campos que comprende la clave compuesta.
- Estructuras de datos de apariciones múltiples:* Una estructura de

datos puede aparecer n veces en un programa. Consulte "Estructuras de datos de apariciones múltiples" en el tema 9.2.3 para obtener más información sobre estructuras de datos de apariciones múltiples.

- Constante con nombre:* Le permite especificar un nombre para una constante. Este nombre representa un valor específico que no se puede cambiar cuando se está ejecutando el programa. Puede especificar constantes con nombre en el Factor 1 y en el Factor 2 de las especificaciones de cálculo y en los campos Constante o Palabra de edición de las especificaciones de salida. Consulte "Constantes con nombre" en el tema 10.1 para obtener más información acerca de la utilización de constantes con nombre.
- Soporte de campos que permiten nulos.
- Variables numéricas:* El lenguaje de programación del RPG/400 da soporte a variables numéricas de 30 dígitos como máximo inclusive. El número máximo de dígitos decimales permitido sigue siendo 9.
- Amplificador de Operación (posición 53):* N le permite especificar los registros de lectura sin bloquearlos. Cinco operaciones dan soporte a esta función: **READ**, **READE**, **READP**, **REDPE** y **CHAIN**. P le permite rellenar el campo del resultado después de realizar una operación **CAT**, **SUBST**, **XLATE**, **MOVE**, **MOVEL** o **MOVEA**.
- Indicadores de desbordamiento:* Puede utilizar los indicadores 01 a 99 como indicadores de desbordamiento indistintamente en archivos **PRINTER** descritos en el programa o descritos externamente (además de los 0A a 0G y 0V para los archivos descritos por el programa). Consulte Capítulo 5, "Consideraciones generales sobre archivos" para obtener más información sobre los indicadores de desbordamiento.
- Opción de control de impresora (PRTCTL):* Puede (1) especificar operaciones de espaciado y de salto dinámicamente en lugar de utilizar valores en las especificaciones de salida y (2) acceder al valor de la línea actual en el programa. Esta opción sólo está permitida con archivos descritos por el programa. Consulte Capítulo 5, "Consideraciones generales sobre archivos" para obtener más información sobre las opciones de **PRTCTL**.
- Parámetro de inicialización del programa.* Le permite transferir parámetros en un programa prearrancado.
- Area de datos de parámetros de inicialización del programa.* Le permite predefinir y almacenar los parámetros de inicialización del programa.
- Código de operación REDPE:* Le permite recuperar el registro previo en secuencia de un archivo controlado en cálculo si la clave del registro coincide con el argumento de búsqueda del Factor 1 (posiciones 18 a 27). También debe especificar un nombre de archivo o un nombre de registro en el Factor 2. Asimismo, puede especificar una estructura de datos en el campo del resultado en la que pueda leerse el registro.
- Código de operación RESET:* Le permite volver establecer los elementos en una estructura de datos, o de un campo, en sus valores originales al final de la inicialización del programa. Cuando se especifica **RESET** para una estructura o una variable, se toma una instantánea de esa variable o estructura al final del ciclo ***INIT**. A continuación se utiliza el valor para restablecer la estructura o variable.
- Indicadores resultantes con operaciones MOVE y MOVEL:* Puede especificar indicadores resultantes en las sentencias **MOVE** y **MOVEL**. Esto elimina la necesidad de operaciones adicionales para comprobar las condiciones de blanco, cero o positivo/negativo.
- Reintento en tiempo excedido:* Se ha actualizado el mensaje de error **RPG1218** para permitir que se solicite un reintento cuando se produce un tiempo excedido en una solicitud de bloqueo de registro.
- La operación **SELEC** le permite especificar las condiciones para seleccionar qué grupo de operaciones va a procesarse.
 - La operación **SELEC** marca el comienzo del grupo **SELEC**.
 - La operación **WHxx** de un grupo **SELEC** le permite determinar el lugar donde se transfiere el control después de procesar la operación **SELEC**.
 - La operación **OTHER** le permite especificar la secuencia de operaciones a procesar si no se satisface la condición **WHxx**.
- Archivos SEQ:* Le permiten ejecutar entrada/salida secuencial a

cualquier archivo organizado secuencialmente, tal como archivos de base de datos, en diskette, en cinta, de salvar o de impresora. El dispositivo realmente utilizado está especificado por un mandato de alteración temporal del programa Programa de control de AS/400.

- Archivo SPECIAL con operación PLIST:* Le permite especificar un dispositivo de entrada/salida que no está soportado directamente por el lenguaje de programación del RPG/400. Puede añadir parámetros adicionales a la lista de parámetros creada por el RPG/400 mediante el uso de los códigos de operación **PLIST** y **PARM**. Consulte el "Archivo especial" en el tema 5.14 para obtener información sobre el dispositivo **SPECIAL**.
- Operaciones de serie CAT, CHECK, CHEKR, SCAN, SUBST y XLATE:..* La operación **CAT** le permite concatenar dos series de caracteres. El campo del resultado puede ser un nombre de campo, elemento de matriz, estructura de datos o nombre de tabla. La operación **SCAN** le permite explorar en una serie de caracteres por si hay una subserie especificada que empiece en una ubicación específica para una longitud específica. La operación **SUBST** le permite extraer una subserie de una serie fuente especificada que empiece en una ubicación específica. La operación **XLATE** le permite traducir los caracteres del factor 2 de acuerdo a las series FROM y TO del factor 1. La operación **CHECK** le permite verificar si cada carácter del factor 2 está entre los caracteres válidos del factor 1. La operación **CHEKR** proporciona una función similar a **CHECK** pero en dirección contraria (de derecha a izquierda).
- SUBR23R3:* La subrutina de recuperación de mensajes se ha ampliado para permitir al sistema recuperar un máximo de 3000 caracteres de texto de segundo nivel y soportará el mensaje **I** que contiene 0-9 o A-F para el identificador de mensaje.
- Inicialización de subcampos:* Le permite inicializar un subcampo de estructuras de datos en un valor específico.
- Operación TESTN (Probar Numérico):* Le permite comprobar la validez de un campo de caracteres para asegurar que contiene dígitos decimales con zona y blancos.
- Códigos de edición definidos por el usuario (5 a 9):* Le permiten la edición orientada para un cliente específico o nacionalmente. Los códigos de edición definidos por el usuario se definen al sistema AS/400.
- La operación **TIME**: Permite 14 dígitos.
- Operación UNLCK:* Permite que se desbloquee el último registro bloqueado para un archivo de actualización de disco. Pueden desbloquearse aún los registros para procesar operaciones de salida definidas por especificaciones de salida sin nombres de campo incluidos.
- Se da soporte a *campos de longitud variable*.
- Se da soporte al *año de 4 dígitos*.

Nota: Para obtener más información sobre las mejoras del RPG/400, consulte la publicación *RPG/400 Reference*.

C.0 Apéndice C. Comunicación de datos

Las operaciones del RPG/400 del AS/400 permiten la comunicación de datos mediante un archivo WORKSTN que utilice ICF. No hay operaciones o especificaciones del RPG/400 exclusivas para las comunicaciones de datos. Los tipos de comunicaciones de datos a los que el archivo WORKSTN da soporte utilizando ICF incluyen APPC, Asíncrona, BSCEL, Financiero, Comunicaciones entre sistemas, Detalle y SNUF. El archivo WORKSTN que se utiliza para comunicación de datos debe definirse como un archivo controlado en cálculo (**F** en la posición 16 de las especificaciones de descripción de archivo). A continuación se muestra una lista de algunos de los códigos de operación y las funciones de comunicación de datos correspondientes a las que da soporte el archivo WORKSTN:

Códigos de operación	Funciones de comunicación de datos
OPEN (Abrir archivo para proceso)	Abrir (entrada y salida), recuperación de errores
CLOSE (Cerrar archivos)	Cerrar (permanentemente), recuperación de errores
EXFMT (Ejecutar formato)	Grabar/Leer (Esperar)
READ (Leer un registro)	Leer (Esperar)
WRITE (Crear nuevos registros)	Grabar (Esperar)
ACQ (Adquirir)	Adquirir un dispositivo, recuperación de errores
REL (Liberar)	Liberar un dispositivo, recuperación de errores

Para obtener más información acerca de comunicaciones remotas, consulte la publicación *ICF Programmer's Guide*. Para obtener más información sobre las operaciones del RPG/400, consulte la publicación *RPG/400 Reference*.

Subtemas

- C.1 Manejo de excepciones y errores con archivos ICF
- C.2 Recuperación de errores de comunicaciones

C.1 Manejo de excepciones y errores con archivos ICF

Cuando un programa tiene un error en tiempo de ejecución, puede cancelar el programa. (Consulte "Manejo de excepciones/errores" en el tema 4.8.) De hacer esto, todos los archivos del programa se cierran anormalmente. Para los archivos ICF, se avisa al otro extremo de la línea de comunicaciones de que se ha producido una anomalía y las comunicaciones han terminado anormalmente. Para un archivo ICF compartido, el aviso se envía cuando el último programa cierra el archivo.

En lugar de cancelar un programa, puede continuar el proceso (por ejemplo, en una subrutina de manejo de errores). La recuperación del error es responsabilidad del usuario.

C.2 Recuperación de errores de comunicaciones

Cuando utilice un archivo de múltiples dispositivos, puede recuperar un error de comunicaciones de un dispositivo procesando una operación **REL** (Liberar) seguida de una **ACQ** (Adquirir) para el dispositivo que tiene el error.

Un error de comunicaciones de un archivo puede recuperarse procesando una operación **CLOSE** seguida de una operación **OPEN** para el archivo del error. En archivos compartidos, el archivo tiene que estar cerrado para todos los programas que comparten el archivo y luego abrirse nuevamente.

Para obtener más información, consulte la publicación *ICF Programmer's Guide*.

D.0 Apéndice D. Archivos de gestión de datos distribuidos (DDM)

La gestión de datos distribuidos (DDM) le permite acceder a archivos de datos que residen en sistemas remotos con una red de comunicaciones que de soporte a DDM. El compilador del REG/400 da soporte a los archivos DDM: puede recuperar, añadir, actualizar o suprimir registros de datos en un archivo que resida en otro sistema.

Para obtener más información acerca del acceso a archivos remotos, consulte la publicación *DDM Guide*.

E.0 Apéndice E. Opción del entorno del Sistema/38 del compilador RPG

Este apéndice describe cómo la opción de entorno del Sistema/38 del compilador RPG da soporte a la misma sintaxis de RPG que el compilador RPG III del Sistema/38 y a los convenios de nomenclatura de los objetos del Sistema/38. El resto del apéndice trata de las diferencias entre el RPG del Sistema/38 RPG y la opción de entorno del Sistema/38 del compilador del RPG, de las diferencias entre la opción de entorno del Sistema/38 del compilador en RPG y el compilador en RPG del sistema AS/400 y los tipos de archivo a los que cada compilador da soporte.

