

3
24



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
CAMPUS ACATLAN

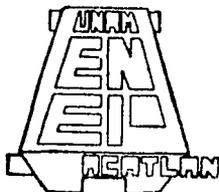


EL USO DE Btrieve EN LENGUAJES DE MICRO
COMPUTADORAS COMO SISTEMA ADMINISTRADOR
DE REGISTROS MULTI-INDEXADOS; BAJO MS DOS,
O REDES CON SISTEMA OPERATIVO NOVELL
NETWARE

T E S I S I N A
para obtener el título de:
LICENCIADO EN MATEMATICAS APLICADAS
Y COMPUTACION

p r e s e n t a
ALEJANDRO DE LA CABADA CELIS

Asesor: ING. REYES GARCIA MONCADA



ACATLAN, ESTADO DE MEXICO

1995

FALLA DE ORIGEN



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL
AVENIDA DE
MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES "ACATLAN"

DIVISION DE MATEMATICAS E INGENIERIA
PROGRAMA DE ACTUARIA Y M.A.C.

ALEJANDRO DE LA CABADA CELIS
Alumno de la carrera de M.A.C.
P r e s e n t e .

Por acuerdo a su solicitud presentada con fecha 19 de mayo de 1994, me complace notificarle que esta Jefatura tuvo a bien asignarle el siguiente tema de Tesina: "EL USO DEL BTRIEVE EN LENGUAJES DE MICRO COMPUTADORAS COMO SISTEMA ADMINISTRADOR DE REGISTROS MULTI-INDEXADOS; BAJO MS-DOS, O REDES CON SISTEMA OPERATIVO - NOVELL NETWARE", el cual se desarrollara como sigue:

INTRODUCCION

CAP. I Introducción a Btrieve

CAP. II Instalación

CAP. III Operaciones

CAP. IV Herramientas para el control de procesos y uso en red.

CAP. V Interfaces con algunos lenguajes comunes y ejemplos.

CAP. VI Otras modalidades y versiones de Btrieve.

CONCLUSIONES.

BIBLIOGRAFIA.

Asimismo, fue designado como Asesor de Tesina el REYES GARCIA MONCADA, Profesor de esta Escuela.

Ruego a usted tomar nota que en cumplimiento de lo especificado en la Ley de Profesiones, deberá presentar servicio social durante un tiempo mínimo de seis meses como requisito básico para sustentar examen profesional, así como de la disposición de la Coordinación de la Administración Escolar en el sentido de que se imprima en lugar visible de los ejemplares de la Tesina el título del trabajo realizado. Esta inscripción deberá imprimirse en el interior de la

A T E N T A M E N T E
"POR MI RAZA HABLARA EL ESPAÑOL"
Acatlán, Edo. Méx. marzo 30 de 1995.

ACT. LAURA MARCELA RIVERA ESCOBAR
Jefe del Programa de Actuaria y M.A.C.

cg'

DEDICATORIAS

A mis padres, porque supieron encaminarme hasta donde me encuentro.

A mi hermana Nuri, por todo su interés hacia mi y por todo el tiempo que dedicó a la revisión de este trabajo.

A mi pareja Ayis, por su apoyo durante la carrera y hasta la fecha.

AGRADECIMIENTOS

Agradezco el tiempo que me otorgaron mi asesor y sinodales para ayudarme a elaborar un mejor trabajo.

A mis compañeros de la carrera ...

Gabriel

Alicia

Luis

Eduardo

Ricardo

Asyadeth

Arturo

Gerardo

Porque con su amistad y ayuda, mis estudios fueron menos difíciles.

Agradezco a mis hermanos por que aunque las condiciones no fueron las mejores siempre nos encontramos unidos de una forma u otra.

Al Ing. Héctor Galindo Miranda por las oportunidades que me brindó, por la confianza que depositó en mí y por su apoyo a lo largo de mis estudios.

ÍNDICE

INTRODUCCIÓN	1
OBJETIVO	6
RESUMEN	7
Capítulo I. INTRODUCCIÓN A BTRIEVE	
1.01 ¿Qué es Btrieve?.....	10
1.02 Clasificación, características, límites e interfaces que proporciona Btrieve versión 5.10a	13
Capítulo II. INSTALACIÓN	
2.01 Generalidades y fundamentos	19
2.02 Diseño de archivos Btrieve	33
2.03 Introducción a los procesos de recuperación, multi-acceso y seguridad de archivos	39
2.04 Instalación y puesta en marcha de la modalidad de <i>basado en cliente</i> de Btrieve para MS-DOS o PS-DOS	41
2.05 Instalación y puesta en marcha de la modalidad de <i>basado en servidor</i> de Btrieve para Novell NetWare.....	51
2.06 Uso del programa BUTIL.EXE o BUTIL.NLM como utilería para crear archivos Btrieve	58
Capítulo III. OPERACIONES	
3.01 Formato y parámetros generales de una operación	69
3.02 Operaciones de Btrieve	73
3.03 Descripción del programa de utilería B.EXE	137
Capítulo IV. HERRAMIENTAS PARA EL CONTROL DE PROCESOS Y USO EN RED	
4.01 Recuperación automática de información y control de procesos por medio de transacciones	139
4.02 Factibilidad del multi-acceso y uso de bloqueos	141
4.03 Descripción del programa de utilería BROLLFWD.EXE	148

Capítulo V. INTERFACES CON DIFERENTES LENGUAJES Y EJEMPLOS

5.01 Descripción del programa ejemplo	151
5.02 Interfaz con Basic Intérprete y el programa ejemplo codificado	159
5.03 Modificaciones al código del programa realizado en Basic Intérprete, para compilarlo en Microsoft QuickBasic versión 2.0 y 4.5	168
5.04 Interfaz con Turbo C versión 2.0 y el programa ejemplo codificado	175
5.05 Interfaz con Turbo Pascal versión 5.0 y el programa ejemplo codificado.....	187

Capítulo VI. OTRAS MODALIDADES

6.01 Posibilidad del uso de Btrieve en lenguajes orientados a objetos	199
6.02 Otras plataformas en las que Btrieve puede ser utilizado	205

CONCLUSIONES

Conclusiones	206
Ventajas y Desventajas	206

RECOMENDACIONES 207

BIBLIOGRAFÍA 209

APÉNDICES

A CÓDIGOS DE ERROR	211
B TIPOS DE LLAVES ADICIONALES	226
C ¿CÓMO FUNCIONAN LAS INTERFACES?	232
D ALGUNAS INTERFACES DE LENGUAJES COMUNES.....	233
E PROGRAMA PARA FACILITAR LA CREACIÓN DE ARCHIVOS BTRIEVE.....	259
F CONTENIDO Y DISTRIBUCIÓN DEL DISCO ANEXO.....	271

GLOSARIO 273

ÍNDICE DE TABLAS

TABLA 3.01. PARÁMETROS DE OPERACIÓN	69
TABLA 3.02. MODOS DE APERTURA	76
TABLA 3.03. VALORES DE BANDERA DE ARCHIVO	100
TABLA 3.03 bis. VALORES DE BANDERA DE ÍNDICE	102
TABLA 3.04. OPERACIONES ORDENADAS POR CÓDIGO	129
TABLA 3.05. OPERACIONES ORDENADAS POR NOMBRE	131
TABLA 3.06. PARÁMETROS REQUERIDOS POR OPERACIÓN	133
CUADRO 6.01. USO DE BTRIEVE EN VISUAL BASIC	200
CUADRO 6.02. MANTENIMIENTO DE ARCHIVOS BTRIEVE EN VISUAL BASIC VERSION 3.0.....	201
TABLA A.01. ERRORES EN BTRIEVE	224
TABLA B.01. CÓDIGOS DE TIPOS DE LLAVES ADICIONALES	226
TABLA B.02. DÍGITO QUE REPRESENTA EL SIGNO EN LA LLAVE TIPO NUMÉRICA	230

INTRODUCCIÓN

Normalmente el primer contacto de un programador con el mundo del diseño de aplicaciones de informática, son los lenguajes de computadora, tales como BASIC, FORTRAN, PASCAL, Lenguaje C, etc.; en estos lenguajes es donde se aprende cuales son las características de una computadora, es decir, todo lo que ofrece y puede realizar, por ejemplo: la manipulación de la pantalla en modo de texto y gráfico, el uso de memoria dinámica, la rapidez de proceso, funciones matemáticas, procesos interactivos, manipulación de puertos de comunicaciones, llamados directos al sistema operativo y juegos; aspectos que se requieren para crear aplicaciones creativas, diferentes y sin barreras, que son razones, por las cuales estos lenguajes no han desaparecido del mercado, por el contrario se actualizan para ofrecer otras posibilidades al programador como interfaces gráficas y mejores ambientes para la codificación.

Pero ¿qué pasa con el vacío que presentan estos lenguajes al no proporcionar herramientas para el manejo de bases de datos?. Realmente es un problema, esto ha provocado que gran cantidad de programadores decidan olvidarse de estos lenguajes, perdiendo parcialmente el control de la computadora al utilizar manejadores de bases de datos que no proveen de las herramientas para el control de la computadora que se encuentran en los lenguajes comunes, además de perder un gran porcentaje de la velocidad de proceso.

Los lenguajes comunes permiten manejar archivos de tipo secuencial, de acceso directo, de registros de longitud fija o variable. Pero esto no es suficiente, el codificar rutinas eficientes para el manejo de bases de datos no es imposible para un programador experto, pero si muy complicado y más si éstas rutinas son diseñadas para uso general o bien para soportar el multi-acceso de una red de computadoras, además del tiempo que le absorbe al programador realizarlas y diseñarlas. Por estos motivos y por considerar que los lenguajes comunes de computadora son muy importantes, se pretende encontrar en el mercado una herramienta que permita el mantenimiento de bases de datos, utilizable en uno o varios de los lenguajes comunes de programación.

Durante el período de investigación de mercado, las empresas que se ofrecieron a platicar sobre el tema; el 90% trabaja sobre Cliper o Foxpro para realizar sus aplicaciones, el 2% sobre manejadores como Oracle y el 8% restante trabajan entre otros sobre lenguaje C, Pascal o Quick Basic en estos casos tienen desarrolladas rutinas para el control de sus bases de datos sobre el lenguaje de su preferencia pero sólo permiten a lo más dos llaves de acceso a sus archivos y no funcionan sobre red. Se logró que una empresa donara para este trabajo algunas de éstas rutinas sobre Basic que pueden ser transportadas a otros lenguajes, se pensó en realizar algunos ajustes para optimizarlas y resolver así este problema, pero se encontró en el mercado un programa que rebasó por mucho las expectativas de la solución inicial. Este programa resuelve el problema descrito anteriormente y se tomó como base para este trabajo, el nombre de dicho programa es "Btrieve", que es un producto de NOVELL, Inc., la empresa que ha desarrollado los mejores sistemas operativos para red, compatibles con el sistema operativo MS-DOS, aplicable también al OS2 y Windows.

Después de seleccionar la solución factible se presenta este trabajo de investigación, con el fin de:

- Exponer en forma clara y accesible cómo podrá un programador hacer uso de esta herramienta.
- Elaborar ejemplos claros y prácticos en algunos lenguajes, que sean el punto de partida para comprender el uso de esta herramienta.

Diseñar y programar, si es necesario, utilerías para facilitar el uso de esta herramienta.

Diseñar y elaborar cuadros y tablas prácticas que sean de ayuda al trabajar esta herramienta.

A continuación se presentan algunas de las principales características de Btrieve, esto es con la intención de enlistar los puntos básicos que permiten afirmar que este programa es ideal para resolver el problema propuesto y percatarse de las ventajas que tiene este programa sobre otras opciones, partiendo de la base de que al trabajar con lenguajes comunes obtenemos una gran flexibilidad y rapidez de ejecución en cualquier tipo de aplicación a desarrollar.

El programa Btrieve es una adición a las instrucciones que presentan diversos lenguajes de programación, y presenta entre otras cosas las siguientes características:

- Permite dar mantenimiento a uno o más archivos, es decir, facilita operaciones tales como insertar registros, actualizarlos, borrarlos y consultarlos.
- Permite definir hasta 24 índices los cuales son colocados físicamente dentro del mismo archivo de datos. Un índice puede ser de diferentes tipos de datos, por ejemplo de un campo numérico (codificado), cadena, fecha, etc. Estos son controlados automáticamente después de cada operación de mantenimiento al archivo y el proceso es totalmente transparente para el programador.
- Permite controlar el multi-acceso a archivos indexados que se encuentran en una red.
- Implementa diversos procedimientos de seguridad de información, tanto en control de procesos, como en contingencias o daños al archivo. También permite asignar claves de acceso a los archivos como medida de seguridad.
- Es factible utilizar tanto registros de longitud fija, como registros de longitud variable, guardando la información en los archivos de manera codificada, comprimida, o encriptada, realizando las conversiones de forma transparente para el programador.