Subtemas

- E.1 Diferencias entre el RPG III del Sistema/38 y la Opción del entorno del Sistema/38 del compilador RPG
- E.2 Diferencias entre la Opción del entorno del Sistema/38 del compilador RPG y el compilador del RPG/400
- E.3 Tipos de archivo soportados por cada compilador

E.1 Diferencias entre el RPG III del Sistema/38 y la Opción del entorno del Sistema/38 del compilador RPG

La opción de entorno de Sistema/38 del compilador RPG se diferencia del compilador RPG III del Sistema/38 en lo siguiente:

- El nombre del miembro fuente del mandato de creación se utiliza como nombre del archivo de salida en spool que contiene la salida del compilador.
- El formato de la fecha utilizado cuando se ejecuta el programa está en el formato descrito en el valor del trabajo (establecido por la descripción del trabajo o por el mandato CHGJOB) en lugar del valor del sistema.
- Se permiten las matrices numéricas en el código de operación **MOVEA**.
- Se da soporte a 30 dígitos numéricos.
- No se da soporte a dispositivos de tarjetas. Si el fuente en RPG especifica cualquiera de la sintaxis de dispositivo de tarjetas, se producirá un error de gravedad 30 al compilar el programa.
- El formato del listado es diferente.
- El formato del mensaje es diferente para los mensajes en tiempo de compilación y de ejecución.

Para saber cuáles son los elementos iguales que en el soporte de RPG III del Sistema/38, consulte la publicación *System/38 RPG III Reference Manual and Programmer's Guide*, SC21-7725.

E.2 Diferencias entre la Opción del entorno del Sistema/38 del compilador RPG y el compilador del RPG/400

Utilice RPG38 como tipo de fuente para el miembro que contiene las sentencias fuente del RPG. Los programas creados tendrán los mismos valores en el atributo del objeto.

Si está utilizando directamente el mandato CRTRPGPGM o CRTRPTPGM, asegúrese de utilizar el mandato en la biblioteca QSYS38.

La mayor parte de la información de esta Guía del Usuario se aplica a la opción de entorno de Sistema/38 del soporte de compilador RPG con las excepciones siguientes:

- Las mejoras del compilador del RPG/400 con relación a la opción de entorno de Sistema/38 del compilador RPG son:
 - *Código de operación CLEAR:* Le permite establecer los elementos de una estructura de datos de una variable en cero, en blancos o 0 (para indicadores), dependiendo del tipo de campo.
 - Se da soporte a los tipos de datos SAA de fecha, hora e indicación de la hora.
 - *DSPPGMREF:* La información del objeto al que se hace referencia proporcionada mediante el mandato de CL **DSPPGMREF** incluye ahora los programas llamados además de los archivos y las áreas de datos (no las estructuras de datos).
 - *Operación ENDyy:* Proporciona una mejor lectura del programa, permitiéndole indicar el tipo de estructura (**CASxx**, **DO**, **DOUxx**, **DOWxx**, **IFxx** o **SELEC**) que la operación **END** está cerrando.
 - *Constantes figurativas:* Le permiten utilizar palabras reservadas del RPG/400 adicionales que se pueden especificar, sin especificar la longitud y las posiciones decimales porque la longitud y posiciones decimales de una constante figurativa son las mismas que las del campo asociado. Estas son:
 - ***ALLX'x1x2..'**
 - ***ON**
 - ***OFF**
 - *Códigos de edición de signo menos flotantes:* Se proporcionan cuatro códigos de edición nuevos (**N**, **O**, **P**, y **Q**) para editar números negativos con un signo menos flotante (-). El signo menos, si está especificado, se imprime a la izquierda del dígito más significativo o del símbolo de moneda flotante.
 - Se da soporte a tipos de datos SAA gráficos.
 - Se da soporte a literales hexadecimales.
 - *Barras de sangrado:* Le permiten especificar que las sentencias **DO** y **SELEC**, y las cláusulas **IF-ELSE** se sangren en los listados de los programas para mejorar la lectura del programa.
 - *Operaciones ITER y LEAVE:* La operación **ITER** le permite finalizar la iteración actual de un grupo **DO** y comenzar la iteración siguiente. La operación **LEAVE** le permite transferir el control desde un grupo **DO** a la sentencia siguiente a la operación **ENDyy** correspondiente.
 - *Constante con nombre:* Le permite especificar un nombre para una constante. Este nombre representa un valor específico que no puede cambiarse cuando se ejecuta el programa. Consulte "Constantes con nombre" en el tema 10.1 para obtener más información acerca de la utilización de constantes con nombre.
 - Se da soporte a campos que permiten nulos.
 - *Amplificador de Operación (posición 53):* 'N' le permite especificar registros de lectura sin bloquearlos. Cinco operaciones dan soporte a esta función: **READ**, **READE**, **READP**, **REDPE** y **CHAIN**. 'P' le permite rellenar el campo de resultado después de efectuar una operación **CAT**, **SUBST**, **XLATE**, **MOVE**, **MOVEL** o **MOVEA**.
 - *Alterar temporalmente para archivo de impresión (OVRPRTF):* Al utilizar el compilador del RPG/400, el mandato CL, **OVRPRTF** produce un archivo de spool con el mismo nombre que el **archivo** del mandato **OVRPRTF** que se va a generar cuando se ejecute el programa. El nombre de archivo de spool generado se puede cambiar entonces especificando un valor para la palabra clave **SPLFNAME** del mandato **OVRPRTF**. La palabra clave **SPLFNAME** no está disponible en el

entorno de Sistema/38.

- *Parámetros de inicialización del programa (PIP)*. Le permite transferir parámetros en un programa prearrancado.
 - *Area de datos de parámetros de inicialización del programa (PDA)*. Le permite predefinir y almacenar los parámetros de inicialización del programa.
 - *Código de operación REDPE*: Le permite recuperar el registro previo en secuencia de un archivo controlado en cálculo si la clave del registro coincide con el argumento de búsqueda del Factor 1 (posiciones 18 a 27). También debe especificar un nombre de archivo o un nombre de registro en el Factor 2. Asimismo, puede especificar una estructura de datos en el campo del resultado en la que pueda leerse el registro.
 - *Código de operación RESET*: Le permite volver establecer los elementos en una estructura de datos, o de un campo, en sus valores originales al final de la inicialización del programa. Cuando se especifica **RESET** para una estructura o una variable, se toma una instantánea de esa variable o estructura al final del ciclo ***INIT**. A continuación se utiliza el valor para restablecer la estructura o variable.
 - *Reintento en tiempo excedido*: Se ha actualizado el mensaje de error **RPG1218** para permitir que se solicite un reintento cuando se produce un tiempo excedido en una solicitud de bloqueo de registro.
 - La operación **SELEC** le permite especificar las condiciones para seleccionar qué grupo de operaciones va a procesarse.
 - La operación **SELEC** marca el comienzo del grupo **SELEC**.
 - La operación **WHxx** de un grupo **SELEC** le permite determinar el lugar donde se transfiere el control después de procesar la operación **SELEC**.
 - La operación **OTHER** le permite especificar la secuencia de operaciones a procesar si no se satisface la condición **WHxx**.
 - *Operaciones de serie CAT, CHECK, CHEKR, SCAN, SUBST y XLATE*: **CAT** le permite concatenar dos series de caracteres. El campo del resultado puede ser un nombre de campo, elemento de matriz, estructura de datos o nombre de tabla. **SCAN** le permite explorar si hay una subserie especificada que empiece en una ubicación específica para una longitud específica dentro de una serie de caracteres. **SUBST** le permite extraer una subserie de una serie fuente especificada que empiece en una ubicación específica. **XLATE** le permite traducir caracteres del factor 2 de acuerdo con las series FROM y TO en el factor 1. La operación **CHECK** le permite verificar si cada carácter del factor 2 está entre los caracteres válidos del factor 1. La operación **CHEKR** proporciona una función similar a **CHECK** pero en dirección contraria (de derecha a izquierda).
 - *SUBR23R3*: La subrutina de recuperación de mensajes se ha ampliado para permitir al sistema recuperar un máximo de 3000 caracteres de texto de segundo nivel y soportará el mensaje **I** que contiene 0-9 o A-F para el identificador de mensaje.
 - *Operación TIME*: Permite 14 dígitos.
 - *Operación UNLCK*: Permite que se desbloquee el último registro bloqueado para un archivo de actualización de disco. Pueden desbloquearse aún los registros para procesar operaciones de salida definidas por especificaciones de salida sin nombres de campo incluidos.
 - Se da soporte a *campos de longitud variable*.
 - Se da soporte al *año de 4 dígitos*.
- Los nombres de archivo y de programa deben seguir los convenios de nomenclatura del Sistema/38 (objeto.biblioteca) en la sentencia **/COPY** y en los códigos de operación **FREE**, **CALL** y **DSPLY**.
 - La opción de entorno de Sistema/38 del compilador de RPG le permite grabar en un número relativo de registro existente mientras que el compilador de RPG/400 no soporta esta función y producirá un error en tiempo de ejecución.

- El formato de la información devuelta desde una operación **POST** a un dispositivo específico es el mismo que el del soporte del Sistema/38.
- Los mandatos de creación son los mismos que en el Sistema/38.

La Tabla 24 muestra las diferencias entre el compilador del RPG/400 y la opción de entorno de Sistema/38 de los entornos del compilador RPG.

Tabla 24. Diferencias entre el compilador del RPG/400 la opción de entorno del Sistema/38 del compilador RPG				
Parámetro del Compilador de RPG	Opción del entorno del Sistema/38 del compilador RPG Parámetro	Opciones del Compilador de RPG/400	Opción del entorno del Sistema/38 del compilador RPG Opciones	Comentarios
REPLACE	N/A			Parámetro nuevo
		*YES	N/A	Opción nueva
		*NO	N/A	Opción nueva
TGTRLS	N/A			Parámetro nuevo
		*CURRENT	N/A	Opción nueva
		*PRV	N/A	Opción nueva
		nivel de release	N/A	Opción nueva
AUT	PUBAUT			AUT sustituye a PUBAUT
		*LIBCRTAUT	N/A	Opción nueva
		*CHANGE	*NORMAL	*CHANGE sustituye a *NORMAL
		*USE	N/A	Opción nueva
		*ALL	N/A	Opción nueva
		*EXCLUDE	*NONE	*EXCLUDE sustituye a *NONE
		nombre de lista autoriz.	N/A	Opción nueva
PGM	PGM			Parámetro existente
		*CURLIB	N/A	Opción/valor por omisión nuevo
SRCFILE	SRCFILE			Parámetro existente
		*CURLIB	N/A	Opción nueva
PRTFILE	PRTFILE			Parámetro existente
		*CURLIB	N/A	Opción nueva
OPTION	OPTION			Parámetro existente
		*SECLVL	N/A	Opción nueva
		*NOSECLVL	N/A	Opción nueva

Diferencias entre la Opción del entorno del Sistema/38 del compilador RPG y el compilador del RPG/400

SAAFLAG	N/A			Parámetro nuevo
		*NOFLAG	N/A	Opción nueva
		*FLAG	N/A	Opción nueva
INDENT	N/A			Parámetro nuevo
		*NONE	N/A	Opción nueva
		valor de carácter	N/A	Opción nueva
CVTOPT	N/A			Parámetro nuevo
		*NONE	N/A	Opción nueva
		*VARCHAR	N/A	Opción nueva
		*DATETIME	N/A	Opción nueva
		*GRAPHIC	N/A	Opción nueva
ALWNULL	N/A			Parámetro nuevo
		*NO	N/A	Opción nueva
		*YES	N/A	Opción nueva

Cuando convierte un programa compatible con el Sistema/38 a un programa AS/400, puede, y en algunos casos debe, utilizar el soporte tal como está documentado en este manual para la lista de elementos anterior.

E.3 Tipos de archivo soportados por cada compilador

De forma similar a los programas, pueden crearse los archivos (tipo de objeto ***FILE**) con un atributo del Sistema/38 o del sistema OS/400. Los archivos del Sistema/38 tienen un 38 añadido a su atributo de objeto. Los archivos del sistema OS/400 no lo tienen. Consulte la publicación *System/38 Environment Programmer's Guide/Reference* para obtener información sobre los atributos de objeto para los distintos tipos de archivo.

Cualquier tipo de un programa del RPG/400 puede utilizar archivos creados con cualquiera de los dos atributos. Los programas del compilador RPG de la opción de entorno de Sistema/38 no están restringidos a utilizar los archivos del Sistema/38. Los programas en RPG/400 no están restringidos a utilizar archivos OS/400. Por ejemplo, un programa del compilador RPG de la opción de entorno de Sistema/38 puede utilizar un archivo de pantalla del sistema OS/400 o un archivo de base de datos. Un programa en RPG/400 del sistema AS/400 puede utilizar el archivo de pantalla o un archivo de base de datos del Sistema/38.

La capacidad de mezclar tipos de archivos y tipos de programas se aplica también a los tipos de archivo de comunicaciones para un programa en en RPG/400 aunque los tipos de archivo del Sistema/38 sean diferentes de los tipos de archivo del sistema de OS/400. Un programa en AS/400 puede utilizar un archivo de comunicaciones BSC o mixto de dispositivo.

Hay algunos elementos dignos a tener en cuenta en tales combinaciones. Un programa del AS/400 que utiliza un archivo BSC, de comunicaciones o mixto de dispositivo utiliza los valores ***STATUS** del RPG/400 del AS/400. Algunos de estos valores se establecen en base a códigos de retorno principal/secundario. Además, el formato de la información que devuelve una operación **POST** a un dispositivo específico es la versión en RPG/400 del AS/400. Algunos de los elementos como el nombre de ubicación remota, no se devuelven porque los tipos de archivo no dan soporte a un nombre de ubicación remota.

Nota: No se da soporte a archivos ICF con una opción de entorno de Sistema/38 del programa compilador en RPG/400. La utilización de un archivo ICF en el entorno de Sistema/38 puede producir resultados imprevisibles.

I*

Figura 207. Creación de una matriz utilizando campos como índices

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
E*
E* Este ejemplo muestra cómo se cargan dieciocho elementos de 5
E* caracteres de la matriz MA1 con dos líneas de especificación
E* únicamente.
E*
E...Arch(de)Arch(a)+NombreN/rN/tbLonPDSNommatLonPDSComentarios+++++*
E          MA1          30  5
E*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
I*
I* En estas especificaciones de entrada, los restantes elementos de
I* MA1 se cargan uno después de otro hasta que se llena la matriz.
I* Cada elemento adicional se codifica en una línea separada. Cada
I* nuevo registro requiere un medio distinto de identificación. Por
I* ejemplo, si otro registro 03 sigue al primero, los campos en el
I* segundo registro podrían recubrir los campos leídos desde el
I* primer registro. Este método es muy útil para matrices pequeñas.
I*
INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
IFILE1 AA 03 100 C1
I.....PDesdeA+++DNomcamL1MlFrP1NeCe...*
I          1  90 MA1
I          BB 04 100 C2
I          1  5 MA1,19
I          6 10 MA1,20
I          "
I          " +-----+
I          " | Más Elementos Matriz|
I          " +-----+
I          "
```

Figura 208. Creación de una matriz utilizando índices fijos

```
*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* Las especificaciones de este ejemplo calculan tres niveles de
C* totales. A medida que van leyéndose desde los registros de entrada,
C* los campos CAMPOA, CAMPOB, CAMPOC Y CAMPOD se suman a los totales
C* de primer nivel L1A, L1B, L1C y L1D. Estos totales de primer
C* nivel se suman en el momento de la ruptura de control de L1, con
C* los totales L2A, L2B, L2C y L2D. Análogamente, cuando se produce
C* una ruptura de control L2, los totales de segundo nivel se suman
C* a los totales de tercer nivel L3A, L3B, L3C y L3D. Además, al
C* producirse las rupturas de control, se efectúa la salida de totales
C* L1, L2 y L3, y se ponen a ceros los campos de totales una vez que
C* se han grabado en el dispositivo de salida.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgcomentarios+++*
C          CAMPOA  ADD  L1A      L1A      62      SUM.A TOT.L1
C          CAMPOB  ADD  L1B      L1B      62
C          CAMPOC  ADD  L1C      L1C      62
C          CAMPOD  ADD  L1D      L1D      62
CL1       L1A      ADD  L2A      L2A      62      SUM.A TOT.L2
CL1       L1B      ADD  L2B      L2B      62
CL1       L1C      ADD  L2C      L2C      62
CL1       L1D      ADD  L2D      L2D      62
CL1       L1A      ADD  L2A      L2A      62      SUM.A TOT.L2
CL2       L2B      ADD  L3B      L3B      62
CL2       L2C      ADD  L3C      L3C      62
CL2       L2D      ADD  L3D      L3D      62
C*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdN01N02N03Nomexc.....*
O          T 20      L1
O.....N01N02N03Campo+YBUposPConstante/palabraedición+++*
O          L1A      KB  15
O          L1B      KB  30
O          L1C      KB  45
O          L1D      KB  60
O          T 20      L2
O          L2A      KB  15
```

```

O          L2B  KB  30
O          L2C  KB  45
O          L2D  KB  60
O      T 20    L3
O          L3A  KB  15
O          L3B  KB  30
O          L3C  KB  45
O          L3D  KB  60
    
```

Figura 209. Cálculo de totales sin matrices

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
E*
E* Los tres niveles de totales mostrados en este ejemplo se calculan
E* con matrices.
E*
E....Arch(de)Arch(a)+NombreN/rN/tbLonPDSNommatLonPDSComentarios++++*
E          SL1          4 6 2          L1 TOTALS
E          SL2          4 6 2          L2 TOTALS
E          SL3          4 6 2          L3 TOTALS
E*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
C*
C* Advierta la reducción del código requerido para especificar las
C* funciones. Por ejemplo, la rotura de control L1 en las
C* especificaciones de cálculo siguientes rellenan la misma función
C* que las 4 líneas de L1 en las especificaciones de cálculo mostradas
C* en el ejemplo anterior.
C*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgcomentarios+++*
C          CAMPOA      ADD  SL1,1      SL1,1      SUM.TOT.L1
C          CAMPOB      ADD  SL1,2      SL1,2
C          CAMPOC      ADD  SL1,3      SL1,3
C          CAMPOD      ADD  SL1,4      SL1,4
CL1         SL1         ADD  SL2         SL2         SUM.TOT.L2
CL1         SL1         ADD  SL2         SL2         SUM.TOT.L2
C*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
O*
O* De forma similar, se reducen las especificaciones de salida de 15
O* líneas a 6. El método de utilizar matrices da como resultado dos
O* posiciones únicamente entre elementos de la matriz.
O*
ONombre++DFADSaSDN01N02N03Nomexc.....*
O      T 20    L1
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O          SL1  KB  60
O      T 20    L1
O          SL2  KB  60
O      T 20    L1
O          SL3  KB  60
O*
    
```

Figura 210. Cálculo de totales con matrices

La figura siguiente muestra un ejemplo de utilización de matrices para dar formato a la salida del campo.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
E....Arch(de)Arch(a)+NombreN/rN/tbLonPDSNommatLonPDSComentarios++++*
E          MAA          4 5 0
E          MAB          5 10
E          MAC          6 4 2
E*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
INombarchSeNOIrPos1NCCPos2NCCPos3NCC.....*
IIN      AA  01  80 C
I      OR  02  80 C1
I.....PDesdeA+++DNomcamL1M1FrP1NeCe...*
I          51  74 MAC
I          1  20 MAA      01
I          1  50 MAB      02
I*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
    
```

```

ONombre++DFADSaSdN01N02N03Nomexc.....*
OOUT      D 1      01
O          OR      02
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O          MAC      84 '0 . &CR'
O          01      MAA,1 Z 89
O          02      MAB,X1 100
O*
    
```

IMAGEN 44

Figura 211. Utilización de matrices para dar formato a la salida de campo

La figura siguiente muestra un método de impresión de un elemento de la matriz por línea en el dispositivo de salida de impresora.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
E...Arch(de)Arch(a)+NombreN/rN/tbLonPDSNommatLonPDSComentarios++++*
E          MA2      5 21 15 0
E*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
CLR          DO 21      IN      30      HACER 21 VECES
CLR          EXCPTTOTAL
CLR          END
C*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OARFILE E 1          TOTAL
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O          MA2,IN      20
O*
    
```

Figura 212. Impresión de un elemento de matriz por línea

La figura siguiente muestra un método de impresión de más de un elemento de la matriz por línea en el dispositivo de salida de impresora.

```

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
E...Arch(de)Arch(a)+NombreN/rN/tbLonPDSNommatLonPDSComentarios++++*
E          MA1      6 10 10
E          MA2      6 50 10
E*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
CL0N01N02N03Factor1+++Oper+Factor2+++ResultLonDHMaMeIgComentarios+++*
C          DO 50      IN      20      HACER 50 VECES
C          MOVEAMA2,IN MA1      MOVER A MATRIZ
C          EXCPTTOTAL      IMPRIMIR
C          END 10      SUMAR 10 A IN
C*

*.. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..*
ONombre++DFADSaSdN01N02N03Nomexc.....*
OARFILE E 1          TOTAL
O.....N01N02N03Campo+YBUposPConstante/palabraedición++...*
O          MA1      B 100
O*
    
```