- Es un programa cien por ciento práctico, puesto que, para poder utilizarlo después de desarrollar un programa, tan sólo se requiere de un archivo ejecutable, que debe ser puesto en marcha antes de poder utilizar las funciones de Btrieve, independientemente del lenguaje que se utilizó para crear el programa. En la modalidad de Btrieve que trabaja en el sistema operativo D.O.S, no requiere de ningún proceso de instalación, más que copiar o contar con dicho archivo ejecutable.
- El paquete de Btrieve provee de los archivos y programas necesarios para poder utilizarlo en lenguajes tales como: BASIC, C, PASCAL, COBOL, FORTRAN, MODULA, en sus diferentes marcas y últimas versiones.
- Los requerimientos del hardware y software son mínimos, este programa puede trabajar en una computadora con procesador desde 8086 con 128KB de memoria y sistema operativo ver. 2.1.

Dentro de una red Novell NetWare se encuentra parte del software de Btrieve en la modalidad en red, pero no se incluyen los manuales del programador, es decir que si se cuenta con una red que utiliza Sistema Operativo Novell NetWare, no se puede utilizar Btrieve a menos de que se adquieran los manuales y software adicional, o bien se puede utilizar este trabajo como complemento en vez del manual original ya que contiene la parte adicional del software que se requiere para utilizar Btrieve.

La intención de este trabajo no es presentar la traducción literal de algunos manuales, si no la interpretación clara y accesible de éstos por una parte, y por otra, diseñar y codificar un juego de ejemplos completos que permitirán percatarse de la facilidad y el beneficio del uso de estas herramientas, el código de estos ejemplos se incluye como parte del trabajo.

Estos ejemplos son de lo más simple para que se facilite entender y seguir el flujo del programa y para demostrar que con un programa pequeño y sencillo se pueden lograr aplicaciones muy poderosas en cuanto a bases de datos se refiere.

En este trabajo se recopilan tres manuales de instalación y el manual del programador de Btrieve, de ellos, únicamente dos manuales de instalación se encuentran actualmente traducidos al español. También se utilizó bibliografía de instalación y operación de redes Novell y literatura de tres lenguajes de programación.

¿Que ventajas y desventajas se pueden encontrar en este trabajo a comparación con los manuales originales del programa?.

La factibilidad del aprendizaje del uso de Btrieve a partir de un manual en español es muy alta, puesto que Btrieve no envía ni recibe ningún tipo de información o mensaje en inglés, tampoco existe ayuda en línea por lo tanto no hay confusión entre las palabras traducidas en este trabajo y el programa, sólo en el caso de las utilerías pero esto no es de consideración.

Por otra parte, se pretende que este trabajo sea más didáctico que el manual original, para lo cual se trabajó de la siguiente forma: después haber seleccionado a Btrieve como herramienta, se recopiló en la medida de lo posible toda la información existente acerca del paquete, se instaló una red de computadoras para realizar pruebas y después de comprender en la práctica todas las operaciones y funcionamiento de Btrieve, fue entonces que se comenzó a elaborar este trabajo, sólo de esta manera fue posible interpretar y ampliar la información contenida en el manual, además de modificar un poco la organización y planteamiento de ideas de éste, teniendo como base los problemas iniciales que se presentaron en la práctica, al iniciar con el uso de Btrieve siguiendo paso a paso el manual original.

En el manual del usuario original únicamente se explica cual es la función y el formato de sus operaciones, no comenta de que manera aplicarlas en la práctica y los ejemplos que contiene son sólo fracciones de código para ejemplificar los formatos de las operaciones, estos ejemplos no se presentan de manera completa, además de que se basan en versiones de lenguajes muy antiguos.

Lo que se puede esperar en éste trabajo en adición al contenido de los manuales es:

- Descripciones amplias de como combinar las operaciones de este programa para crear fácilmente aplicaciones que trabajen en red y procesos de mantenimiento de archivos, lecturas ordenadas, catálogos, búsquedas. etc.
- Tablas útiles de consulta rápida.
- Programas de utilería para disminuir algunos de los problemas que se consideran más complicados y que podrían representar un obstáculo para iniciar con el uso de Btrieve.

La organización e interpretación de este trabajo está basado en el punto de vista de una persona con definiciones y términos que pueden ser muy claros y concretos para algunas personas pero no para otras, pensando en eso, se presenta al final de este trabajo un glosario de términos, para dejar más claro el sentido que se le dá a algunas palabras a lo largo de la tesina.

¿Que tan completo es éste trabajo en comparación con el manual original?

Este trabajo contiene todos los puntos que se tratan en el manual original y ¡más!, es posible que de no quedar claro un punto al leer este trabajo, quede la inquietud de consultar el manual original, pero por la experiencia adquirida en la realización de este trabajo se recomienda ejecutar los programas muestra y entender como funcionan para aclarar cualquier duda.

Únicamente existen dos puntos del manual original que no se incluyen: La interfaz con COBOL, y el manejo de archivos extendidos que se refiere a los archivos almacenados en dos dispositivos magnéticos, por lo obsoleto de su propuesta se decidió no incorporar éste último.

Perfil del lector

Esta investigación se realiza con la intención de ayudar a los programadores que requieren utilizar lenguajes comunes de computadora y a la vez deben contar con herramientas para el manejo de bases de datos.

OBJETIVO

Describir el uso de Btrieve orientado a programadores que utilizan lenguajes de medio y alto nivel y así, poner a su disposición las ventajas de contar con una serie de herramientas, que les permitirán administrar sus archivos multi-indexados de una manera rápida, sencilla, eficiente y segura, tanto en computadoras independientes bajo el sistema operativo MS DOS o bien, dentro de una red de computadoras personales con sistema operativo NOVELL NETWARE.

Al hacer uso de estas herramientas el programador podrá reducir sus tiempos de codificación y no desviará su atención del problema real a resolver.

RESUMEN POR CAPÍTULO

Capítulo I. INTRODUCCIÓN A BTRIEVE.

En este capítulo se explica qué es Btrieve, cómo funciona e interactúa con los lenguajes de programación, se resaltarán sus características con el fin de fundamentar su elección dentro de su género, así como sus límites y capacidades, también se definen algunos conceptos básicos de este programa. Por último se enlistan todas las interfaces que provee la versión 5.10a de Btrieve.

Capítulo II. INSTALACIÓN.

En este capítulo se detallan algunos fundamentos y reglas que se requieren al utilizar Btrieve. Se describe cómo se instalan los programas y utilerías de Btrieve, se enlistan los requerimientos de hardware y software mínimos para su uso. Se especifica cuáles son los tipos de archivos y tipos de llave que maneja y cómo se definen los campos índice dentro del registro del archivo, con esto se estará preparado para hacer uso de las operaciones de Btrieve en cualquier interfaz seleccionada. Se describe el uso de una utilería incluida con el programa para facilitar la definición de los campos índice y crear un nuevo archivo Btrieve.

Capítulo III. OPERACIONES.

Se describe cómo opera Btrieve en forma generalizada. Se explican todas las operaciones que provee Btrieve, así como su formato general, su entrada y salida. También se describe el uso de una utilería incluida en el programa que facilita el entendimiento de su funcionamiento.

Capítulo IV. HERRAMIENTAS PARA EL CONTROL DE PROCESOS Y USO EN RED.

Se describe el uso de algunos procedimientos para conservar la seguridad e integridad de la información. También se explica cómo propone Btrieve el uso compartido de información en red sin que haya posibilidad de error o inconsistencia en los archivos que controla la aplicación a desarrollar. Además se explica el uso de una utilería incluida con el programa que sirve para recuperar los archivos dañados que se encuentran en el servidor de archivos de una red.

Capítulo V. INTERFACES CON DIFERENTES LENGUAJES Y EJEMPLOS.

En este capítulo se explica concretamente cómo se usa Btrieve en algunos lenguajes comunes, cuáles son las consideraciones que se deben tomar en cuenta según sea el caso. También se incluye un programa muestra codificado en los lenguajes seleccionados, el cual ejemplifica lo conveniente que es utilizar las operaciones de Btrieve en cada caso.

Capítulo VI. OTRAS MODALIDADES.

En este capítulo se ilustra otro tipo de incursiones de este programa dentro del mercado, como su uso en lenguajes que tienen mayores requerimientos de hardware y software como son los lenguajes visuales u orientados a objetos, así como otras plataformas en las que este programa tiene posibilidad de ser utilizado.

Apéndice A. CÓDIGOS DE ERROR.

Cuando se solicita una operación a Btrieve (descritas en el capítulo III) siempre regresa un status o código de error, que representa el éxito o fracaso de la operación, este código es numérico. El éxito está representado por el número cero y el fracaso por un número entero, en este apéndice se enlistan 76 códigos de error con una descripción de su significado. Al final se presenta una tabla de apoyo de consulta rápida que es esencial en la etapa de programación cuando se comienza a utilizar Btrieve.

Apéndice B. TIPOS DE LLAVE ADICIONALES.

Es común que al ordenar un archivo se tome como llave un campo alfanumérico, Btrieve ofrece la posibilidad de ordenar un archivo por medio de un campo codificado, como por ejemplo: para un número real la longitud de llave es de 4 bytes, para un campo entero la longitud es de 2 bytes; en este apéndice se enlistan las 13 diferentes posibilidades de tipos de llave (además del tipo alfanumérico), se describen completamente los detalles de compatibilidad ya que algunos lenguajes comunes de computadora manejan de forma distinta la representación interna de los números y cadenas.

Apéndice C. ¿CÓMO FUNCIONAN LAS INTERFACES?

Para que un lenguaje de computadora pueda hacer uso de Btrieve debe seguir ciertos pasos, Btrieve provee de archivos que contienen estos pasos codificados, estos archivos son la interfaz entre el lenguaje y Btrieve.

En este apéndice se explica la forma genérica en la que los lenguajes de computadora pueden acceder a Btrieve, es decir que pasos sigue normalmente una interfaz para lograr la comunicación con Btrieve.

Apéndice D. ALGUNAS INTERFACES DE LENGUAJES COMUNES.

Como el objetivo principal de este trabajo es poner a la disposición del programador herramientas para el manejo de bases de datos a partir de lenguajes comunes de computadora, se anexa con el trabajo un disquete que contiene lo necesario para utilizar Btrieve. De no contar con el disco, este apéndice provee del código fuente o programas para generar algunas de las interfaces de los lenguajes más conocidos como:

- Borland Turbo Pascal V. 5.0 (TUR5BTRV.PAS)
- Borland Turbo C V. 2.0 (TURCBTRV.C)
- Basic compilado compatibles con GWBASICA. Ejem. Microsoft Quick Basic V2.0 (BASXBTRV.OBJ)
- Microsoft BASIC V. 7.0 (BC7RBTRV.OBJ)
- Microsoft QuickBASIC V.6.0, 4.5, 4.0 (QB6RBTRV.OBJ)
- IBM BASIC V. 1.0, 2.0 Y 3.0
- Microsoft COBOL (COBxBTRV.OBJ)
- Micro Focus COBOL /2 (COBRBTRV.OBJ)
- Microsoft COBOL V.3.0
- Relia COBOL
- Genérico para Microsoft Fortran (FORX.BTRV.OBJ)
- Genérico para Pascal (PASXBTRV.OBJ)

Apéndice E. PROGRAMA PARA FACILITAR LA CREACIÓN DE ARCHIVOS BTRIEVE.

Para comenzar a utilizar Btrieve es lógico que los movimientos se realicen sobre un archivo. Btrieve opera con archivos creados por el mismo, es decir no trabaja con archivos de otros manejadores de datos ni archivos generados por los lenguajes. Por lo tanto existe necesidad de crear un archivo con el formato de Btrieve antes de intentar manipularlo. De manera inicial un archivo nuevo sólo contiene información general sobre su estructura y sus índices.

Dado que la forma de definir archivos descrita en el capítulo II es compleja, la intención de este apéndice es auxiliar por medio de un programa la creación inicial de un archivo Btrieve.

Apéndice F. CONTENIDO Y DISTRIBUCIÓN DEL DISCO ANEXO.

Al final de cada original de este trabajo se incluye un disco de computadora de 3_1/2" HD que contiene todos los programas que se discuten a lo largo de la tesina, en este apéndice se describe cada archivo que se encuentra en el disco.

CAPÍTULO I

INTRODUCCIÓN A BTRIEVE.

1.01 ¿Qué es Btrieve?

Btrieve es un sistema manejador de registros multi-indexados muy completo que provee 40 operaciones diferentes que en general son útiles para insertar, actualizar, consultar o borrar registros por medio de una o más llaves. Estas operaciones se pueden utilizar desde una gran variedad de lenguajes como BASIC, C, PASCAL, MODULA, COBOL, FORTRAN. Para que Btrieve pueda funcionar con distintos lenguajes, sus rutinas quedan residentes en la memoria de la computadora (ver fig. 1.01) y el paquete contiene interfaces para que los lenguajes puedan acceder las operaciones, estas interfaces formarán parte del código de la aplicación o en otros casos; son archivos objeto que la complementan al ligar el programa en el momento de compilarlo.