Figura 213. Impresión de más de un elemento de matriz por línea

G.0 Apéndice G. Glosario de abreviaturas

Abreviatura	Significado	Definición
CL	Lenguaje de Control	Conjunto de todos los mandatos con los que un usuario solicita funciones.
CPF	Programa bajo licencia de soporte al sistema Sistema/38.	Proporciona muchas funciones que están integradas completamente en el sistema tales como gestión de trabajos, base de datos, gestión de datos, control de trabajos, manejo de mensajes, seguridad, ayudas de programación y servicios. El recurso equivalente en el sistema AS/400 es el sistema OS/400.
ICF	Función para las comunicaciones entre sistemas	Una función del sistema OS/400 que permite que un programa se comunique interactivamente con otro programa o sistema.
DBC	Caracteres de doble byte (DBC) Variables	Variables que pueden contener datos de doble byte (ideogramas en japonés, chino simplificado, chino tradicional y coreano).
DDS	Especificaciones de Descripción de Datos	Una descripción de los archivos de dispositivo de la base datos del usuario que se entra en el sistema utilizando una sintaxis fija. A continuación se utiliza la descripción para crear archivos.
DDM	Gestión de Datos Distribuidos	Una función del sistema operativo que permite a un programa de aplicación o a un usuario de un sistema fuente tener acceso a archivos de datos en sistemas remotos conectados mediante una red de comunicaciones que utilice también DDM.
GDDM	Gestor de Representación Gráfica de Datos	Un grupo de rutinas que permiten definir imágenes y visualizarlas correctamente mediante rutinas de función que se corresponden con los gráficos primitivos. Compare con <i>Rutinas de Gráficos de Presentación</i> .
OS/400	N/A	Sistema operativo para el sistema AS/400. Proporciona muchas funciones que está totalmente integradas en el sistema. Entre estas se incluyen gestión de trabajos, base de datos, gestión de datos, control de trabajos, manejo de mensajes, seguridad, ayudas de programación y servicio.
PGR	Rutinas de Gráficos de Presentación	Un grupo de rutinas que permiten definir y visualizar correctamente gráficos de gestión mediante rutinas de función. Compare con <i>Gestor de Representación Gráfica de Datos</i> .
SNA	Arquitectura de Red de Sistemas	Descripción de la estructura lógica, formatos, protocolos y secuencias operativas para la

Guía del usuario RPG/400
Apéndice G. Glosario de abreviaturas

		<p>transmisión de unidades de información y el control de la configuración y la operación de las redes de la Arquitectura de Red de Sistemas.</p> <p>Nota: La estructura por capas de la SNA permite que los extremos origen y destino de la información (o sea los usuarios finales) sean independientes y no estén afectados por servicios y recursos específicos de la red SNA utilizados para el intercambio de información.</p>
SQL	Lenguaje de Consulta Estructurada	Un lenguaje que puede utilizarse dentro de programas escritos en otros lenguajes o de forma interactiva, para tener acceso a los datos del gestor de base de datos y para controlar el acceso a los recursos del gestor de la base de datos.

BACK_1 Bibliografía

Guía de Publicaciones, GC10-9237 (GC41-9678), que contiene una breve descripción de cada manual de la biblioteca AS/400 así como información sobre la forma de pedir publicaciones adicionales.

Guía para la Gestión de Datos, SC10-9008 (SC41-9658), que contiene información sobre aspectos clave en la gestión del sistema.

Data Description Specifications Reference, SC41-9620, que describe las especificaciones de descripción de datos que se utilizan para describir archivos.

Distributed Data Management Guide, SC41-9600, que contiene información acerca de comunicaciones remotas para el programador del RPG/400.

Guía para la Base de Datos, SC10-9009 (SC41-9659), que contiene una explicación detallada de la estructura de base de datos del AS/400. Este manual describe también cómo utilizar las palabras clave de las especificaciones de descripción de datos (DDS).

Communications: Intersystem Communications Function Programmer's Guide, SC41-9590, que proporciona la información que necesita el programador de aplicaciones para escribir aplicaciones que utilicen comunicaciones del AS/400 y el archivo Función para las comunicaciones entre sistemas.

Programming: GDDM Programming Guide SC41-0536, y *Programming: GDDM Programming Reference, SC41-0537*, que proporcionan directrices sobre el Gestor de Representación Gráfica de Datos (GDDM) para programadores que necesitan escribir aplicaciones gráficas.

Programming: System/38 Environment Programmer's Guide and Reference, SC41-9755, que describe la migración desde el Sistema/38 y la conversión al sistema AS/400.

Instalación de Software, SC10-9279 (SC41-3120), que describe cómo instalar el programa bajo licencia RPG/400 en el sistema.

Operación del Sistema, SC10-9280 (SC41-3203), que describe cómo operar con el sistema AS/400.

SAA Lenguaje de Consulta Estructurada SQL/400 Manual de Consulta, SC10-8997 (SC41-9608)*, que describe el SQL en el sistema AS/400.

Languages: RPG Reference Summary, SX09-1164, que contiene un resumen de los códigos de operación, códigos de edición, indicadores y códigos de estado.

RPG Debugging Template, GX21-9129, que proporciona una plantilla de las especificaciones en RPG.

RPG/400 Reference, SC09-1817, que proporciona información de consulta para el compilador en RPG/400.

Caracteres Especiales

/delimitador END EXEC 1.4.3
 /delimitador EXEC SQL 1.4.3
 * (asterisco)
 especificaciones generadas 12.0 12.4
 ** (doble asterisco)
 especificaciones generadas 12.0 12.4
 *** (triple asterico)
 especificaciones generadas 12.0 12.4
 *ATR, parámetro *GENOPT A.2.5
 *ENTRY PLIST 11.1.2
 *OPTIMIZE, parámetro GENOPT A.2.9
 *PATCH, parámetro GENOPT A.2.8
 *PRV 3.2
 *SOURCE 3.2 A.2.1

A

abreviaturas de términos G.0
 agrupación/desagrupación en bloques de registros 5.7
 alteración de orden de clasificación
 alteración temporal de descripción de archivo 5.10.5
 alteración temporal, archivo 5.10.5 7.4
 ampersand (&)
 función de copia del generador automático de informes 12.2.1.1
 archivo
 alteración temporal 5.10.5
 base de datos
 Véase archivo DISK
 bloqueo 5.5
 claves válidas 7.1.3
 compartimiento 5.8
 consideraciones generales 5.1
 de disco descritos externamente 7.1
 dependencia de dispositivo 5.1
 descrito externamente 5.3
 Véase también archivo descrito externamente
 descrito por programa 5.3 5.11
 diagrama de proceso
 archivo PRINTER 5.12.4
 archivo secuencial 5.13
 archivo SPECIAL 5.14
 DISK
 Véase archivo DISK
 formato de campo 9.1
 independencia de dispositivo 5.1
 indexado 7.2.1
 matriz
 Véase matriz
 nombre
 alteración temporal 5.10.5
 descrito externamente 5.3
 descrito por programa 5.11
 opciones abiertas 5.8
 PRINTER 5.12
 redireccionamiento 5.1
 SEQ 5.13 7.2.2
 SPECIAL 5.14
 supresión de registros de
 tipos
 soportados por el compilador del RPG en entorno del Sistema/38 E.3
 soportados por RPG/400 E.3
 WORKSTN 8.0
 archivo combinado 8.4.2.3
 archivo con números de registro relativos a direcciones de registro
 Véase archivo de direcciones de registros
 archivo de direcciones de registros
 con número relativo de registro 7.2.3
 con registros de límites 7.2.3.1
 límites en secuencial 7.2.3
 número relativo de registro 7.2.3
 archivo de direcciones de registros con número relativo de registro
 Véase archivo de direcciones de registros
 archivo de límites de direcciones de registros
 Véase archivo de direcciones de registros
 archivo de múltiples dispositivos 8.5
 archivo de referencia de campos, ejemplo de 7.1.2
 archivo de salida 8.4.2.2
 archivo descrito externamente
 adición a descripción externa 5.10
 alteración temporal 5.10.5
 cambio de nombre de formato de registro 5.10.2
 como archivo WORKSTN 8.2 8.2.1
 como descrito por programa 7.2.4
 definición 5.3
 especificaciones 5.10

- especificaciones de descripción de archivo para 5.10
- especificaciones de formato de registro 7.1.1
- especificaciones de salida para 5.10.6
- formato de datos 9.1
- métodos de proceso 7.1.4
- ventajas 5.3
- vía de acceso 7.1.2
- archivo descrito por programa 5.3 7.2.1.1
 - argumentos de búsqueda válidos 7.2.1.1
 - como archivo DISK 7.2
 - como archivo WORKSTN 8.4 8.4.1 8.4.2
 - definición 5.3
 - formato de los datos 9.1.1
- archivo DISK
 - códigos de operación de archivo permitidos
 - para métodos de proceso por clave 7.4
 - para métodos de proceso sin clave 7.4
 - descripción general 7.0
 - descrito externamente
 - como descrito por programa 7.2.4
 - descripción general 7.1
 - ejemplos 7.1.2
 - especificaciones de formato de registro 7.1.1
 - proceso 7.1.4
 - vía de acceso 7.1.2
 - descritos por programa
 - archivo de direcciones de registros 7.2.3
 - archivo indexado 7.2.1
 - archivo secuencial 7.2.2
 - proceso 7.3
 - diagramas de proceso
 - especificaciones de formato de registro 7.1
 - métodos de proceso
 - proceso al azar por clave 7.3.5
 - proceso consecutivo 7.3.2
 - proceso por número relativo de registro 7.3.1
 - proceso secuencial entre límites 7.3.4
 - proceso secuencial por clave 7.3.3
 - visión general 7.3
- archivo físico
- archivo indexado
 - argumentos de búsqueda válidos 7.2.1.1
 - descripción general 7.2.1
 - vía de acceso 7.2.1
- archivo lógico
 - Véase archivo DISK
- archivo PRINTER
 - acceder a valor línea actual 5.12.4
 - códigos de operación de archivo permitidos 5.12
 - desbordamiento de página 5.12.1
 - diagrama de proceso 5.12.4
 - indicadores de desbordamiento 5.12
 - lógica de obtención de desbordamiento 5.12.3
 - modificar control de formularios 5.12.4
 - número máximo de archivos permitidos en programa 5.12
 - PRTCTL (control de impresora) 5.12.4
- archivo secuencial 7.2.2
- archivo SEQ
 - códigos de operación de archivo permitidos 5.13
 - descripción general 5.13
 - diagrama de proceso 5.13
 - ejemplo 5.13
 - longitud variable 5.13
 - restricciones 5.13
- archivo SPECIAL
 - definición 5.14
 - descripción general 5.14
 - lista de parámetros 5.14
 - nombre de dispositivo 5.14
 - operaciones de archivo válidas 5.14
 - supresión de registros de 5.14
- archivo TAPE 7.2.2
- archivo WORKSTN
 - códigos de operación de archivo permitidos con 8.3
 - definición 8.0
 - descrito externamente 8.2
 - proceso 8.2.1
 - descrito por programa 8.4
 - archivo combinado 8.4.2.3
 - archivo de entrada 8.4.2.1
 - archivo de salida 8.4.2.2
 - con nombre de formato 8.4.1
 - consideraciones 8.4.1.4
 - especificaciones de cálculo 8.4.1.3

- especificaciones de entrada 8.4.1.2
- especificaciones de salida 8.4.1.1
- sin nombre de formato 8.4.2
- ejemplos 8.6
- indicadores de tecla de función con 8.2.2
- múltiples dispositivos 8.5
- proceso 8.3
- programa de consulta de ejemplo 8.6.1
- programa de ejemplo de consulta y búsqueda 8.6.5
- programa de ejemplo de entrada de datos 8.6.2
- programa de ejemplo de lectura con límite de tiempo de espera excedido 8.6.8
- programa de ejemplo de mantenimiento de datos 8.6.3
- programa de ejemplo de nombre de formato 8.6.6
- programa de ejemplo de proceso de subarchivo 8.6.4
- programa de ejemplo de línea de comienzo variable 8.6.7
- subarchivos
 - ejemplos 8.3.5.1
 - formato de registro 8.3.5
 - formato de registro de control 8.3.5
 - para archivo de dispositivo de pantalla 8.3.5
 - utilizaciones de 8.3.5.1
- utilización 8.0
- archivos
 - consideraciones generales 5.0
 - archivos DDM (gestión de datos distribuidos) D.0
 - archivos de dispositivo
 - archivos DISK 7.0
 - archivos PRINTER 5.12
 - archivos SEQ 5.13
 - archivos SPECIAL 5.14
 - archivos WORKSTN 8.0
 - dependencia de dispositivo 5.1
 - independencia de dispositivo 5.1
 - múltiples dispositivos 8.5
- área común
 - VCOMMON A.1.2
 - XREGN A.5
- área de datos de Parámetros de inicialización del programa
 - inicialización 10.2.4
- área de datos PIP (Parámetros de inicialización del programa)
 - inicialización 10.2.4
 - y áreas de datos (PDA) 11.4.1
- Área de datos RETURNCODE 3.1
- áreas de datos
 - cómo acceder 11.4
 - compilador A.1.2
 - desbloqueo
 - implícitas 9.3
 - grabación
 - implícitas 9.3
 - impresión A.2.3
 - información general 11.4
 - recuperación
 - implícitas 9.3
 - restricciones 11.4
 - RETURNCODE 3.1
- argumento de búsqueda
 - archivo descrito externamente
 - descripción 7.1.3
 - referencia a una clave parcial 7.1.3.2
 - válido 7.1.3.1
 - archivo descrito por programa 7.2.1.1
- asterisco (*)
 - especificaciones generadas 12.0 12.4
 - indicación en líneas de totales del generador automático de informes 12.3.1
- autorización de uso público sobre un objeto 3.2
- ayuda para el diseño de pantallas (SDA) 1.4.2
- ayudas de programación 12.5
- B**
- bibliografía BACK_1
- biblioteca de pruebas, utilización VERSION 4.3 4.4
- bifurcación condicional 1.6.2
- blanco después
- bloqueo
 - archivo 5.5
 - autónomo 5.6
 - bajo control de compromiso 6.1.4
 - lectura sin bloqueo 5.6
 - reintento en tiempo de espera excedido 5.6
 - tiempo de espera excedido del bloqueo del registro 5.6
 - UNLCK 5.6
- bloqueo de archivo 5.5
- bloqueo de registro 5.6
- BSC

Véase comunicaciones síncronas en binario
 bucle controlado
 descripción general 1.6.3
 operación DOU (Hacer hasta) 1.6.3.3
 operación DOW (Hacer mientras) 1.6.3.2

C
 cálculos generados por generador automático de informes VERSION 12.4.1
 cambio de nombre de formato de registro 5.10.2
 campo
 binario 9.1.3
 empaquetado 9.1.1
 en generador automático de informes 12.3.2
 formato en archivo 9.1
 campo binario 9.1.3
 definición 9.1.3
 especificaciones de subcampos de estructura de datos 9.2.1
 campo de control
 campos de coma flotante 5.10.4
 campos de datos de fecha (tipo de datos SAA) 10.3.2
 campos de datos de hora (tipo de datos SAA) 10.3.2
 campos de datos de indicación de la hora (tipo de datos SAA) 10.3.2
 campos de longitud variable 10.3.1
 campos numéricos
 centrado de encabezamientos de columna 12.3.2
 formato 9.1 9.1.2
 generador automático de informes
 carácter
 ciclo del programa
 control de compromiso 6.1.5
 inicialización 10.2.6
 lógica de obtención de desbordamiento 5.12.3
 clave
 compuesta 7.1.3.2
 para un registro o archivo 7.1.3
 parcial 7.1.3.2
 clave parcial 7.1.3.2
 métodos de proceso
 archivos DISK descritos externamente 7.1.4
 claves válidas
 para archivo 7.1.3
 para registros 7.1.3
 CODE/400
 opciones del compilador 3.2
 código de operación CALL (llamada a un programa)
 comunicación del programa 11.1.1
 consulta de nombres de programas llamados 11.1.1
 DSPPGMREF 11.1.1
 GDDM 11.1.3
 otros programas 11.1.1
 subrutina de recuperación de mensajes (SUBR23R3) 11.2.1
 traslado de datos de doble byte y supresión de caracteres de control (SUBR40R3) 11.2.3
 traslado de datos de doble byte y adición de caracteres de control (SUBR41R3) 11.2.4
 una subrutina 11.2
 código de operación CLEAR 10.2.2
 código de operación COMMIT (compromiso)
 con múltiples dispositivos 6.1.3
 consideraciones del sistema 6.1.3
 control de compromiso 6.1.3
 código de operación DO (hacer)
 gráfico de flujo y resumen 1.6.3.1
 código de operación DOUxx (hacer hasta)
 y programación estructurada 1.6.3.3
 código de operación DOWxx (hacer mientras)
 y programación estructurada 1.6.3.2
 código de operación ENDSR (finalizar subrutina)
 código de operación FREE (desactivar un programa)
 comunicación del programa 11.1.4
 código de operación PARM (identificar parámetros)
 comunicación del programa 11.1.2
 reglas para especificación 11.1.2.2
 código de operación PLIST (identificar una lista de parámetros)
 comunicación del programa 11.1.2
 reglas para especificación 11.1.2.1
 código de operación READ (leer un registro)
 con comunicación de datos C.0
 código de operación READC (leer el siguiente registro modificado)
 con subarchivo WORKSTN 8.3.5
 código de operación RESET 10.2.2
 código de operación RETRN (retorno al llamador)
 comunicación del programa 11.3
 código de operación WRITE (crear nuevos registros)
 con comunicación de datos C.0
 códigos de edición
 códigos de edición de signo menos flotantes, especificación B.1 E.2

códigos de operación 8.3 8.3.4
 permitido con archivo PRINTER 5.12
 permitido con archivo secuencial 5.13
 permitido con archivo SPECIAL 5.14
 permtido con archivo DISK 7.4
 códigos de operación de archivo
 permitido con archivo DISK 7.4
 permitido con archivo PRINTER 5.12 5.12.4
 permitido con archivo secuencial 5.13
 permitido con archivo SPECIAL 5.14
 permitido con archivo WORKSTN 8.3 8.3.4
 códigos de operación de clave compuesta
 Véase argumento de búsqueda
 comentarios
 sentencia /COPY 12.2
 comienzo del control de compromiso 6.1.1
 fin del control de compromiso 6.1.1
 compartimiento de archivo 5.8
 compartimiento de una vía de datos abierta para un archivo 5.8
 compilación
 CRTRPGPGM 3.0
 CRTRPTPGM 12.6
 programa fuente en RPG 3.0
 programa fuente generador automático de informes 12.6
 compilador
 áreas de datos A.1.2
 descripción de las fases A.1
 diferencias entre el RPG/400 y la Opción del entorno del Sistema/38 del compilador RPG E.2
 diferencias entre el Sistema/38 RPG III y la Opción del entorno del Sistema/38 del compilador RPG E.1
 directivas
 fases A.1.1 A.5
 información de servicio A.0
 Opción de entorno del Sistema/38 E.0
 opciones de depuración A.2 A.3
 organización de los mensajes de error A.1.3
 tipos de archivo, soportados E.3
 comprobación de nivel 5.4
 comprobación de secuencia
 en las especificaciones de entrada 5.11
 comunicación
 acceso a otros programas y sistemas 8.1
 con objetos del sistema 11.1
 datos C.0
 entre objetos 11.0
 manejo de la condición de excepción/error C.1
 recuperación de errores C.2
 comunicación de datos C.0
 comunicación del programa
 CALL/FREE/RETRN 11.1
 datos C.0
 retorno desde un programa llamado 11.3
 SPECIAL 5.14
 comunicación entre programas 11.1
 comunicaciones síncronas en binario C.0
 condición de archivo
 Véase external indicators, UC
 condicionamiento de archivos
 Véase indicadores externos
 condicionamiento de salida
 indicadores de desbordamiento 5.12.2
 consideraciones sobre archivos
 generales 5.0
 constantes
 Véase constantes con nombre
 constantes con nombre 10.1
 especificación 10.1 B.1
 reglas 10.1.1
 constantes figurativas
 Véase *ALL'X..',
 Véase *BLANK/*BLANKS
 Véase *HIVAL
 Véase *LOVAL
 Véase *ZERO/*ZEROS
 consulta de nombres de programas llamados 11.1.1
 control de compromiso 6.0 6.1
 CoOperative Development Environment/400
 opciones del compilador 3.2
 sentencia /COPY
 Véase generador automático de informes
 cuerpo de un informe 12.3.2.3
D
 D-*AUTO
 Véase también salida AUTO
 desbordamiento de líneas de impresión 12.3.2.4

- datos de bases de datos
 - campos de fecha 10.3.2
 - campos de hora 10.3.2
 - campos de indicación de la hora 10.3.2
 - campos de longitud variable 10.3.1
 - campos gráficos DBCS 10.3.3
 - valores nulos 10.3.4
- DDS
 - Véase la referencia a DDS
- depuración
 - discusión general 4.0
 - ejemplos A.3
 - utilización de DEBUG 4.6
 - utilización de DUMP 4.7
- desagrupación/agrupación en bloques de registros 5.7
- desbordamiento
 - indicadores 5.12.2
 - líneas de impresión D/T-*AUTO 12.3.2.4
 - página 5.12.1
- desbordamiento de página en archivo PRINTER 5.12.1
- diagrama de flujo
 - lógica de obtención de desbordamiento 5.12.3
- diseño de aplicaciones 1.7
- diseño de programa propio
 - Véase también diseño de aplicaciones
 - programación estructurada 1.6
 - visión general 1.5
- diseño del programa
 - aplicaciones 1.7
 - RPG 1.5
- dispositivo, nombre de función de 5.1
- doble asterisco (**)
- especificaciones generadas 12.0 12.4
- DS
 - Véase estructuras de datos
- DSPPGMREF 11.1.1 B.1
- *DUMP
 - parámetro GENOPT A.2.7
 - valor de parámetro GENOPT 3.2
 - valor de parámetro OPTION 3.2
 - valor del parámetro OPTION A.2.3
- E**
- edición del programa 11.3
- ejemplo de listado de fuente y referencia cruzada 4.2
- encabezamiento de página
 - AUTO 12.3.2.2
 - en generador automático de informes 12.3.2.1
- encabezamientos de columna en el generador automático de informes 12.3.2 12.3.2.3
- enlace a otros programas
 - CALL/FREE/RETRN 11.1
 - dispositivo SPECIAL 5.14
- entorno
 - Sistema/38 1.3 E.1
- entorno del Sistema/38
 - compilación de programa fuente 3.3
 - compilador del RPG/400 E.0
 - diferencias entre el RPG/400 y la Opción del entorno del Sistema/38 del compilador RPG E.2
 - diferencias entre el Sistema/38 RPG III y la Opción del entorno del Sistema/38 del compilador RPG E.1
 - entorno en el sistema AS/400 1.3
- entrada
 - archivo 8.4.2.1
 - diseño 1.5.3
- especificaciones
 - archivo descrito externamente 5.10
 - campos de entrada, modificación 12.2.1.2
 - copiadas, modificación 12.2.1
 - descripción de archivo 5.10.1
 - descripción de archivos, modificación 12.2.1.1
 - entrada y codificación 2.0
 - formato de registro 7.1.1
 - formatos 2.0
 - generadas 12.4
 - impresión por grupos 12.1.1
 - salida 5.10.6
 - salida, generadas por generador automático de informes 12.4.2
 - sentencia /COPY 12.2
 - sentencia de estructura de datos 9.2.2
 - subcampos de estructuras de datos 9.4
 - subcampos de estructuras de datos, reglas para 9.4.1
- especificaciones adicionales 2.1.3
 - descripción general 2.1.3 2.1.4
- especificaciones de cálculo 2.1.6
 - archivo WORKSTN descrito por programa 8.4.1.3
 - descripción general 2.1.6