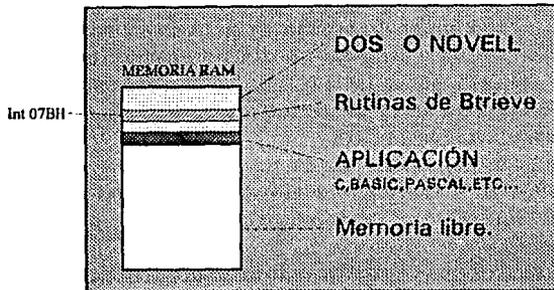


Figura 1.01
Esquema de la memoria

De manera estricta Btrieve es un programa residente (TSR) de tipo pasivo, es decir que se aloja en la memoria como una interrupción del sistema operativo. Cualquier lenguaje común de una forma u otra puede tener acceso a las interrupciones y funciones del BIOS o del DOS, sólo se requiere conocer los parámetros y consideraciones específicas de la función, pero como ya se mencionó anteriormente entre los archivos que conforman el paquete de Btrieve se pueden encontrar una gran variedad de interfaces para los diferentes lenguajes en sus distintas versiones desde lenguaje ensamblador hasta Visual Basic para Windows.

En este estudio se identificarán dos modalidades de Btrieve:

- La modalidad *BASADO EN CLIENTE*, para DOS
- La modalidad *BASADO EN SERVIDOR*, para NOVELL NETWORK.

MODALIDAD BASADO EN CLIENTE, PARA DOS

La modalidad *Basado en cliente*, para DOS, puede funcionar con diversos sistemas operativos incluyendo NetWare, por lo que podemos decir que trabaja en dos ambientes diferentes:

- 1.- En una computadora personal IBM o compatible con sistema operativo MS-DOS ó PS-DOS ambos de la versión 2.0 en adelante, o algún otro DOS compatible. En este ambiente únicamente se tiene la capacidad de controlar archivos locales.
- 2.- Sobre una estación de trabajo de una red de computadoras personales IBM ó compatibles, que trabaje bajo un sistema operativo de red compatible con MS-DOS ó PS-DOS ambos de la versión 3.1 en adelante, o algún DOS completamente compatible. En este ambiente se pueden manejar tanto archivos locales de la estación, como archivos en el servidor, donde permite multi-acceso en los archivos, es decir, que dos o más estaciones de trabajo, accedan a un mismo archivo sin tener conflictos.

Para implementar el uso compartido de archivos del servidor, Btrieve utiliza las llamadas a cerraduras o lock's del sistema operativo, esta propiedad se encuentra a partir de la versión 3.1 del DOS, por lo tanto uno de los requerimientos para funcionar en red con esta modalidad de Btrieve, es que el sistema operativo que se utilice sea compatible completamente con esa versión de DOS.

La característica fundamental en esta modalidad es que todos los procesos para el manejo de archivos son realizados desde la estación de trabajo.

El programa que contiene las rutinas de Btrieve es BTRIEVE.EXE, éste es el único programa que se ejecuta para tener acceso al manejador en esta modalidad de Btrieve.

MODALIDAD BASADO EN SERVIDOR

La modalidad *Basado en servidor* funciona un poco diferente, en este caso el BSERVER (Manejador de Registros Btrieve), es ejecutado en el (o los) servidor(es) de la red, y en las estaciones de trabajo se debe ejecutar el programa BREQUEST.EXE (en el caso de trabajar en una estación de trabajo bajo DOS) en vez del programa BTRIEVE.EXE, de la modalidad *basado en cliente*. En este caso el programa BREQUEST.EXE no contiene las rutinas del manejador, si no que sólo controla los requerimientos entre la estación de trabajo y el BSERVER del servidor correspondiente. El servidor en este caso es el que maneja las entradas y salidas de datos de los archivos.

El BSERVER es implementado en forma distinta en los sistemas operativos Novell NetWare para servidores con procesador 286, que para 386 y 486. En el caso de las versiones de sistemas operativos para 286 se implementa como un proceso de tipo VAP (Value Added Process), y en el caso de versiones de sistemas operativos para 386 y 486 como NLM (NetWare Loadable Modules).

Tanto para la implementación VAP como para la NLM , el concepto es el mismo, el BSERVER queda como parte del sistema operativo de la red, es decir que en cualquiera de los dos casos la función del BSERVER es la misma, la diferencia radica en la forma de instalarlo y ejecutarlo en la red. Posteriormente se especificará cómo hacerlo, en los diferentes tipos de implementación.

El fabricante considera que la forma de trabajo basada en la modalidad *basado en servidor*, es la que optimiza mejor los recursos de la red en cuanto a saturación de información en la línea de comunicación, lo cual es un punto muy importante a considerar, puesto que en redes con un gran volumen de trabajo este aspecto es vital para el buen funcionamiento de la red en general.

Una ventaja notable al desarrollar aplicaciones con base a Btrieve es que una aplicación diseñada para la modalidad *basado en cliente* y una diseñada para la modalidad *basado en servidor* son compatibles entre sí, únicamente se deben considerar algunos aspectos de control, cuando se desea que una aplicación que fue diseñada para trabajar en una computadora independiente, funcione para red en cualquiera de las modalidades posibles.

La modalidad *basada en servidor* es distribuida con la compra del sistema operativo de Novell, no así en el caso de la modalidad *basada en cliente* que se adquiere según el sistema operativo con el cual se desea trabajar, DOS, OS/2 o Windows.

A diferencia de lo que se podría pensar Btrieve no es un programa nuevo, tiene una larga trayectoria casi a la par de la aparición de la microcomputadora. El primer sistema operativo para microcomputadora PC fue desarrollado en agosto de 1981, pero éste sólo trabajaba con manejadores de discos de un solo lado, la siguiente versión permitió trabajar con discos de doble lado de 5 ¼", que fue la 1.1 de mayo de 1982, año en que se produce la primera versión de Btrieve, con los derechos para la compañía SoftCraft. Inc., que desarrolló de la versión 1.0 a la 4.0. A partir de la versión 4.1 del año de 1987, los derechos aparecen para la compañía Novell. Inc., incluyendo la modalidad de *basado en servidor* por primera vez. Existen planes para transferir los derechos a la compañía Btrieve, INC.

1.02 CLASIFICACIÓN, CARACTERÍSTICAS, LÍMITES E INTERFACES QUE PROPORCIONA BTRIEVE Versión 5.10a

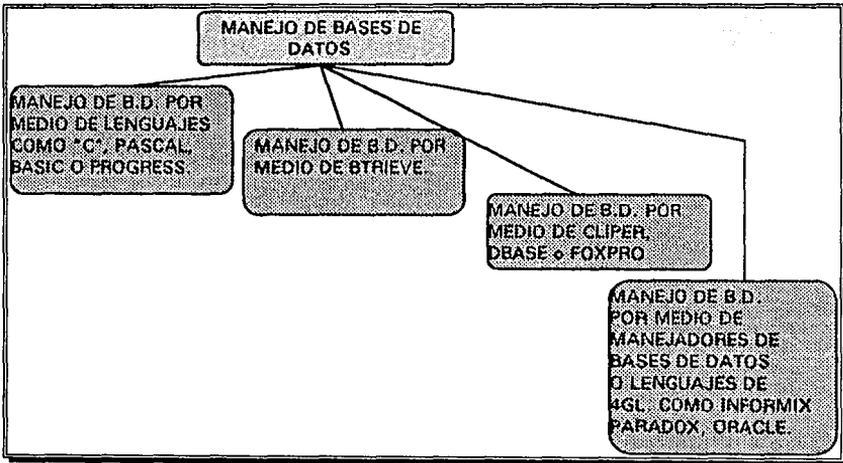
CLASIFICACIÓN DE BTRIEVE.

Como se mencionó antes, Btrieve es un manejador de registros multi-indexados, de una manera estricta ésta sería su clasificación, sin embargo, ha sido la base para productos muy famosos que pertenecen a una clasificación mayor como por ejemplo:

- XQL que sirve para desarrollar aplicaciones que accesan bases de datos relacionales.
- NetWare SQL Servidor de Bases de datos para redes Novel NetWare.

Por lo tanto se puede concluir que Btrieve es un manejador de registros multi-indexados que permite desarrollar aplicaciones que trabajan con el modelo relacional de bases de datos.

El siguiente esquema muestra el lugar que ocupa Btrieve en comparación con las opciones de manejo de bases de datos, los ejemplos que se indican son los que pueden ser representativos para cada nivel. El nivel se considera, en cuanto al grado de elaboración y del producto para el manejo de bases de datos.



En este esquema se ilustra el nivel sobre el cual está dirigido este trabajo, puesto que Btrieve no es un manejador de bases de datos muy elaborado.

CARACTERÍSTICAS Y LÍMITES DE BTRIEVE.

La intención en este subcapítulo es dar un panorama general de las características y límites generales, con el fin de conocer más a fondo el producto antes de describir el procedimiento de instalación.

A continuación se presentan las características y límites más importantes de Btrieve:

- FUNCIÓN BTRIEVE

Btrieve se presenta en cualquiera de los lenguajes como una función, esta función es definida por la interfaz correspondiente al lenguaje. Como toda función, también requiere un cierto número de parámetros para realizar cualquiera de las operaciones que posee, es decir, que todas las operaciones que se pueden manejar con Btrieve, son realizadas por una sola función. De manera general la función tiene la siguiente forma:

$ST\% = BTRV(\text{oper}\%, \text{blok_de_posición}\$, \text{largo_del_buffer_de_datos}\%, \text{buffer_de_datos}\$, \text{número_de_llave}\%, \text{buffer_de_llave}\$)$

donde:

El identificador { $\%$ } representa a una variable entera y { $\$$ } representa una cadena de caracteres (identificadores comunes en BASIC) y por razones obvias los parámetros son enviados por la referencia de la variable y no el valor.

La explicación completa de cada uno de los parámetros de la función será ampliamente descrita en el capítulo III, así como el valor de cada uno de los parámetros para cada operación.

- MANTENIMIENTO DE ÍNDICES

Btrieve da mantenimiento a los índices del archivo, en cualquiera de sus operaciones como insertar, actualizar y borrar registros. Lo importante es que los índices que soporta forman parte del mismo archivo de datos, es decir, aunque se tengan por ejemplo 3 índices, únicamente es necesario tener un archivo abierto, esto es una gran ventaja por dos motivos:

1. Al trabajar de esta forma es más rápido el acceso a la información, porque cuando se realiza la búsqueda de un registro por alguna llave, sólo accesa un archivo, físicamente la información necesaria está más cerca.
2. Se optimiza el número de archivos que se pueden tener abiertos al mismo tiempo y por lo tanto se ahorra memoria.

Las características de los índices son las siguientes:

- Son posibles hasta 24 índices por archivo.
- Es factible adicionar índices suplementarios después de que el archivo fue creado.
- Pueden manejarse 14 diferentes tipos de índices, es decir que un índice puede ser de cualquiera de los 14 tipos de datos definidos en Btrieve. (Ver tabla B.01 del apéndice B)
- Posee diferentes cualidades para cada índice: única, duplicable, modificable o no, segmentada, nula, manual y ascendente o descendente.

- ESPECIFICACIONES EN LOS ARCHIVOS

Para realizar la mayoría de la operaciones de Btrieve, se especifica en uno de los parámetros de la función, el archivo sobre el cual se va a trabajar, este archivo debe, como en todo lenguaje, ser abierto antes de intentar cualquier operación y para abrirlo es necesario que haya sido creado. Se pueden crear archivos de dos formas diferentes, por medio de una operación de Btrieve o mediante el uso de una utilidad incluida con el paquete que facilita este paso. Los archivos manejados por Btrieve tienen las siguientes cualidades:

- Tamaño máximo de archivo hasta más de 4 billones de bytes
- El número de registros está limitado solamente por el tamaño máximo del archivo.
- Los archivos pueden ser manejados en dos dispositivos (discos duros). Útil únicamente en computadoras con procesador 286 o anterior.
- Estructura consistente de archivo.

- MANEJO DE MEMORIA.

Al ser instalado el manejador, en cualquiera de sus dos tipos, el usuario debe definir una cantidad de memoria que será destinada como memoria intermedia (buffer), ésta se utiliza para optimizar el acceso a disco.

Cuando una aplicación solicita consultar un registro, lo primero que Btrieve realiza es verificar si el registro requerido se encuentra en la memoria intermedia, si lo está, Btrieve lo transfiere a la aplicación, de no ser así, el registro es leído directamente del disco y se escribe en la memoria intermedia antes de transferirlo a la aplicación.

Si la memoria intermedia está llena y Btrieve requiere escribir un registro adicional, el algoritmo LRU (least-recently-used), "el usado recientemente", determina qué registro dentro de la memoria intermedia, será sobrescrito. El algoritmo LRU reduce el tiempo de procesamiento guardando en memoria los registros utilizados recientemente .

Cuando la aplicación inserta o actualiza un registro, primero se modifica la memoria intermedia y posteriormente se escribe directamente al disco, éste registro queda dentro de la memoria intermedia hasta que el algoritmo LRU decide sobre-escribirle un registro diferente.

- INTEGRIDAD DE LA INFORMACIÓN

Btrieve usa las siguientes técnicas para permitir el multi-acceso y asegurar la integridad de los archivos:

- Bloqueo simple y múltiple de registros.
- Bloqueo total del archivo.
- Detección de bloqueos muertos. (Necesario únicamente en modalidad *Basado en servidor*)
- Archivos imagen que se utilizan para realizar las operaciones en ellos, antes de tocar el archivo original (opcional).
- Grabación automática (si es solicitada) de cada una de las operaciones que afectan un archivo, con el fin de poder deshacer operaciones o restablecer una falla del servidor o de información corrupta en el archivo.