- especificaciones de campos de entrada
 - formato binario 9.1.3
 - modificación 12.2.1.2
- especificaciones de campos de salida
 - formato binario 9.1.3
- especificaciones de control 2.1.1
 - descripción general 2.1.1
- especificaciones de descripción de archivo 2.1.2
 - control de compromiso 6.1.2
 - descripción general 2.1.2
 - modificación 12.2.1.1
 - opción de línea de continuación CFILE 8.1
 - para archivos descritos externamente 5.10.1
 - utilización de la función de copia con generador automático de informes 12.2.1
- especificaciones de descripción de datos (DDS)
 - Véase la referencia a DDS
- especificaciones de encabezamiento
 - Véase especificaciones de control
- especificaciones de entrada 2.1.5
 - archivo WORKSTN descrito por programa 8.4.1.2
 - descripción general 2.1.5
- especificaciones de salida
 - archivo WORKSTN descrito por programa 8.4.1.1
 - descripción general 2.1.7
- especificaciones generadas, generador automático de informes 12.0 12.4
- estructura de datos de estado del programa
 - apariciones múltiples de 9.2.3
 - información general 9.2.3
 - sentencia
 - descritos externamente 9.2
 - descritos por programa 9.2
 - especificaciones 9.2.2
 - reglas 9.2.2.1
 - subcampos
 - especificaciones 9.4
 - reglas 9.4.1
- estructura de datos de información de archivo
 - información general 9.3.2
 - sentencia
 - descritos externamente 9.2
 - descritos por programa 9.2
 - especificaciones 9.2.2
 - reglas 9.2.2.1
 - subcampos
 - especificaciones 9.3
 - reglas 9.4.1
- estructura de datos del área de datos
 - información general 9.3.1
 - inicialización 10.2.4
 - sentencia
 - descritos externamente 9.2
 - descritos por programa 9.2
 - especificaciones 9.2.2.1
 - reglas 9.2.2.1
 - subcampos
 - especificaciones 9.4
 - reglas 9.4.1
 - utilizadas para acceder a área de datos 11.4
- estructura de datos descrita externamente 9.5
- Estructura IF-THEN-ELSE 1.6.2
- estructuras de datos 9.0
 - apariciones múltiples de 9.2.2.1 9.2.3
 - área de datos 9.3.1
 - consideraciones especiales en la inicialización 10.2.4
 - descrita externamente 9.5
 - ejemplos de 9.5
 - ejemplos de inicialización 10.2.7
 - especiales 9.3
 - estado del programa 9.3.3
 - formato 9.2.1
 - información de archivo 9.3.2
 - información general 9.2
 - inicialización 10.2.3
 - longitud de 9.2.2.1
 - sentencia
 - Véase sentencia de estructura de datos
 - subcampos
 - solapamiento 9.4.1
 - tipos especiales
 - área de datos 9.3
 - estado de programa 9.3
 - información de archivo (INFDS) 9.3

F
fases

- compilador A.1.1
- programa generador automático de informes A.5
- rastreo A.2.12
- formato
 - proceso externo 9.1.4.1
 - proceso interno 9.1.4.2
 - signos 9.1.4
- formato binario
 - definición 9.1.3
 - especificación de campo de entrada 9.1.3
 - especificación de campo de salida 9.1.3
- formato de datos
 - binario 9.1.3
 - datos en campos 9.1
 - decimal con zona 9.1.1 9.1.2
 - decimal empaquetado 9.1.1
- formato de generador de informes (generador automático de informes) 12.3
- formato de registro
 - cambio de nombre 5.10.2
 - especificaciones para archivo descrito externamente 7.1.1
 - ignorado 5.10.3
 - para un subarchivo 8.3.5
- formato de registro de control, subarchivo 8.3.5
- formato decimal con zona
 - definición 9.1.2
 - descripción 9.1.2
- formato decimal empaquetado
 - campo de entrada 9.1.1
 - campo de salida 9.1.1
 - campo matriz/tabla 9.1.1
 - definición 9.1.1
 - descripción 9.1.1
 - longitud en bytes 9.1.1
 - longitud en dígitos 9.1.1
- formato, utilización del vuelco con 4.7
- formatos de campos de datos 9.0
- formulario de codificación
 - Véase data description specifications
 - Véase especificaciones de cálculo
 - Véase especificaciones de control
 - Véase extension specifications
 - Véase file description specifications
 - Véase input specifications
 - Véase line counter specifications
 - Véase output specifications
- función de copia, generador automático de informes
 - Véase generador automático de informes
- Función para las comunicaciones entre sistemas (ICF) VERSION 8.1
- Funciones del RPG II compatibles con el Sistema/36 y RPG/400 B.0
- funciones del sistema
 - spooling 5.2
- fusión de miembros B.1

G

- GDDM 11.1.3
- generación de programa
 - Véase compilación
- generador automático de informes
 - Véase también salida AUTO
 - /COPY 12.2
 - AUTO 12.3.2.2
 - ayudas de programación 12.5
 - clasificación de especificaciones con /COPY 12.2
 - colocación de sentencia /COPY en programa 12.2
 - compilación de un programa fuente 3.0
 - cuerpo del informe 12.3.2.3
 - definición 12.0
 - ejemplos 12.7 a 12.7.6
 - encabezamientos de página 12.3.2.1
 - encabezamientos y campos de columna, ubicación 12.3.2
 - espaciado y salto 12.3.1
 - especificaciones generadas
 - cálculos 12.4.1
 - salida 12.4.2
 - fases A.5
 - formato 12.3
 - información de servicio A.0
 - líneas de impresión, desbordamiento 12.3.2.4
 - mandato CRTRPTPGM 12.6
 - orden de especificaciones incluidas con /COPY 12.2
 - sentencia /COPY
 - descripción general 12.2
 - formato 12.2
 - modificación de especificaciones copiadas VERSION 12.2.1 12.2.1.2
 - subrutina A\$\$SUM 12.4.1

utilización 12.0
gestión de datos distribuidos (DDM) D.0 BACK_1
Gestor de Representación Gráfica de Datos 11.1.3
gráficos 11.1.3

H
H1-H9
Véase indicadores de parada (H1-H9)

I
ICF archivo de comunicaciones 8.1
identificación del programa
Véase nombre del programa
IGC
Véase variables ideográficas (IGC)
ignorado de formato de registro 5.10.3
impresión de detalles, generador automático de informes
Véase salida AUTO
impresión por grupos
ejemplos 12.1.2
especificaciones 12.1.1
imprimir mensajes 4.1
indicador de retorno (RT)
utilizada para finalizar un programa 11.3.1
indicador de último registro (LR)
utilizada para finalizar un programa 11.3.1
indicador LR (último registro)
Véase indicador de último registro (LR)
indicador RT (retorno)
Véase indicador de retorno (RT)
indicadores
desbordamiento
con archivo PRINTER 5.12
descripción general 5.12
ejemplos 5.12.2 5.12.3
lógica de obtención de desbordamiento 5.12.3
presencia o ausencia de 5.12.2
relación con ciclo del programa 5.12.3
valor de 5.12.3
externo (U1-U8)
nivel de control (L1-L9)
parada (H1-H9)
utilizada para finalizar un programa 11.3.1
retorno (RT)
tabla, XINTAB A.1.2
tecla de función (KA-KN, KP-KY)
con archivo WORKSTN 8.2.2
último registro (LR)
indicadores 01-99
Véase indicadores de relación de campo y de registro de campo
Véase indicators conditioning calculations and output
Véase resulting indicators
indicadores de desbordamiento
con archivo PRINTER 5.12
condicionamiento de salida 5.12.2
descripción general 5.12.2
ejemplos 5.12.2 5.12.3
lógica de obtención de desbordamiento 5.12.3
presencia o ausencia de 5.12.2
relación con ciclo del programa 5.12.3
valor de 5.12.3
indicadores de nivel de control (L1-L9)
indicadores de parada (H1-H9)
utilizada para finalizar un programa 11.3.1
indicadores externos (U1-U8)
indicadores KA-KN KP-KY
Véase indicadores de tecla de función
indicadores L1-L9 (nivel de control)
Véase indicadores de nivel de control (L1-L9)
indicadores OA-OG, OV (desbordamiento)
Véase indicadores de desbordamiento (OA-OG, OV)
INFDS
Véase estructura de datos de información de archivo
información
información de servicio
compilador A.0
función automática de informes A.0
información en línea
inicialización
consideraciones especiales para estructuras de datos 10.2.4
de estructuras de datos 10.2.3
de estructuras de datos, ejemplos 10.2.7
de subcampos de estructuras de datos 10.2.5
visión general 10.2.1
Interfaz común de programación 11.2.2

J

juego de caracteres de doble byte
traslado 11.2.3 11.2.4

L

Lenguaje de Consulta Estructurada (SQL) 1.4.3
liberación de un registro bloqueado 5.6
líneas de detalle
 Véase salida AUTO
líneas de impresión en un generador automático de informes 12.3.2.4
*asterisk.LIST
 valor del parámetro GENOPT A.2.4
lista de parámetros
 Véase también código de operación PARM (identificar parámetros)
 creado por SPECIAL 5.14
 creados por PARM 11.1.2
 identificación 11.1.2
 reglas para especificación
listado de referencias cruzadas
 cómo solicitarlo para un programa fuente en RPG 3.2
 CRTRPTPGM 12.6.2
listado IRP, diseño 3.2 A.4
listado, ejemplo de fuente y referencia cruzada 4.2