Cada una de estas técnicas serán explicadas ampliamente en el capítulo IV.

- SEGURIDAD DE LA INFORMACIÓN

En algunas ocasiones es muy importante que la información que maneja un sistema no esté a la vista de cualquier persona. Para esto Btrieve ofrece lo siguiente:

- Asignación de una clave de acceso al archivo.
- Codificación y decodificación dinámica de todo el archivo, si se requiere.

- OPERACIONES DE BTRIEVE

Btrieve provee 40 operaciones que pueden ser llamadas desde una aplicación. (Detalle completo de cada una en el capítulo III)

- UTILERÍAS INCLUIDAS

El disco de programas provee de utilerías para crear, manejar y recuperar daños en archivos de Btrieve.

En el caso de *basado en servidor*, el disco de programas contiene adicionalmente utilerías para monitorear las actividades de la red, utilerías adicionales para recuperar datos perdidos por una falla en el equipo.

- INTERFACES INCLUIDAS

Como se indicó antes, el paquete de Btrieve provee diferentes interfaces. A continuación se presenta la lista de lenguajes de los que es posible contar con su interfaz en la versión 5.10a de Btrieve:

BASIC:

- BASIC interface for the TI
- Borland Turbo Basic
- Digital Research CBASIC (CB-86)
- Generic BASIC
- Microsoft QuickBasic v2.0
- Microsoft QuickBasic v4.0
- Microsoft QuickBasic v4.5
- Microsoft BASIC v6.0;
- Microsoft BASIC v7.0
- IBM BASIC v1.0, v2.0, v3.0
- IBM BASIC/2 for OS/2

LENGUAJE C:

- Ci-C86
- DeSmet C
- Lattice C
- Lattice C v3.0
- Mark Williams C
- Turbo C
- IBM C/2 for OS/2
- IBM C/2 for OS/2(family application)
- WATCOM C

COBOL:

MBP COBOL-85
MBP COBOL
Micro Focus COBOL/2
Micro Focus COBOL
Realia COBOL
RM COBOL
RM/COBOL-8X
Microsoft COBOL v3.0

PASCAL:

Borland Turbo Pascal
NETWARE Pascal
SBB Pascal
Borland Turbo Pascal v4.0
Borland Turbo Pascal v5.0

FORTRAN:

Microsoft Fortran

MODULA:

Logitech Modula-2/86

Nota: Con algunas modificaciones o sin ellas, las interfaces de una versión pueden ser utilizadas para versiones posteriores del mismo lenguaje.

CAPÍTULO II

INSTALACIÓN

2.01 GENERALIDADES Y FUNDAMENTOS

A continuación se presentan diversos conceptos fundamentales que darán la pauta para entender los siguientes capítulos, estos conceptos se refieren en general al mantenimiento de archivos, registros, llaves e índices.

- ARCHIVOS BTRIEVE

Anteriormente se ha hecho el uso del concepto archivo Btrieve; un archivo Btrieve no se refiere a un archivo común, éste tiene algunas características únicas, empezando porque el control de la entrada y salida de éste es totalmente manejada por Btrieve. Este archivo está integrado por la información que se refiere al número de llaves y su ubicación en el registro (cabecera), todos los índices y todos los registros de datos. Por el sólo hecho de que la distribución del archivo es manejada por Btrieve, estos archivos no se accesan por las vías normales de manejo de archivo del lenguaje en el que se realiza la aplicación, es decir, que nos desentendemos totalmente de las instrucciones de apertura y acceso que provee el lenguaje (excluyendo algunas modalidades del lenguaje Basic.)

- COMPONENTES DE UN ARCHIVO

Los archivos Btrieve son separados lógicamente por una serie de *páginas*. Una página es tomada como una unidad de almacenamiento que Btrieve transfiere entre la memoria y el disco. De forma estricta, un archivo Btrieve está compuesto de una página de cabecera, páginas de índices y páginas de datos.

El tamaño de las páginas se definen al crear el archivo Btrieve. El tamaño de la página siempre debe ser un múltiplo de 512 bytes hasta 4096 bytes.

• PÁGINA DE CABECERA

La primer página en cada archivo Btrieve es una página de cabecera que posteriormente será llamada Registro de Control de Archivo RCA. El RCA contiene información acerca del archivo, tal como su tamaño, el tamaño de página, la secuencia de ordenamiento y otras características del archivo.

• PÁGINA DE ÍNDICES

Las páginas de índices contienen los valores de las llaves para los registros de datos. Generalmente, contienen muchas llaves. Cada llave en la página contiene la dirección del registro que corresponde. Btrieve utiliza esta dirección para poder llamar al registro.

- **PÁGINA DE DATOS**

Cuando una aplicación inserta algún registro en un archivo, Btrieve lo coloca en una página de datos. Btrieve ocupa dos tipos de páginas de datos: de registros de longitud fija y variable.

Si al archivo no se le definió que ocuparía registros de longitud variable o la característica de comprimir la información; entonces su página de datos va a contener uno o más registros, pero siempre serán registros completos, no así el caso de los registros de longitud variable que un solo registro podrá estar en una o más páginas. El número de registros en una página de datos depende directamente del tamaño del registro.

Posteriormente se explicará cómo calcular el tamaño óptimo de las páginas.

- **TIPOS DE ARCHIVO**

- Btrieve permite crear tres diferentes tipos de archivo: *normales*, *sólo-datos* y *sólo-llaves*.

- **ARCHIVOS BTRIEVE DE TIPO *NORMALES***

Los archivos Btrieve *normales* contienen una página de cabecera seguida por un número variable de páginas de índices y páginas de datos. Se pueden crear archivos Btrieve *normales* para que contengan registros de longitud fija o variable.

Dado que los archivos *normales* contienen toda la estructura de los índices y datos, Btrieve puede realizar un mantenimiento dinámico tanto de la información de índices como la de los datos. En este tipo de archivo se puede hacer uso de todas las operaciones de consulta que provee Btrieve.

- **ARCHIVOS BTRIEVE DE TIPO *SÓLO-DATOS***

Btrieve también permite crear archivos de *sólo-datos*, éstos únicamente pueden contener información. Cuando se crea un archivo de este tipo, no se debe especificar ninguna llave, ni información de página de índices.

Cuando una aplicación escribe sobre un archivo de este tipo, Btrieve los acomoda en el orden cronológico en que fueron insertados. (El orden cronológico puede variar si se eliminan registros y posteriormente se insertan nuevos registros.) Btrieve no mantiene o crea ninguna página de índices de registros insertados.

Sin embargo, en cualquier momento, es posible adicionar uno o más índices a un archivo *sólo-datos*, y de esta forma se convierte en un archivo normal.

- ARCHIVOS BTRIEVE DE TIPO *SÓLO-LLAVES*

Los archivos de tipo *sólo-llaves* contienen solamente una página de cabecera seguida de una o más páginas de llaves, el registro completo es colocado con la llave, por lo tanto las páginas de datos nos son necesarias. Un uso común de este tipo de archivos, es su uso como un índice externo temporal para los archivos Btrieve de tipo *normales*.

Para utilizar este tipo de archivos se deben considerar las siguientes restricciones:

- ◆ El archivo debe contener una llave única.
- ◆ La longitud máxima del registro que se puede definir es de 255 bytes.
- ◆ Los registros colocados en el archivo no pueden ser actualizados o borrados, solamente pueden ser insertados. Las únicas operaciones de registro válidas que se pueden utilizar en este tipo de archivo son abrir, cerrar, estado, insertar, y las operaciones de consulta.
- ◆ Estos archivos no pueden tener la característica de compresión de datos.

- OPTIMIZACIÓN DE LA ASIGNACIÓN DE ESPACIO.

Btrieve reasigna automáticamente el espacio cuando se inserta o borra un registro de un archivo Btrieve. A continuación se detalla, cómo se realiza esto dinámicamente.

- EXPANSIÓN DINÁMICA

Btrieve asigna el espacio en disco que requiere. Si el archivo no cuenta con lugar disponible dentro de sí mismo, entonces Btrieve dinámicamente asigna páginas de índice o páginas de datos para la nueva inserción. Btrieve actualiza automáticamente la estructura del directorio para reflejar correctamente el nuevo tamaño del archivo. Por esta forma de asignaciones le es posible manejar los índices junto con los datos, ya que se podría pensar que si se requiere espacio para algún índice, Btrieve recorrería los datos para obtenerlo, esto está muy lejos de hacerse.

Una vez que el espacio ha sido asignado al archivo, ese espacio permanece asignado al archivo para siempre. Para eliminar el espacio requerido por los registros borrados en el archivo, se puede crear otro archivo con las mismas características y codificar una pequeña aplicación que lea los registros del archivo original y los inserte dentro del nuevo archivo, después renombrar o borrar el archivo original y substituirlo por el nuevo. Este proceso está contemplado en una utilidad que se detallará posteriormente. Únicamente se debe tener cuidado en realizar esta operación, pues si la aplicación que ocupa este archivo no trabaja estrictamente bajo el modelo de bases de datos relacionales y en cambio requiere de registros físicos para relacionar la información, no se podrá realizar esta operación fácilmente.

- **USO DEL ESPACIO LIBRE**

Cuando se realiza una operación de borrado de algún registro, el espacio que ocupaba anteriormente es asignado a una lista de espacio libre. Cuando se inserta un nuevo registro, Btrieve usa el espacio libre de esta lista si existe, de no ser así se asignan nuevas páginas al archivo. Esta cualidad de retomar estos espacios libres disminuye, de cierta manera, el realizar procesos de reescritura para reclamar espacio en disco.

- REGISTROS

Un *registro* representa una colección de datos lógicamente asociados en un archivo Btrieve. Esta es la unidad de transferencia de información entre el programa de aplicación y Btrieve, en una sola operación.

No hay ninguna restricción inherente en cuanto al número de registros contenidos en un archivo. Un registro puede tener una longitud fija o puede consistir en una porción fija seguida de una porción variable (ver figura 2.01). Los registros de longitud fija que tengan asignada la característica de compresión, son siempre colocados en el archivo como un registro de longitud variable y las llaves definidas en él serán colocadas en la porción fija del registro.

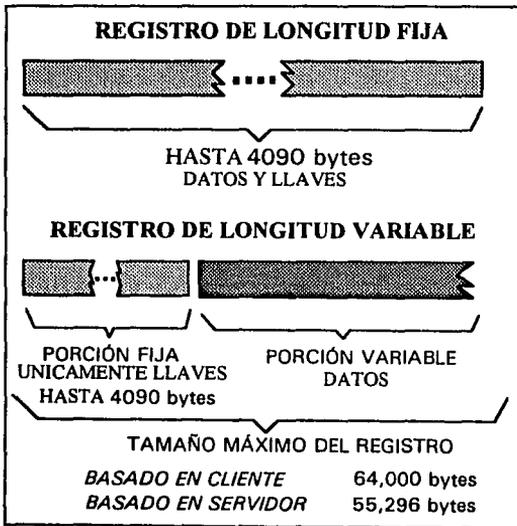


Figura 2.01

- **REGISTROS DE LONGITUD FIJA**

Es posible determinar el tamaño lógico del registro calculando; cuántos bytes de información serán necesarios para guardarlos en él. En el caso de un archivo de longitud fija donde sus llaves no acepten duplicados, la longitud del registro será simplemente su longitud lógica.

El tamaño máximo de un registro de longitud fija depende del tamaño físico de la página y el número de llaves duplicadas definidas para el archivo. Btrieve permite un tamaño máximo de 4,090 bytes para un registro de longitud fija.

- **REGISTROS DE LONGITUD VARIABLE**

Cuando se crea un archivo Btrieve, se puede especificar que el archivo contendrá registros de longitud variable. Esto significa que cada registro del archivo puede tener un tamaño diferente. El tamaño máximo para este tipo de registros depende de el ambiente operativo en el que se trabaja. Btrieve permite registros variables de hasta 55,296 bytes en la modalidad de Btrieve *basado en servidor* y 64,000 en la modalidad *basado en cliente*.

Cuando se crea un archivo que usará registros de longitud variable, se especifica únicamente el largo de la porción fija del registro, que es la que contiene las llaves del registro, esta porción sigue teniendo un límite de 4090 bytes.

Cuando se inserta o actualiza un registro, la aplicación debe usar el parámetro *largo del buffer de datos*, para especificar el largo del registro. Si el largo que se indica es menor que el asignado para la porción fija del registro entonces Btrieve regresará un código de error y no insertará o actualizará el registro.

Cuando se intenta consultar un registro de longitud variable y se especifica un largo de registro menor que el largo de la porción fija, Btrieve regresará un código de error y no regresará ninguna información. Si se especifica un largo de registro igual o mayor al largo de la porción fija pero menor al largo del registro que se pide consultar, entonces, Btrieve regresa el número de bytes que se solicitaron y un código de error indicando que la información regresada no corresponde al registro en su totalidad.