M

mandato Borrar 8.2.3
Mandato Crear programa generador automático de informes (CRTRPTPGM) 12.6 12.6.1 12.6.2
mandato Crear programa RPG/400 (CRTRPGPGM) 3.0 A.2
mandato CRTRPTPGM 12.6.2
mandato editar fuente (STRSEU) 2.2
mandato GRTOBJAUT 3.2
mandato Otorgar autorización sobre objeto 3.2
mandato RCLRSC 5.9
Mandato Reclamar recursos (RCLRSC) 5.9
mandato Revocar autorización sobre objeto 3.2
mandato RVKOBJAUT 3.2
mandato supervisar mensaje (MONMSG) 3.1
manejo de errores 10.3.5
 Véase también manejo de la codición de excepción/error
manejo de la codición de excepción/error 4.8
 archivos de comunicaciones C.1
 comunicaciones del dispositivo C.2
 con archivos ICF C.1
 valor por omisión del RPG 4.8
manejo de la condición de excepción/error
 Véase manejo de la codición de excepción/error
matriz
 alternativa
 ejemplos de utilización VERSION F.0
 formato binario 9.1.3
 formato de salida de campo F.0
 formato empaquetado 9.1.1
 impresión de elementos de F.0
 índices fijos, utilización F.0
 inicialización 10.2.4
 matrices en tiempo de preejecución
 tiempo de compilación
 tiempo de ejecución
 Véase matriz en tiempo de ejecución
matriz en tiempo de ejecución
 Véase matriz
matriz o tabla en tiempo de compilación
 Véase también matriz
 inicialización 10.2.4
matriz o tabla en tiempo de preejecución
 inicialización 10.2.4
mensajes
 impresión 4.1 4.1.3
 SAA 4.1.2
 utilización 4.1
 visualización 4.1 4.1.3
mensajes de error
 organización A.1.3
 utilización, visualización e impresión 4.1
mensajes SAA, señalización 4.1.2
métodos de proceso
 consecutivo 7.3.2
 número relativo de registro 7.3.1
 para archivo descrito externamente 7.1.4
 para archivo DISK 7.1.4 7.3
 por clave
 Véase proceso por clave
 secuencial entre límites 7.3.4
 secuencial por clave 7.3.3
 sin claves 7.4
 sólo secuencial 7.3.3 7.4
múltiples dispositivos conectados a programa de aplicación 6.1.3

N

nombre de campo
 tabla
 FLDTBL A.5
 XFDTAB A.1.2
 nombre de formato 8.4.1
 nombre del programa
 parámetro *PGM 3.2
 nombres de campo duplicados en sentencia /COPY modificadora 12.2.1.2
 número de página en archivo PRINTER 5.12.1
 número relativo de registro 7.2.3.2

O

objetos
 comunicación 11.0
 obtención de desbordamiento
 Véase también indicadores de desbordamiento (OA-OG, OV)
 descripción general 5.12.3
 lógica 5.12.3
 opción de control de impresora
 Véase PRTCTL
 opción IGNORE 5.10.3
 Opción RENAME 5.10.2
 operación secuencial 1.6.1
 operaciones de archivo válidas
 archivo SPECIAL 5.14
 archivo WORKSTN
 orden de clasificación
 Véase alteración de orden de clasificación

P

palabra clave PLIST para archivo SPECIAL
 descripción de parámetros 5.14
 palabra de edición
 palabras clave
 DDS 7.0
 para archivo de dispositivo de pantalla
 CLEAR 8.2.3
 HELP 8.2.3
 HOME 8.2.3
 PRINT 8.2.3
 ROLLDOWN 8.2.3
 ROLLUP 8.2.3
 para línea de continuación 7.0
 CLEAR 8.2.3
 HELP 8.2.3
 HOME 8.2.3
 PRINT 8.2.3
 ROLLDOWN 8.2.3
 ROLLUP 8.2.3
 pantallas de solicitud
 mandato CRTRPGPGM 3.2
 parámetro ALWNULL 3.2
 parámetro AUT 3.2
 parámetro CODELIST 3.2 A.2.12
 parámetro CVTOPT 3.2
 parámetro de archivo fuente 3.2
 parámetro de área para SPECIAL PLIST 5.14
 parámetro de autorización 3.2
 parámetro de error para SPECIAL PLIST 5.14
 parámetro de estado de retorno 5.14
 parámetro de estado para SPECIAL PLIST 5.14
 parámetro de miembro fuente 3.2
 parámetro de nivel de release 3.2
 parámetro de nivel de seguridad de generación 3.2
 parámetro de opción para SPECIAL PLIST 5.14
 parámetro de opciones de conversión de tipos 3.2
 parámetro de opciones de generación 3.2
 parámetro de opciones de listado fuente 3.2
 parámetro de perfil de usuario 3.2
 parámetro de release destino 3.2
 parámetro de sangrado de listado fuente 3.2
 parámetro de seguimiento de fases 3.2
 Parámetro de señalización de SAA 3.2
 parámetro de vuelco breve 3.2
 parámetro de vuelco de texto intermedio 3.2
 parámetro del programa 3.2
 parámetro GENLVL 3.2
 parámetro GENOPT 3.2
 *ATR A.2.5
 *DUMP A.2.7
 *LIST A.2.4
 *OPTIMIZE A.2.9
 *PATCH A.2.8
 *XREF A.2.6
 parámetro IGNEDECERR 3.2

parámetro ignorar errores de datos decimales 3.2
 parámetro imprimir archivo 3.2
 parámetro INDENT 3.2
 parámetro ITDUMP A.2.10
 parámetro OPTION 3.2 A.2.1
 *DUMP A.2.3
 *SOURCE A.2.1
 *XREF A.2.2
 parámetro para permitir valores nulos 3.2
 parámetro PGM 3.2
 parámetro PHSTRC 3.2 A.2.13
 parámetro PRTFILE 3.2
 parámetro SNPDUMP 3.2 A.2.11
 parámetro SRCFILE 3.2
 parámetro SRCMBR 3.2
 parámetro TEXT 3.2
 parámetro TGTRLS 3.2
 parámetro USRPRF 3.2
 parámetros de inicialización del programa
 y área de datos (PDA) 11.4.1
 parámetro ITDUMP 3.2
 PDA (áreas de datos PIP)
 descripción 11.4.1
 inicialización 10.2.4
 PGR (Rutinas de Gráficos de Presentación) 11.1.3
 plantilla, programa A.1
 prevención de impresiones sobre agujeros 5.12.3
 proceso al azar por clave VERSION 7.3.5
 proceso consecutivo 7.3.2
 proceso por clave
 archivo de límites de direcciones de registros 7.2.3
 archivo indexado 7.2.1
 ejemplos 7.3.5
 secuencial 7.3
 secuencial entre límites 7.3.4
 vía de acceso 7.1.2
 proceso por número relativo de registro 7.3.1
 proceso secuencial entre límites VERSION 7.3.5
 proceso secuencial por clave VERSION 7.3.3
 proceso sin claves 7.4
 proceso sólo secuencial 7.3.2 7.3.3
 programa
 áreas de datos 11.4
 desactivar 11.1.4
 ejemplos 13.1
 entrada 2.2
 plantilla A.1
 retorno sin una terminación 11.3.3
 terminación anormal 11.3.2
 terminación normal 11.3.1
 programa de utilidad para entrada del fuente (SEU) VERSION 1.4.1 2.2
 programa en lenguaje de control (CL)
 definición G.0
 llamada 11.1
 mandatos utilizados con frecuencia 1.2.1
 mandatos utilizados con RPG/400 1.2.1
 programa encapsulado A.1
 programa fuente
 compilación
 en entorno del Sistema/38 3.3
 utilización del mandato CRTRPGPGM 3.1 3.1.1
 ejecución 4.2
 entrada en el sistema 2.2
 listado A.2.1
 programa generado en RPG/400, generador automático de informes
 cálculos 12.4.1
 ejemplos 12.7
 especificaciones de salida 12.4.2
 fuente de especificaciones 12.2
 impresión por grupos 12.1
 reasignación de formato en encabezamientos de página *AUTO 12.3.2.1
 subrutina (A\$\$SUM) 12.4.1
 programas de ejemplo 13.0
 actualización semanal del archivo de tiempos 13.7
 actualización del archivo maestro 13.7.3
 diseño del informe semanal de transacciones de empleados 13.7.2
 edición de entradas del archivo de tiempos 13.7.1
 informe semanal de transacciones 13.7.3
 archivo de control del área de datos 13.1.3 13.2.2
 archivo de entrada de transacciones mensual 13.2.7
 archivo de entrada de transacciones semanal 13.2.6
 archivo maestro de códigos de razón 13.1.1.3 13.2.5
 archivo maestro de empleados 13.1.1.1 13.2.3
 archivo maestro de proyectos 13.1.1.2 13.2.4

archivo maestro de referencias de la base de datos 13.2.1
 archivos históricos de transacciones 13.1.2
 consulta 8.6.1
 consulta por código postal y búsqueda por nombre 8.6.5
 definición de campo de base de datos 13.2
 diseño de la base de datos 13.1.1
 diseño del menú de información de tiempos 13.3
 entrada de datos 8.6.2
 entrada de transacciones de información de tiempos 13.6.1
 descripciones de datos 13.6.2
 pantalla de selección de empleados 13.6.1.1
 entrada de transacciones del archivo de tiempos 13.6
 línea de comienzo variable 8.6.7
 lista de comprobación 13.1
 mantenimiento de datos 8.6.3
 mantenimiento del archivo de control 13.5.1
 descripciones de datos 13.5.2
 mantenimiento del archivo de control del área de datos 13.1.5
 mantenimiento del archivo maestro 13.1.4 13.4 13.4.1 13.5
 descripciones de datos 13.4.8
 formatos de selección 13.4.1.1
 mantenimiento del maestro de códigos de razón 13.4.7
 mantenimiento del maestro de proyectos 13.4.5
 selección del maestro de códigos de razón 13.4.6
 selección del maestro de empleados 13.4.2
 selección del maestro de proyectos 13.4.4
 mantenimiento del maestro de empleados
 nombre de formato en especificaciones de salida 8.6.6
 operación de lectura con límite de tiempo de espera excedido 8.6.8
 proceso de actualización semanal de archivo de tiempos 13.1.7
 proceso de entrada del archivo de tiempos 13.1.6
 proceso de final de año 13.9
 proceso de información y actualización mensual del archivo de entrada de tiempos 13.1.8
 proceso de subarchivo 8.6.4
 proceso mensual 13.8
 actualización e información mensual del archivo de tiempos 13.8.1
 descripciones de datos del informe de resumen de códigos de razón 13.8.9
 descripciones de datos del informe de resumen de empleados 13.8.3
 descripciones de los datos del informe de resumen del proyecto 13.8.6
 diseño de la información de resumen de la información de tiempos de empleados 13.8.2
 diseño del informe de resumen de información de tiempos de códigos de razón 13.8.8
 diseño del informe de resumen del proyecto de información de tiempos 13.8.5
 informe de resumen de información de tiempos de empleados 13.8.4
 programa RPG/400 de actualización y borrado mensual del archivo maestro 13.8.11
 programa RPG/400 de informe de resumen de códigos de razón 13.8.10
 programa RPG/400 de informe de resumen del proyecto 13.8.7
 programa en RPG/400 de entrada de transacciones de información de tiempos
 PRG03 13.6.3
 programa en RPG/400 de mantenimiento del archivo maestro
 PRG01 13.4.9
 PRG02 13.5.3
 programas de utilidad
 ayuda para el diseño de pantallas (SDA) 1.4.2
 Programa de Utilidad para Entrada del Fuente (SEU) 1.4.1
 programas en CL
 Véase programa en lenguaje de control (CL)
 protección de registros/archivos
 Véase bloqueo de archivos mediante RPG
 PRTCTL (control de impresora)
 ejemplo 5.12.4
 información general 5.12.4
 pruebas
 biblioteca de pruebas 4.3
 campos
 Véase indicadores de campo
 punto de interrupción 4.4
 rastreo 4.5
 *PSSR
 Véase subrutina de condición de excepción/error del programa
 puntos de interrupción VERSION 4.4 4.4.2
R
 rastreo de fase A.2.13
 rastreo, utilización VERSION 4.5
 RECNO
 con proceso por número relativo de registro 7.3.1
 recuperación de mensajes 11.2.4
 recuperación del área de datos
 implícita 9.3.1
 redirección de archivos
 definición 5.1
 descripción general 5.1
 registro
 bloqueo 5.6
 claves válidas 7.1.3