Si se especifica un largo de registro mayor que el registro que se pretende consultar, Btrieve regresa únicamente los bytes que componen al registro. En todos los casos, el parámetro donde se indicó el largo del registro se modifica y en su lugar queda el número de bytes leídos realmente por la operación.

Btrieve guarda los registros de longitud variable en sus propias páginas, separadas de la porción fija del registro. Btrieve deja cierta cantidad de espacio libre en cada página donde los registros de longitud variable están colocados. Esto permite expandir el registro cuando se actualiza y minimizar la fragmentación del registro a lo largo de múltiples páginas. Cuando se crea un archivo, se puede especificar la cantidad de espacio libre que se desea que Btrieve deje en cada página de longitud variable.

- LLAVES

Btrieve usa llaves para permitir un acceso directo y rápido a los registros. Btrieve no tiene forma de conocer la estructura de cada campo en el registro, por lo tanto cada llave debe ser especificada por medio de su posición a partir del inicio del registro y de su longitud en bytes.

Por ejemplo, suponiendo que una llave empieza en el octavo byte del registro y su longitud es de cuatro bytes, cuando se inserte un registro en este archivo, Btrieve va a extraer cuatro bytes, empezando en el octavo byte del registro, este valor será tomado como el valor de llave para el índice del archivo.

Es posible definir una llave que esté conformada por dos o más campos, esta llave se denomina llave segmentada, un campo puede pertenecer entonces a diferentes llaves.

- ATRIBUTOS DE LAS LLAVES

Cuando se crea un archivo, se pueden especificar seis diferentes atributos por cada llave en el archivo. Estos seis atributos son duplicable, modificable, segmentada, descendente, nula y manual.

- **LLAVES DUPLICABLES**

Se puede definir una llave como duplicable para identificar un conjunto de registros que contienen la misma llave. Btrieve acomoda las llaves repetidas en el orden cronológico de su inserción en el archivo. Para el caso de las llaves segmentadas, si un componente de la llave tiene el atributo de duplicable, entonces, los demás componentes de la llave deberán ser duplicables. Si se especifica que una llave no será duplicable, entonces Btrieve no le permitirá a la aplicación adicionar un registro que contenga un valor de llave que ya esté contenido en el índice que le corresponda.

Por ejemplo, en un archivo que contiene registros con la información de clientes, se puede definir el campo de COLONIA como llave duplicable y de esta manera no habrá problemas al insertar clientes que se encuentren en una misma colonia. Sin embargo el campo que corresponde a su NÚMERO o CLAVE, normalmente no se desea un valor duplicado dentro del archivo, puesto que este número es de identificación.

- **LLAVES MODIFICABLES**

Es posible definir una llave como modificable, si es así Btrieve le permite a la aplicación actualizar el valor de llave de un registro existente, esto implica la actualización de los índices que hayan sufrido cambios. Es importante notar que se debe tener cuidado cuando se define una llave como modificable y ésta relaciona al archivo con otro, puesto que Btrieve sólo actualiza los índices de un sólo archivo y si no se toman las medidas pertinentes, la base de datos perderá su consistencia.

- **LLAVES SEGMENTADAS**

Una llave puede consistir en uno o más segmentos de un registro. Un segmento normalmente corresponde a un campo del registro, sin embargo, en general, un segmento puede ser una colección de bytes contiguos en el registro. El largo total de la llave es igual a la suma del largo de los segmentos y puede tener un máximo de 255 bytes solamente. Una llave segmentada puede contener bytes que conforman otras llaves.

El número máximo de llaves segmentadas en Btrieve, está limitado por el número de llaves posibles que es de 24, pero también por el tamaño de página del archivo. Se pueden definir hasta 8 llaves con un segmento cada una en una página de 512 bytes. Para un archivo con una página de 1024 bytes o más, se pueden definir hasta 24 llaves segmentadas.

Un archivo que tiene un tamaño de página de 512 bytes puede contener 1 llave con 8 segmentos, 8 llaves con 1 segmento o cualquier combinación entre sí. Si el tamaño de página es de 1024 bytes puede contener 1 llave con 24 segmentos, 24 llaves con 1 segmento o cualquier combinación entre sí.

En una llave segmentada se pueden definir segmentos de diferentes tipos y también el sentido del orden ascendente o descendente.

- **LLAVES DESCENDENTES**

Btrieve normalmente ordena los valores de las llaves en orden ascendente (de menor a mayor). Sin embargo, es posible especificar a Btrieve el orden contrario, es decir, descendente (de mayor a menor).

- **LLAVES NULAS**

Es posible indicarle a Btrieve que excluya la inserción de ciertas llaves en su índice por medio de la definición de llaves nulas. Cuando se define una llave nula, se especifica el valor que se deberá reconocer como nulo para esa llave. Btrieve trata a una llave como nula únicamente si el valor de cada caracter de la llave es igual al valor definido como nulo.

Se puede usar un valor nulo para la llave cuando la información para ésta no se tiene o cuando no se desea la inclusión de ese registro en el índice para esa llave. Las llaves nulas permiten evitar búsquedas a través de índices con caminos más largos por la existencia de llaves no necesarias y también permiten ahorrar el tiempo de una inserción de una llave no requerida. Cuando el valor para una llave nula se encuentra disponible o cuando se requiere que el registro sea insertado en ese índice, se debe actualizar el registro, reemplazando el valor nulo de la llave con otro valor.

Para las llaves segmentadas, cada byte en cada segmento deberá contener el valor nulo para que Btrieve la excluya del índice. También, si un segmento de la llave está definido con el atributo de llave nula, entonces se deben definir todos los demás segmentos con el mismo atributo. Es posible definir valores nulos distintos para diferentes segmentos en una llave segmentada. Los valores nulos más comunes son el *blanco* con valor ASCII de 32 y el 0.

- **LLAVES MANUALES**

Una llave manual es una modificación de una llave nula y puede ser usada para excluir registros específicos de un índice. Así como en el caso de una llave nula, es necesario indicar un valor para definir si el registro se incluye o no en el índice. En el caso de las llaves segmentadas, si un segmento tiene el atributo de llave manual, los demás segmentos también deben tenerlo.

Las llaves manuales tienen todas las características de una llave nula con excepción de una. Si en una llave manual, cada byte de un solo segmento contiene el valor nulo, Btrieve los excluirá del índice aunque los demás segmentos tengan información.

- TIPOS DE LLAVE

Los tipos de llave en Btrieve caen dentro de dos categorías generales: normal y adicional. Los tipos de llave normales son las binarias y de cadena. Los tipos de llave adicionales comprenden una extensa colección de los tipos de datos más comunes.

- **TIPOS DE LLAVES NORMALES**

Cada segmento de llave de tipo normal es binario o cadena. Btrieve ordena el valor de una llave normal como un entero sin signo. Internamente, Btrieve compara las llaves de cadena comparando byte por byte de izquierda a derecha.

Btrieve ordena las llaves de cadena de acuerdo con el valor ASCII de cada carácter, valuando de izquierda a derecha. Sin embargo es posible especificar una secuencia de valores diferente para cada llave de cadena, con el fin de que no siga el orden de la tabla ASCII.

- **TIPOS DE LLAVE ADICIONALES**

Los tipos de llave adicionales le permiten a Btrieve reconocer e intercalar valores de llaves basados en un formato de almacenamiento interno de hasta 14 tipos de datos, considerados como los más comunes. (Ver apéndice B). Esta capacidad permite una gran flexibilidad al diseñar los archivos Btrieve, además de permitir la escritura de los índices en forma codificada, ahorrando espacio y ganando velocidad de acceso.

- **SECUENCIA ALTERNATIVA DE ORDENAMIENTO**

Es posible especificar una secuencia alternativa de ordenamiento para las llaves de cadena. Una secuencia alternativa de ordenamiento puede ser usada para ordenar llaves de acuerdo a un alfabeto diferente al americano, como el Español, el Suizo o Alemán. En el caso del alfabeto Español es útil para colocar la letra {ñ} en su lugar alfabético, o bien, las vocales acentuadas, o que valore por igual a las letras minúsculas y mayúsculas.

En el momento de crear el archivo de Btrieve se indica si se lleva una secuencia diferente, de ser así, únicamente se especifica el nombre del archivo que contiene el nuevo orden. Este archivo tiene las siguientes características:

- ⇒ El primer byte debe ser siempre AC hex (172 en decimal).
- ⇒ Después ocupando los siguientes ocho bytes, un nombre particular, para que Btrieve identifique la secuencia.
- ⇒ Y seguido del nombre se debe indicar el orden en que se desea cada uno de los 256 caracteres. El primer carácter escrito corresponde al primer valor de la nueva secuencia.

- **ÍNDICES**

Un índice es una estructura que se encuentra en un archivo Btrieve, contiene los valores de llave y los mantiene ordenados. Btrieve mantiene dinámicamente los índices en una estructura conocida como *árbol binario balanceado* (balanced B-tree). Cuando se inserta, actualiza o borra un registro, Btrieve ajusta todos los índices del archivo para reflejar los últimos cambios en los valores de llave contenidos en el registro. Btrieve permite definir dos tipos de índices: permanentes y suplementarios.

- **ESTRUCTURA DE ÁRBOL BINARIO BALANCEADO**

Btrieve guarda todos los índices de los registros de datos en forma de árbol binario standard. Un árbol binario es una estructura de datos que permite búsquedas por medio de diversas ramas, obteniendo un acceso rápido y un uso eficiente de espacio en disco. Una vez que la estructura de un árbol binario ha sido creada, no requiere ningún mantenimiento periódico. Árboles separados son creados para cada llave definida en el archivo.

- **ÍNDICES PERMANENTES**

Cuando se crea un archivo Btrieve, se pueden definir una o más llaves para que Btrieve las utilice para construir índices para cada una. Cualquier llave definida cuando el archivo fue creado es permanente porque no hay manera de borrarla del archivo.

- **ÍNDICES SUPLEMENTARIOS**

Btrieve también permite definir índices internos para un archivo después de que éste ha sido creado. Estos índices son los llamados suplementarios. Una vez creado un índice suplementario, Btrieve le da mantenimiento así como a un índice permanente. Las reglas de posicionamiento son las mismas que para un índice permanente. El número total de índices permanentes y suplementarios no debe exceder de 24 si el tamaño de página es al menos de 1024 u 8 si el largo de página es de 512.

Los índices suplementarios se diferencian de los permanentes de dos formas:

1. A diferencia de los índices permanentes, es factible borrar un índice suplementario cuando éste no es necesario para la aplicación. El espacio que ocupaba el índice después de ser borrado, es utilizado por otras páginas de índice pertenecientes a otra llave.

2. Btrieve ordena de diferente manera la colección de valores de llave duplicadas. En un índice permanente la colección de valores de llave duplicadas son colocadas en el orden que fueron insertadas en el archivo. En un índice suplementario, Btrieve coloca los valores duplicados dentro del índice en orden semejante a la posición física del registro.

También es posible definir una secuencia de valores alternativos para los índices suplementarios. Si ya existe una secuencia alternativa en el archivo, Btrieve utiliza ésta para el índice suplementario. Si no existe una secuencia alternativa, se puede incluir una con la definición del nuevo índice suplementario.

- ACCESO DE REGISTROS

Btrieve permite acceder registros basándose en su dirección física dentro del archivo o por el valor de una llave contenida en él.

- **ACCESO DE REGISTROS POR MEDIO DE SU POSICIÓN FÍSICA**

El acceso de registros por su posición física es el método más rápido porque Btrieve no tiene que verificar ninguna página de índices y porque el registro físico anterior o siguiente se encuentra normalmente en la memoria. A continuación se explica el efecto que se obtiene cuando se accesan registros por su posición física y las operaciones de *escalón*.

- ◆ **POSICIÓN FÍSICA ESTABLECIDA**

Cuando se inserta un registro en un archivo Btrieve, el registro es colocado en el primer espacio libre que se encuentre disponible, a pesar de cualquier valor de llave contenido en el registro. El lugar donde es colocado es su posición física o dirección del registro. El registro permanece en esta posición hasta que es borrado del archivo. La posición física de los registros en el archivo determina el orden físico de los mismos.

- ◆ **OPERACIONES DE ESCALÓN**

Una aplicación puede hacer uso de las operaciones de escalón para accesar registros por medio de su posición física dentro del archivo.

Por ejemplo la operación *Escalón inicial*, lee el registro que está colocado en la primer posición física del archivo. La operación *Escalón Siguiete*, regresa el registro colocado en la siguiente posición física. La operación *Escalón Anterior*, consulta el registro colocado en la posición física anterior. La operación *Escalón final*, lee el registro colocado en la última posición física.

Las operaciones de escalón son muy útiles para consultar y/o recorrer rápidamente el archivo de datos, siempre y cuando no sea necesario hacerlo en un orden específico.

Las operaciones de escalón establecen la posición física, pero eliminan la *posición lógica establecida* (si estuviera anteriormente), que es colocada cuando se consulta por medio de una llave.

- **ACCESO DE REGISTROS POR MEDIO DEL VALOR DE UNA LLAVE**

Accesar registros por medio del valor de una llave, permite obtener registros basándose en los valores de una llave específica. A continuación se explica la *posición física establecida* (el efecto que se obtiene en la posición cuando se accesan registros por un valor de llave), las operaciones de *consulta* (las operaciones de Btrieve para accesar registros por el valor de su llave), y el *block de posición* (el espacio utilizado para colocar la información del posicionamiento lógico ó físico del archivo).

◆ POSICIÓN LÓGICA ESTABLECIDA

Cuando se inserta un registro en algún archivo, Btrieve actualiza cada índice, para cada campo llave en el registro, que no tenga un valor nulo. El lugar donde los valores de llave del registro se acomodan dentro del índice se denomina posición lógica.

Cada llave del archivo determina un orden lógico de los registros. El ordenamiento puede ser de menor valor a mayor o de mayor a menor, dependiendo si la llave fue definida como descendente o ascendente. Un registro con un valor duplicado en cierta llave siempre es insertado en la última posición lógica dentro del mismo grupo de valores de llave. Por lo tanto, los registros duplicados se encuentran en el orden cronológico de su inserción.

Al tomar en cuenta cualquier posición lógica actual basada en cierta llave, se considera que: el siguiente registro que se encuentra en la secuencia lógica se conoce como la *posición lógica siguiente*, y el registro anterior en la secuencia lógica actual como *posición lógica anterior*.

La posición lógica actual, la *siguiente* y la *anterior* conforman juntas, la *posición lógica establecida*.

Las operaciones de *consulta* en Btrieve utilizan la *posición lógica establecida*, cuando accesan registros.

◆ LAS OPERACIONES DE CONSULTA.

Una aplicación puede utilizar las operaciones de *consulta* para obtener registros en base al valor de una llave específica. Las operaciones de *consulta* pueden obtener un registro específico o bien, permiten extraer registros en un orden determinado.

Por ejemplo, la operación *Consulta el Primero* obtiene el primer registro del índice indicado en el parámetro de *número de llave*. De forma similar, la operación *Consulta el Último* obtiene el último registro del índice especificado. Algunas operaciones de consulta, como *Consulta Igual a* o *Consulta Menor a*, regresan un registro basándose en el valor de llave especificado en el parámetro *Buffer de llave*.

Cada operación de *Consulta* establece la posición lógica para un sólo índice. Para que una aplicación pueda establecer la posición lógica para otro índice rápidamente, se pueden aplicar los siguientes pasos:

1. Obtener un registro utilizando cualquiera de las operaciones de *Consulta*
2. Utilizar la operación *Obtener Posición* que regresa un entero de 4 bytes con la posición física o dirección del registro.

3. Utilizar la operación Consultar *directamente* con la posición física obtenida en el paso anterior, e indicar el número de llave que se desea, para establecer la nueva posición lógica.

Además de establecer la posición lógica, todas las operaciones de *Consulta*, definen también, la posición física, por lo tanto, después de utilizar una operación de *Consulta* es factible continuar con la operación *Escalón Siguiente* o *Escalón Anterior*. Sin embargo al utilizar cualquier operación de *Escalón*, la posición lógica se pierde, como ya se ha mencionado anteriormente.

Una aplicación puede restablecer la posición lógica de un índice después de que una operación de *Escalón* la borró, siguiendo este procedimiento:

1. Obtener un registro utilizando cualquiera de las operaciones de *Escalón*.
2. Utilizar la operación *Obtener Posición* que regresa un entero de 4 bytes con la posición física ó dirección del registro.
3. Utilizar la operación Consultar *directamente* con la posición física obtenida en el paso anterior y pasar el número llave que se desea establecer como posición lógica actual.

Después de realizar las operaciones anteriores, la aplicación podrá consultar el registro anterior o posterior, basándose en el valor de la llave indicada en el paso 3.

◆ EL BLOCK DE POSICIÓN

Btrieve utiliza un área de memoria llamada *block de posición* para mantener la información del posicionamiento lógico, usada para accesar registros. Btrieve mantiene el posicionamiento e información asociada con cada archivo abierto. Esta información es colocada en un block de 128 bytes de memoria, que es transferida desde la aplicación.

Algunos de los datos contenidos en el *block de posición* son los siguientes:

- El índice actual que identifica el camino. (El número de llave actual)
- Tres apuntadores que reflejan la posición lógica establecida, basándose en el número de llave actual:
 1. Un apuntador que indica la posición lógica siguiente.
 2. La dirección física del registro actual, que no es necesariamente la del último registro consultado.
 3. Un apuntador que indica la posición lógica anterior.

Muchas de las operaciones de Btrieve modifican el contenido del *block de posición* para reflejar la nueva posición dentro de un archivo. Btrieve utiliza la información del posicionamiento para leer secuencialmente a través del archivo a partir de una llave dada.

La aplicación debe mantener un *block de posición* de 128 bytes para cada archivo Btrieve abierto. La aplicación nunca deberá escribir directamente dentro de la memoria destinada para el *block de posición*, ya que puede causar errores de posición o se puede ocasionar daño en el archivo.

Muchas de las operaciones de Btrieve modifican el contenido del *block de posición*.

2.02 DISEÑO DE ARCHIVOS BTRIEVE.

Este subcapítulo provee algunas fórmulas y guías para diseñar óptimamente un archivo Btrieve. Los siguientes temas serán de ayuda para reducir el tamaño que el archivo Btrieve ocupará dentro del disco. Se explica lo referente a la seguridad de la información. También se definen conceptos como la recuperación de información, multi-acceso, conceptos que se describirán completamente en el capítulo V.

Nota. la siguiente discusión y fórmulas para determinar el tamaño del archivo no son aplicables en archivos que utilizan la compresión de datos.

- CÓMO DETERMINAR EL TAMAÑO DE: REGISTRO, PÁGINA Y ARCHIVO.

Cuando se crea un archivo, es necesario especificar el tamaño lógico de página y el largo lógico del registro. A continuación se describe cómo determinar el largo del registro, tamaño de página y tamaño de archivo. Para archivos que permiten la existencia de registros de longitud variable, el tamaño lógico del registro se refiere a la porción fija del registro.

• LARGO DEL REGISTRO.

Para calcular el largo del registro, se debe realizar un proceso de dos pasos:

1. Determinar el largo lógico del registro, que se obtiene calculando cuantos bytes de información se requieren colocar en la porción fija del registro. Este valor se indica cuando el archivo es creado.
2. Determinar el largo físico del registro, que se obtiene con la suma del largo lógico, más el número de bytes requeridos para la cabecera adicional de Btrieve.

Para archivos que contengan únicamente registros de longitud fija y no tengan definida ninguna llave como duplicable; el largo físico del registro será igual al largo lógico del mismo. Para archivos de registros comprimidos, la cabecera adicional siempre es de 5 bytes.

El número de bytes requeridos para colocar cada registro en el disco, depende de cuantas llaves con la característica de duplicables se hayan asignado en la definición del archivo. Btrieve coloca ocho bytes de información extra por cada llave que permita duplicados.

Para determinar el largo físico del registro, se deben adicionar ocho bytes al largo lógico del registro, por cada llave que permita duplicados; sumar cuatro bytes adicionales, si el archivo permite registros de longitud variable, o seis bytes si el archivo permite la truncación de blancos.

- **TAMAÑO DE PÁGINA.**

Btrieve coloca tantos registros como le sea posible en una página de datos del archivo. Cada página de datos requiere de seis bytes para información de la cabecera. La porción fija de los registros no puede estar a lo largo de dos o más páginas, por lo tanto si el tamaño de la página menos seis, no es un múltiplo exacto del largo físico del registro, siempre existirá un espacio sin uso dentro del archivo. Esta es una de las razones más importantes, por la que se deben realizar los cálculos correctamente, si deseamos optimizar al máximo el espacio en disco. Si se utiliza la cualidad de compresión de datos, no hay manera de conocer de antemano el largo físico que ocupa cada registro en el archivo.

Para optimizar el uso del disco, es mejor calcular tamaños de página que contengan uno o más registros con el mínimo de espacio desperdiciado. El tamaño de página debe ser siempre un múltiplo de 512 bytes, con un límite de 4096 bytes. Las páginas muy grandes normalmente utilizan el disco de manera más eficiente.

Por ejemplo, suponiendo que se necesitan 320 bytes para guardar determinada información y una de las llaves de ese registro está definida con la cualidad de aceptar duplicados. El largo lógico del registro a especificar al crear el archivo será de 320. El largo físico del registro será de 328 bytes por los ocho bytes que se agregan por la llave duplicable.

Si selecciona un tamaño de página de 512 bytes, sólo un registro se podrá colocar dentro de cada página y 178 ($178 = 512 - 6 - 328$) bytes por cada página quedarán sin uso. Sin embargo, si selecciona un tamaño de página de 1024 bytes, tres registros podrán estar dentro de cada página y únicamente 34 ($34 = 1024 - 6 - [328*3]$) bytes por cada página quedarán sin uso.

- **TAMAÑO DE ARCHIVO.**

Es posible estimar el número de páginas que se requieren para contener un cierto número de registros en un archivo y de esta manera calcular el tamaño del registro por medio de las siguientes fórmulas.

La estructura de *Árbol Binario* garantiza al menos un 50% de utilización de las páginas de índice. Por lo tanto, el siguiente número de páginas de índice calculadas es multiplicado por dos, para considerar con esto el tamaño máximo posible.

La mayoría de los archivos requiere menos espacio, pero se deben considerar los números máximos cuando se trata de estimar el tamaño máximo del archivo.

Para calcular el número máximo de páginas de datos se debe utilizar la siguiente fórmula:

$$\text{Pág_de_Datos} = \frac{\text{Núm.Reg.}}{[(\text{Tam.Pág.}-6) / (\text{Larg.Reg.} + (8 * \text{Núm.Ilaves.Dup}))]}$$

Para calcular el número de páginas de índices para cada llave definida se debe utilizar alguna de las siguientes fórmulas dependiendo de cada caso:

Para llaves que permiten duplicados.

$$\text{Pág_de_Índice} = \frac{\text{Núm.de.valores.Únicos.de.llave}}{[(\text{Tam.Pág.}-12)/(\text{Larg.de.llave}+12)]} * 2$$

Para llaves que no permiten duplicados.

$$\text{Pág_de_Índice} = \frac{\text{Núm.de.llaves}}{[(\text{Tam.Pág.}-12)/(\text{Larg.de.llave}+8)]} * 2$$

Después de calcular el número de páginas para cada tipo, el siguiente paso es calcular el total de páginas del archivo; adicionando al número de páginas de datos, el número de páginas de índices y una página de índice más para el Registro de Control de Archivo RCA ; de la siguiente manera:

$$\text{Total_de_Páginas} = 1 + \text{Pág_de_datos} + \sum_{i=1}^n \text{Pág_de_Índice}(i)$$

Donde 'n' es el número de llaves (Núm.de.llaves) definidas para el archivo.

Por último el número máximo de bytes requerido para el archivo será de:

$$\text{Tamaño_de_archivo} = \text{Total_de_Páginas} * \text{Tam.Pág.}$$

- OPTIMIZACIÓN DEL ESPACIO EN DISCO.

Btrieve provee de cuatro procedimientos o cualidades para optimizar el espacio en disco y el rendimiento del sistema de computo en general. Estas cualidades son: pre-colocación del archivo, truncación de blancos, compresión de datos y compactación de índice:

- PRE-COLOCACIÓ DE ARCHIVO.

La rapidez de las operaciones de archivos puede ser sensiblemente incrementada si el archivo de datos ocupa dentro del disco áreas físicamente contiguas. Btrieve permite opcionalmente pre-colocar hasta 65,535 páginas de un archivo cuando éste es creado.

Para pre-colocar espacio de disco contiguo a un archivo, la unidad donde se va a crear el archivo debe tener libres, el número requerido de bytes. Btrieve pre-coloca el número de páginas indicado, no importando si el espacio disponible en la unidad esté contiguo o no, Btrieve realiza los ajustes necesarios.

Si no hay el suficiente espacio en el disco para pre-colocar el número de páginas indicadas, Btrieve regresa el código de error 18 que se describe como *Disco lleno*, y no crea el archivo.

Las fórmulas descritas en la sección anterior son de gran ayuda para calcular el número de páginas requeridas para un número máximo de registros estimados.

Cuando se pre-colocan páginas para un archivo, éste ocupa realmente ese espacio. Ningún otro archivo puede utilizar las áreas destinadas en la pre-colocación de otro archivo, hasta que éste sea borrado o reemplazado.

Cuando se insertan registros, Btrieve utiliza el espacio pre-colocado para datos e índices. Cuando todo el espacio pre-colocado del archivo esté en uso, Btrieve expande el archivo de manera normal en el momento que se inserten nuevos registros.

● TRUNCACIÓN DE BLANCOS.

Cuando un archivo contiene registros de longitud variable, es posible especificar que sea utilizado el método de truncación de blancos para optimizar espacio en disco. Cuando se usa este método Btrieve no coloca ningún blanco que se encuentre en la porción variable del registro, no tiene efecto alguno en la porción fija del registro. Btrieve no remueve los blancos sino que los codifica junto con la información.