- liberación 5.6
- límites 7.2.3.1
- nombre de tabla, XRCTAB A.1.2
- registro de entrada
 - desagrupación 5.7
- registro de salida
 - agrupación en bloques 5.7
- registros compartidos
 - Véase bloqueo de archivos mediante RPG
- registros de límites 7.1.2
- registros de longitud variable 5.13
- reintento en un tiempo de espera excedido de bloqueo de registro 5.6
- relación AND
- retorno desde un programa llamado VERSION 11.3
- RPG II compatible con S/36 y RPG/400 B.0
- RPGOBJ 3.2
- ruptura de control
- Rutinas de Gráficos de Presentación 11.1.3
- S**
- salida
 - diseño 1.5.1
 - especificaciones 5.10.6
 - generadas 12.4.2
- especificaciones de salida *AUTO
 - Véase también generador automático de informes
 - espaciado y salto 12.3.1
 - especificaciones generadas en RPG/400 12.4
 - formato de generador de informes 12.3
 - impresión por grupos 12.1
 - posiciones finales generadas 12.3.2
- salvar-mientras-activo 4.2.1
- SDA
 - Véase ayuda para el diseño de pantallas (SDA)
- secuencia
 - secuencia, clasificación
 - Véase alteración de orden de clasificación
- secuencia ascendente
- secuencia descendente
- sentencia de estructura de datos
 - descritos externamente 9.2
 - descritos por programa 9.2
 - especificaciones 9.2.2 9.2.2.1
 - reglas 9.3
 - reglas para especificación 9.2.2.1
- sentencias modificadoras (función /COPY)
 - especificaciones de campos de entrada 12.2.1.2
 - especificaciones de descripción de archivos 12.2.1.1
- SEU
 - Véase programa de utilidad para entrada del fuente (SEU)
- signos
 - formato externo 9.1.4
 - formato interno 9.1.4
- sintaxis
 - mandato CRTRPTPGM 12.6.1
- sintexis
 - mandato CRTRPGPGM 3.2
- solapamiento de subcampos 9.4.1
- soporte CPI 11.2.2
- soporte de gráficos 11.1.3
- soporte de valores nulos 10.3.4
- soporte SAA 11.2.2
- spooling 5.2
- spooling de salida 5.2.1
- STRSEU (mandato editar fuente) 2.2
- subarchivo
 - códigos de operación de archivo permitidos con 8.3.5
 - descripción general 8.3.5
 - descripciones 8.3.5
 - ejemplos 8.3.5.1
 - formato de registro 8.3.5
 - formato de registro de control 8.3.5
 - utilizaciones de 8.3.5.1
- subarchivos
- subcampos
 - dentro de una estructura de datos 9.4
 - descripción 9.4
 - especificaciones para 9.4
 - para especificaciones de subcampos de estructura de datos 9.4
 - para estructura de datos de información de archivo 4.7
 - para PRTCTL 5.12.4
 - reglas para 9.4.1
 - reglas para inicialización 10.2.5
- subcampos de estructura de datos
 - formato 9.2.1

subcampos de estructuras de datos
 especificaciones 9.4
 reglas 9.4.1
 solapamiento 9.4.1
 SUBR23R3 (recuperación de mensajes) 11.2.1
 SUBR40R3 (manipulación de variables Caracteres de doble byte) 11.2.3
 SUBR41R3 (manipulación de variables Caracteres de doble byte) 11.2.4
 subrutinas en tiempo de ejecución A.1.4
 subrutina A\$\$SUM (generador automático de informes) 12.4.1
 subrutina de condición de excepción/error del programa
 ejemplo 4.8
 subrutina de inicialización (*INZSR)
 visión general de inicialización 10.2.1
 subrutina de recuperación de mensajes (SUBR23R3) 11.2.1
 subrutinas
 A\$\$SUM (generador automático de informes) 12.4.1
 llamada a subrutinas especiales 11.2
 SUBR23R3 (recuperación de mensajes) 11.2.1
 SUBR40R3 (manipulación de variables DBC) 11.2.3
 SUBR41R3 (manipulación de variables DBC) 11.2.4
 tiempo de ejecución A.1.4

T

T-*AUTO
 Véase salida AUTO

tabla
 Véase matriz
 búsqueda
 Véase operación LOKUP

tabla de nombres de archivos, XFLTAB A.1.1

tablas de resumen
 códigos de operación de archivo permitidos con
 DISK 7.4
 PRINTER 5.12 5.12.4
 secuencial 5.13
 SPECIAL 5.14
 WORKSTN 8.3 8.3.4
 proceso de archivo secuencial 5.13
 proceso de archivo SPECIAL 5.14
 proceso de archivo WORKSTN 8.3.4
 proceso de archivos PRINTER 5.12.4

tecla de mandato Ayuda 8.2.3
 tecla de mandato GIRAR HACIA ABAJO 8.2.3
 tecla de mandato GIRAR HACIA ARRIBA 8.2.3
 tecla de mandato IMPR 8.2.3
 tecla de mandato Inicio 8.2.3
 teclas de atención de mandato (CA) 8.2
 teclas de función
 con archivo WORKSTN 8.2.2
 indicadores 8.2.2
 teclas de función de mandato (CF) 8.2
 teclas de mandatos especiales 8.2.3
 técnicas para una codificación eficaz 1.6
 terminación de un programa 11.3 11.3.1

texto intermedio
 descripción A.1
 listado A.1

tipo de datos gráficos 10.3.3

Tipos de datos SAA
 campos de longitud variable 10.3.1
 fecha, hora e indicación de la hora 10.3.2
 soporte de valores nulos 10.3.4
 tipo de datos gráficos DBCS 10.3.3
 visión general 10.3

trabajos de re arranque 11.4.1

traslado de datos de doble byte y adición de caracteres de control (SUBR41R3) 11.2.4
 traslado de datos de doble byte y supresión de caracteres de control (SUBR40R3) 11.2.3

triple asterisco (***)
 especificaciones generadas 12.0 12.4

U

U1-U8
 Véase indicadores externos (U1-U8)

utilización de indicadores
 Véase indicadores

V

valor *ALL 3.2
 valor *CTLSPEC 3.2
 valor *CURLIB 3.2
 valor *CURRENT 3.2
 valor *CHANGE 3.2
 valor *DATETIME 3.2
 valor *DUMP 3.2
 valor *EXCLUDE 3.2
 valor *FLAG 3.2
 valor *GEN 3.2

valor *GRAPHIC 3.2
 valor *LIBCRTAUT 3.2
 valor *LIBL 3.2
 valor *LSTDBG 3.2
 valor *NODUMP 3.2
 valor *NOFLAG 3.2
 valor *NOGEN 3.2
 valor *NOLSTDBG 3.2
 valor *NONE 3.2
 valor *NOOPTIMIZE 3.2 A.2.9
 valor *NOSECLVL 3.2
 valor *NOSOURCE 3.2
 valor *NOSRCDBG 3.2
 valor *OPTIMIZE 3.2
 valor *OWNER 3.2
 valor *PGM 3.2
 valor *SECLVL 3.2
 valor *SRCDBG 3.2
 valor *SRCMBRTXT 3.2
 valor *USE 3.2
 valor *USER 3.2
 valor *VARCHAR 3.2
 valor ALL 3.2
 valor CTLSPEC 3.2
 valor CURLIB 3.2
 valor CURRENT 3.2
 valor CHANGE 3.2
 valor DATETIME 3.2
 valor de nivel de seguridad 3.2
 valor de nombre de lista-autorización 3.2
 valor DUMP 3.2
 valor EXCLUDE 3.2
 valor FLAG 3.2
 valor GEN 3.2
 valor GRAPHIC 3.2
 valor LIBCRTAUT 3.2
 valor LIBL 3.2
 valor LSTDBG 3.2
 valor NODUMP 3.2
 valor NOFLAG 3.2
 valor NOGEN 3.2
 valor NOLSTDBG 3.2
 valor nombre-fase 3.2
 valor NONE 3.2
 valor NOOPTIMIZE 3.2
 valor NOSECLVL 3.2
 valor NOSOURCE 3.2
 valor NOSRCDBG 3.2
 valor NOXREF 3.2
 valor OPTIMIZE 3.2
 valor OWNER 3.2
 valor PGM 3.2
 valor PRV 3.2
 valor QRPGSRC 3.2
 valor QSYSPRT 3.2
 valor SECLVL 3.2
 valor SOURCE 3.2
 valor SRCDBG 3.2
 valor SRCMBRTXT 3.2
 valor USE 3.2
 valor USER 3.2
 valor VARCHAR 3.2
 valor XREF 3.2
 variables ideográficas (IGC)
 Caracteres de doble byte (DBC) variables 11.2.4
 subrutinas SUBR40R3 11.2.3
 subrutinas SUBR41R3 11.2.4
 VCOMMON A.1.2
 vía de acceso
 ejemplo de 7.2.1.1
 para archivo DISK descrito externamente VERSION 7.1.2
 para archivo indexado 7.2.1
 vía de acceso en secuencia de clave 7.1.2
 vía de acceso en secuencia de llegada 7.1.2
 vía de datos abierta 5.8
 compartimiento 5.8
 visualización de mensajes 4.1
 vuelco con formato 4.7
X
 XFDTAB A.1.2
 XFLTAB A.1.2
 XINTAB A.1.2
 XRCTAB A.1.2
 *XREF

parámetro GENOPT A.2.6
valor de parámetro GENOPT 3.2
valor de parámetro OPTION 3.2

COMENTARIOS Hoja de Comentarios
Application System/400
Guía del usuario RPG/400
Versión 3 Release 1.0

Número de Publicación SC10-9426-00

Por favor, sírvase facilitarnos su opinión sobre esta publicación (utilidad, facilidad de lectura, ...), sugiriendo posibles adiciones y supresiones, y liste los errores y omisiones específicos (indicando número de página). Todos los comentarios y sugerencias pasarán a ser propiedad de IBM, sin incurrir por ello en ninguna obligación para con el remitente.

Sus comentarios nos ayudarán a mejorar las futuras ediciones de esta publicación. Cada una de las observaciones que se reciban será detenidamente revisada por las personas responsables de la redacción, traducción y/o revisión de este material. Sírvase anotar sus comentarios en esta hoja y remitirla a la dirección que figura preimpresa al dorso.

International Business Machines, S.A.
Centro de Traducciones y Publicaciones
Avda. Diagonal, 571
08029 Barcelona, España

Nombre _____
Compañía u Organización _____
Dirección _____

Teléfono _____