Cuando un registro codificado por este método es consultado, Btrieve expande el registro a su largo original. Si se define para un archivo la cualidad de truncación de blancos se deben considerar dos bytes más para el tamaño físico del registro.

Esta cualidad es muy recomendable en los casos en que el valor inicial de los campos sean blancos y es probable que el usuario de la aplicación no deba llenarlos.

● COMPRESIÓN DE DATOS.

Cuando se crea un archivo Btrieve, es posible solicitar que todos los registros que se inserten en el archivo se coloquen en el disco de manera comprimida. La compresión de datos puede resultar en una reducción significativa del espacio requerido para colocar registros que contienen muchos caracteres repetidos.

Se recomienda utilizar la compresión de datos en las siguientes circunstancias:

- ◆ La naturaleza de la información que se insertará en el archivo promueve la compresión de los datos. (muchos caracteres repetidos)

- ♦ La necesidad de una mejor utilización del espacio en disco, vale más que un posible incremento en el tiempo de acceso a la información; requerido para los archivos comprimidos.
- ♦ La aplicación que utiliza el archivo no requiere de la memoria extra utilizada por Btrieve para los buffers de compresión y expansión necesarios.

Cuando se realiza la entrada y/o salida en archivos comprimidos, Btrieve utiliza el buffer de compresión que provee de un block de memoria para los procesos de compresión y expansión. Para contar con la memoria necesaria para comprimir o expandir un registro, Btrieve requiere de un buffer suficiente para colocar dos veces el largo del mayor registro que la aplicación inserte dentro de un archivo comprimido. Este requerimiento puede tener un impacto significativo en la cantidad de memoria libre después de activar Btrieve. Por ejemplo, si el registro más largo que la aplicación inserte o consulte es de 2000 bytes, Btrieve requiere entonces, de 4000 bytes extra para comprimir y expandir ese registro.

Porque el largo final del registro no puede ser determinado hasta que el registro es guardado en el disco, Btrieve siempre crea un archivo comprimido como un archivo de registros de longitud variable.

Los archivos *sólo llave* no pueden tener compresión de datos, puesto que únicamente guardan valores de llave.

• COMPACTACIÓN DE ÍNDICE.

Siempre que una página de índice se llena, automáticamente se crea o se dispone de una nueva página y se colocan los valores de la página llena a lo largo de dos páginas. Btrieve provee de una opción llamada compactación de índice, con la cual es posible evitar de cierta forma la necesidad de disponer de nuevas páginas de índice.

Si se especifica la cualidad de compactación de índice, cuando se activa Btrieve, Btrieve verifica si existe espacio disponible en otras páginas, de ser así, lo ocupa rotando los valores de la página llena en la página que tiene espacio disponible.

El resultado del proceso de compactación, es una mayor densidad en las páginas de índice, un mejor uso del espacio en disco y una respuesta más rápida respuesta en las operaciones de consulta. Si se insertan llaves de manera semi ordenada, el uso de las páginas de índice se incrementa de un 50% a casi 100%, cuando se adicionan llaves de forma aleatoria el uso de las páginas de índice se incrementa de 50% a 66%.

En las operaciones de escritura, la compactación lógica requiere que se examinen más páginas del archivo y normalmente se requiere de más entrada y salida de información al disco. La entrada y salida adicional hace más lentas las actualizaciones al archivo. Aunque los efectos exactos de la compactación de índice varían en diferentes situaciones, en las operaciones de escritura el rendimiento baja de un 5% a un 10%.

Los archivos Btrieve permanecen compatibles si la compactación de índice es activada o no. No es necesario especificar la compactación de índice para acceder archivos que contengan páginas de índice compactadas.

2.03 INTRODUCCIÓN A LOS PROCESOS DE RECUPERACIÓN, MULTI-ACCESO Y SEGURIDAD DE ARCHIVOS.

- RECUPERACIÓN AUTOMÁTICA DE INFORMACIÓN.

En esta sección se enlistan los métodos de recuperación y control de información. Estos métodos son descritos ampliamente en el capítulo IV.

Btrieve provee de tres métodos para implementar la recuperación automática de información.

- **Pre-imagen**
Cada archivo de Btrieve que es abierto tiene relacionado un archivo temporal que se utiliza para realizar en él las operaciones que modifican al archivo Btrieve. De haber alguna contingencia o problema el archivo Btrieve puede quedar dañado, es en estos casos cuando se utiliza el archivo de pre-imagen para reparar las últimas modificaciones al archivo de forma automática al abrir nuevamente el archivo.
- **Pre-imagen directa**
Es el mismo método que el anterior, la diferencia es que el archivo de pre-imagen asociado al archivo de datos no se encuentra en el mismo disco.
- **Control de transacción**
Este método asegura la consistencia de la base de datos. Es muy útil cuando se desea mucha seguridad al realizar procesos de escritura en archivos, cuando es preferible que el proceso no se concluya a que termine parcialmente por algún error, o bien cuando el proceso sea susceptible a ser abortado.

- MÉTODOS Y HERRAMIENTAS PARA EL MULTI-ACCESO.

A continuación se enlistan los métodos y herramientas para el control del multi-acceso de archivos. Estos métodos son descritos ampliamente en el capítulo IV.

El multi-acceso a un archivo se realiza cuando dos o más computadoras tienen abierto un archivo de un disco común.

Btrieve provee algunos métodos para resolver conflictos que pueden ocurrir cuando dos o más aplicaciones desean acceder los mismos registros de un archivo Btrieve. Estos métodos son:

- **Bloqueo por transacción.**
Los registros que son accedidos dentro de una transacción se bloquean automáticamente, es decir, ninguna otra aplicación podrá actualizar uno de estos registros hasta que sean liberados.

- **Concurrencia pasiva.**
Este es el método más simple para el control de problemas y consiste en que Btrieve advierte en el momento de actualizar o borrar un registro, si éste fue utilizado antes por otra computadora.
- **Detección de bloqueos muertos.**
Es la advertencia que libera la posibilidad de la espera indefinida de dos o más computadoras por algún recurso.
- **Bloqueo de registros.**
Es la posibilidad de apartar uno o más registros para procesarlos, en el transcurso del proceso ninguna otra computadora podrá modificar uno de estos registros hasta que sean liberados. Un ejemplo de este control está implementado en el programa ejemplo del capítulo V.

Es posible utilizar cualquiera de estos métodos de manera individual o combinada dentro de una aplicación.

- SEGURIDAD EN LOS ARCHIVOS.

Btrieve provee dos métodos para establecer cierto grado de seguridad en los archivos: Asignación de clave de acceso al archivo y la apertura del archivo en modo exclusivo.

- **ASIGNACIÓN DE CLAVES.**

Btrieve permite restringir el acceso a un archivo mediante una clave de acceso. Una vez asignada la clave a un archivo, Btrieve requiere la clave para poder abrir el archivo. Con esto se previene cualquier acceso no autorizado que intente modificar o consultar la información de ese archivo.

Es posible definir la cualidad de consulta a los archivos con clave, si se desea, pero no se permite escribir en el archivo.

Cuando se asigna una clave a un determinado archivo, se recomienda, definir la característica de encriptación, para que se complemente la seguridad y no se pueda obtener información del archivo por medio de accesos externos.

- **APERTURA EN MODO EXCLUSIVO.**

Si se desea limitar el uso de un archivo a una sola estación de trabajo, es factible especificar en la apertura del archivo que se desea abrir éste en modo exclusivo y así, ninguna otra aplicación podrá abrir el archivo hasta que la aplicación lo cierre.

2.04 INSTALACIÓN Y PUESTA EN MARCHA DE LA MODALIDAD DE BASADO EN CLIENTE DE BTRIEVE PARA MS-DOS O PS-DOS

En este subcapítulo se ilustra cual es el contenido y como se instala el paquete de Btrieve en la modalidad de *Basado en cliente*.

- REQUERIMIENTOS DE EQUIPO

Para utilizar Btrieve en el ambiente *basado en cliente*, se debe contar con los siguientes requerimientos mínimos.

- Computadora IBM personal o compatible.
- Un mínimo de 128KB de memoria.
- Al menos una unidad de discos flexibles.
- Sistema operativo PS-DOS o MS-DOS ambos de la V2.0 o mayor, u otro sistema operativo compatible.

Para acceder archivos comunitarios de un servidor de red, se debe contar con los siguientes requerimientos.

- Sistema operativo PS-DOS o MS-DOS ambos de la V3.1 o mayor, u otro sistema operativo compatible.
- Una area local de la red que utilice o emule completamente las funciones de llamadas extendidas del sistema operativo DOS 3.1.

Al poner en marcha el programa residente (TSR) de Btrieve éste ocupa un mínimo de memoria de 38KB, más la memoria requerida para comunicarse con la aplicación y sus buffers.

- CONSIDERACIONES ESPECIALES

A continuación se definen algunos conceptos necesarios para trabajar con esta modalidad.

• ARCHIVOS LOCALES

Los archivos locales se encuentran en una computadora independiente, es decir, que no contiene o no utiliza archivos comunitarios.

Si una aplicación realiza operaciones de bloqueo (propias de un archivo comunitario) en un archivo local, serán ignoradas por Btrieve como si éstas se hubiesen realizado sin problemas. Esto es con el fin, de que una aplicación diseñada para manejar conflictos propios de archivos comunitarios, pueda también trabajar sin ningún problema en archivos locales.

- **ARCHIVOS COMUNITARIOS**

Es posible poner en marcha a Btrieve en diferentes estaciones de trabajo de una misma red y acceder el mismo archivo al mismo tiempo. Esta modalidad de Btrieve puede abrir archivos que se encuentren en el servidor de una red (archivos comunitarios), siempre y cuando éstos no estén abiertos por Btrieve en su modalidad de *Basado en cliente*.

Btrieve provee los controles para el manejo de conflictos entre los accesos de los usuarios. Cada estación de trabajo en la red ejecuta o pone en marcha su propia copia de Btrieve y realiza llamadas de acceso a los archivos comunitarios en la red. Las múltiples copias de Btrieve se comunican entre sí por medio del acceso a disco y operaciones de bloqueo propias del DOS 3.1.

Los siguientes puntos se deben considerar únicamente si la aplicación de Btrieve accesa archivos comunitarios.

- **OPERACIONES INDIVIDUALES EN LAS CABECERAS DE LOS ARCHIVOS.**

Por la naturaleza de los archivos indexados, al realizar una sola operación de Btrieve, se requiere realizar varios accesos a disco para actualizar los índices. Durante estos accesos, el árbol binario que contiene el índice se puede encontrar temporalmente inconsistente.

Por ejemplo, si ocurriera una falla de corriente en la computadora durante una operación de inserción de un registro a un archivo, es posible que un apuntador tenga colocada una dirección hacia información que no se alcanzó a escribir en el disco. Por esta razón, Btrieve se asegura que una estación termine completamente su operación para que otra estación de trabajo pueda realizar otra operación.

Esta modalidad de Btrieve, utiliza las funciones de bloqueo del sistema operativo que están disponibles a partir de la versión 3.1 de DOS, estas funciones le permiten a Btrieve asegurarse que sólo una copia del manejador realice una operación a un archivo a la vez. Si dos aplicaciones en diferentes estaciones de trabajo intentan ejecutar una operación al mismo archivo al mismo tiempo, Btrieve realiza el proceso de una estación, antes de realizar el de la otra. Esta sincronización es lo que se refiere a *operaciones individuales en las cabeceras de los archivos* y es totalmente transparente para el programa de aplicación.

Mientras una operación individual de Btrieve tenga una corta duración, la sincronización no causa una demora significativa. Sin embargo, las operaciones que accesan archivos comunitarios son normalmente más lentas que las operaciones que accesan archivos locales por las siguientes causas:

- El bajo rendimiento asociado con la sincronización de los bloqueos y liberación de registros.
- Demoras por el acceso múltiple al mismo archivo.
- El tiempo adicional requerido por la red para transferir la información.

Si una aplicación abre un archivo en modo *acelerado* o *exclusivo*, Btrieve abrirá el archivo en el modo exclusivo del DOS. Mientras el archivo se encuentre abierto de ese modo, ninguna otra aplicación lo podrá utilizar.

- **MEMORIA INTERMEDIA.**

Btrieve mantiene una *memoria intermedia* para eliminar en lo posible las entradas y salidas a disco. Además, Btrieve se asegura de que la memoria intermedia de cada estación, corresponda con el estado actual del archivo. Al iniciar cualquier operación, Btrieve lee en disco la información que se encuentra en la cabecera del archivo. La información de la cabecera indica si el archivo fue modificado después de que éste se consultó, si el archivo se modificó, la memoria intermedia de la estación es borrada, puesto que ésta ya no refleja el estado actual del archivo.

Es imposible eliminar totalmente las E/S al disco para cualquier proceso que utilice la memoria intermedia porque Btrieve siempre extrae la información de la cabecera del archivo del disco. Sin embargo, en el caso de que ninguna estación haya modificado el archivo que se está utilizando, Btrieve asume que la memoria intermedia actual refleja el contenido del archivo.

Si se abre un archivo en modo *acelerado*, Btrieve no realiza la consulta a la cabecera en cada operación para asegurarse que la memoria intermedia es válida. Mientras que alguna estación no modifique el modo de apertura, Btrieve asume que la memoria intermedia local es correcta.

- ANTES DE EMPEZAR.

Antes de comenzar con la instalación se recomienda hacer una copia de respaldo de los discos originales. Utilice la copia para cualquier proceso referente a la instalación o reinstalación y guarde sus discos originales en un lugar seguro para evitar que se dañen. Los discos originales vienen con la protección contra escritura para evitar un borrado o reemplazo accidental de su contenido.

- INSTALACIÓN DE BTRIEVE EN DISCO FIJO.

El paquete para esta modalidad de Btrieve contiene:

- ♦ Manual de instalación y operación
- ♦ Manual para el programador
- ♦ Dos discos de 5 1/4 " de baja densidad, el primer disco es el de programas y el segundo es el de interfaces. (Esta presentación varía dependiendo de los requerimientos en la compra)

Se tomará como ejemplo la presentación del paquete con la presentación que contiene discos de 5 1/4".

CONTENIDO DEL DISCO ETIQUETADO

Btrieve for DOS
V. 5.10a
PROGRAMS AND UTILITIES
Disk 1 of 2

ARCHIVOS
BROLLFWD.EXE
BTRIEVE.EXE
B.EXE
UPPER.ALT
README.DOC
BUTIL.EXE

CONTENIDO DEL DISCO ETIQUETADO

Btrieve for DOS
V. 5.10a
LENGUAJE INTERFACES
Disk 2 of 2

MBPXBTRV.OBJ	TURXBTRV.PAS	MF2BTRV.OBJ	ABSXBTRV.EXE
BEXTERN.PAS	BASXBTRV.OBJ	MFXBTRV.BIN	REAXBTRV.OBJ
OS.INC	INTRFACE.DOC	LCXBTRV2.C	MAKEMF2.BAT
LCXBTRV.OBJ	RMXBTRV.EXE	LCXBTRV3.C	WAT2BTRV.C
TURCBTRV.C	CSUPPORT.OBJ	MPLXBTRV.OBJ	MPMXBTRV.OBJ
MPXBTRV.PAS	TURBBTRV.BAS	C2FXBTRV.C	RM8XBTRV.EXE
C2XBTRV.C	COBRBTRV.OBJ	COBPBTRV.OBJ	COXBTRV.ASM
QB6PBTRV.OBJ	QB6RBTRV.OBJ	MBP85BTV.OBJ	SBBXBTRV.OBJ
SBBEXTRN.PAS	SAMPLE.FOR	QBIXBTRV.OBJ	MWCXBTRV.C
MODXBTRV.MOD	MODXBTRV.DEF	MBP85FAC.OBJ	MAKERUN.BAT
FORXBTRV.OBJ	FORXBTRV.ASM	DESXBTRV.C	DEMOBTRV.AWS
COXBTRV.OBJ	CBTESTVA.BAS	CBXBTRV.OBJ	CBISSET.OBJ
CBIGET.OBJ	C86XBTV2.C	C86XBTRV.C	BTIXBTRV.OBJ
BTIXBTRV.EXE	B86XBTRV.OBJ	B86XBTRV.EXE	ASM.OBJ
MSCXBTRV.C	BC7RBTRV.OBJ	TUR5BTRV.TPU	TUR5BTRV.PAS
BC7XBTRV.ASM	TUR4BTRV.PAS		

En esta modalidad no se requiere ninguna configuración especial en el momento de instalar los archivos necesarios, inclusive es factible trabajar Btrieve 100% desde los Disquettes. Pero considerando el estado actual de una computadora común, normalmente toda computadora trabaja con una unidad de disco duro. A continuación se indica cuales son los archivos necesarios para utilizar Btrieve en disco duro.

Los pasos a seguir para la instalación de los archivos necesarios en disco son:

1. Realizar la copia de respaldo de los discos originales.
2. Crear un directorio para Btrieve, el nombre es indistinto.
3. Copiar los siguientes archivos al directorio:
 - ◆ BTRIEVE.EXE (Es el programa TSR que contiene el manejador de registros).
 - ◆ B.EXE y BUTIL.EXE (Utilerías).
 - ◆ UPPER.ALT (Archivo con el formato que define una secuencia alternativa de ordenamiento. Considera semejantes las minúsculas y las mayúsculas.)
 - ◆ EL ARCHIVO QUE CONTIENE LA INTERFAZ CORRESPONDIENTE(*) AL LENGUAJE O LENGUAJES QUE SE UTILIZARÁN.

- PONIENDO EN MARCHA BTRIEVE

Después de haber realizado satisfactoriamente la instalación y asegurarse que los requerimientos mínimos del programa estén cubiertos, entonces es factible poner en marcha Btrieve, para lo cual es necesario especificar las opciones de configuración, que se discuten en esta sección.

Es indispensable poner en marcha a Btrieve antes de que alguna aplicación realice cualquier operación, de lo contrario la aplicación recibirá el código de error 20. (Ver apéndice B)

A continuación se presenta el formato de la instrucción BTRIEVE.EXE, y cuales son las opciones de configuración, para poner en marcha el manejador de registros Btrieve.

◆ FORMATO DEL COMANDO.

Para poner en marcha a Btrieve y que cualquier aplicación pueda acceder al manejador de registros se debe ejecutar el comando con el siguiente formato:

```
BTRIEVE [/M:n] [/P:n] [/T:arch] [/I:drive] [/C] [/B:n] [/F:n] [/L:n] [/E] [/O] [/U:n] [/N:n] [/A]
```

(*) Para poder identificar el archivo que contiene la interfaz correcta se debe revisar cada archivo, normalmente tiene la extensión que utiliza el lenguaje descado, dentro de las primeras líneas del archivo se encuentran comentarios específicos de la interfaz, en éstos se encuentra la indicación de sobre que versión o versiones del lenguaje trabaja. En el caso de BASIC, COBOL FORTRAN y otros, las interfaces para estos lenguajes, son archivos objeto (.OBJ), en estos casos el nombre del archivo es el que nos indica para que lenguaje trabaja.

OPCIÓN DE CONF	DESCRIPCIÓN
[/M: n]	Tamaño de memoria.
[/P: n]	Tamaño de página.
[/T: arch]	Nombre del archivo de transacción.
[/I: drive]	Drive de los archivos de pre- imagen.
[/C]	Compactación de índices.
[/B: n]	Tamaño de buffer de pre-imagen.
[/F: n]	Número de archivos abiertos.
[/L: n]	Bloqueos múltiples.
[/E]	Memoria expandida.
[/O]	Manejo de errores críticos de DOS
[/U: n]	Tamaño del buffer de compresión.
[/N: n]	Núm. máx. de archivos dentro de una transacción.
[/A]	Activa el modo que permite grabar el detalle de los movimientos a los archivos.

◆ OPCIONES DE CONFIGURACIÓN.

TAMAÑO DE MEMORIA (/M).

La opción de tamaño de memoria define el área total de datos con la que contará Btrieve en Kbytes. El área de datos consiste en una porción variable para la memoria intermedia y una porción fija para el resto de las variables propias del manejador. El valor por omisión para esta opción es de 32KB. El rango de valores permitidos va de 14KB hasta 64KB.

Dependiendo de lo que se especifique en la opción de *Tamaño de página*, es el mínimo de tamaño de memoria que se debe definir para esta opción. Por ejemplo, para un *tamaño de página* = 512 bytes, el tamaño mínimo de memoria = 14KB; o un *tamaño de página* = 4,096 bytes, el tamaño mínimo de memoria = 38KB.

Ejemplo; para poner en marcha a Btrieve y definir un tamaño de memoria intermedia y otros buffers, igual a 48KB, se debe especificar como sigue.

/M: 48

TAMAÑO DE PÁGINA (/P).

La opción de tamaño de página, especifica el tamaño máximo de páginas de cualquier archivo de Btrieve que se manejará en esa sesión de trabajo. El tamaño de página que se especifique debe ser un múltiplo de 512 bytes, pero no puede exceder los 4,096 bytes. El valor por omisión depende de la opción de *tamaño de memoria*.

Esta opción, junto con la opción /M, determinan el tamaño de los buffers que Btrieve coloca en la memoria intermedia.

Ejemplo; si el tamaño de página más grande de cualquiera de los archivos Btrieve que se van a acceder es de 1,536 bytes, entonces se debe especificar la opción /P como sigue:

/P: 1536

NOMBRE DEL ARCHIVO DE TRANSACCIÓN (/T).

La opción de nombre del archivo de transacción especifica el lugar donde se va a colocar el archivo de control de transacción. Se debe utilizar ésta opción si la aplicación hace uso de operaciones como *inicio*, *fin*, o *abortar transacción*.

Si se trabaja en ambiente de red, todos los usuarios que pongan en marcha a Btrieve utilizando la opción /T deben indicar el mismo lugar para el archivo de control de transacción, aunque es posible que el Drive lógico para cada estación tenga diferente letra (F:, D:, I: etc.). Sin embargo, todos los nombres de archivo especificados por las diferentes estaciones de trabajo deben coincidir en el mismo lugar físico dentro de la red, mientras todas las copias de Btrieve se ejecutan dentro de la red deben compartir el mismo archivo de control de transacción.

El nombre del archivo debe contar siempre que sea necesario, con la especificación del drive y directorio. Por ejemplo, si se desea que Btrieve utilice el archivo llamado BTRIEVE.TRN en el directorio llamado \DATOS del drive por omisión para el archivo de control de transacción, ponga en marcha a Btrieve con la siguiente opción:

/T: \DATOS\BTRIEVE.TRN

DRIVE DE LOS ARCHIVOS DE PRE-IMAGEN (/I).

La opción de drive de archivos de pre-imagen, direcciona los archivos de pre-imagen a un drive diferente al que ocupan los archivos de Btrieve. Si el drive que se especifica corresponde a un RAM drive, la E/S que se requiere normalmente para un archivo de pre-imagen será eliminada, pero Btrieve no podrá recuperar automáticamente los archivos dañados después de una falla de la computadora. En ambiente de red, si una estación dentro de la red pone en marcha a Btrieve con la opción /I, todas las demás estaciones también deberán hacerlo.

Para indicar que se deben crear todos los archivos de pre-imagen en el drive D:, el formato es el siguiente:

/I: D

El drive D: debe contar con la misma estructura (en cuanto a directorios se refiere) que el drive donde se encuentra el archivo de Btrieve. Por ejemplo, si el archivo Btrieve se encuentra en el directorio F:\SISTEMA\DATOS\, entonces el archivo de pre-imagen va a buscar el directorio D:\SISTEMA\DATOS\.

COMPACTACIÓN DE ÍNDICES (/C).

La opción de compactación de índices ordena a Btrieve realizar la compactación de índices en las operaciones de *actualiza e inserta*.

Para activar la compactación de índices, especifique lo siguiente:

/C

TAMAÑO DE BUFFER DE PRE-IMAGEN (/B).

La opción de tamaño de buffer de pre-imagen especifica el tamaño total de memoria que Btrieve va a utilizar para sus buffers de pre-imagen en Kbytes. El valor por omisión es de 16KB. El rango válido para esta opción es de 1KB hasta 64KB.

Para seleccionar 5KB para el buffer de pre-imagen, indique lo siguiente:

/B: 5

NÚMERO DE ARCHIVOS ABIERTOS (/F).

La opción de número de archivos abiertos especifica el número máximo de archivos de datos de Btrieve que pueden estar abiertos simultáneamente en el sistema. El valor por omisión es de 20. El número máximo es de 255.

Por ejemplo, para indicar que 128 archivos podrán estar abiertos simultáneamente, indique lo siguiente:

/F: 128

BLOQUEOS MÚLTIPLES (/L).

La opción de bloqueos múltiples únicamente se aplica cuando se efectúa multi-acceso en archivos.

Ésta opción determina el número máximo de bloqueos múltiples que se pueden definir simultáneamente en una estación de trabajo.

Estrictamente indica el número de entradas a la tabla de bloqueos que Btrieve construye cuando se pone en marcha.

El valor por omisión de la opción /L es de 20. El valor máximo que se puede indicar para esta opción es de 255. Cada bloqueo incrementa en 6 bytes el programa que queda residente en memoria cuando Btrieve se pone en marcha.

Por ejemplo, para permitir a una estación de trabajo contar con un máximo de 40 bloqueos activos simultáneamente, se debe especificar lo siguiente:

/L: 40

TAMAÑO DEL BUFFER DE COMPRESIÓN (/U).

Btrieve utiliza la opción de tamaño de buffer de compresión para definir y colocar el largo del buffer de compresión en memoria, este espacio se utiliza para la conversión y expansión de datos. Si la aplicación a trabajar no utiliza archivos comprimidos, entonces no se debe incluir esta opción, de lo contrario es indispensable indicarla.

El valor que se especifica con esta opción debe ser igual o mayor que el registro más largo que la aplicación va a leer o escribir dentro de un archivo comprimido. Si la aplicación va a actualizar algún registro, el tamaño del buffer que se especificó en la opción /U debe ser lo suficientemente amplio para poder colocar el registro expandido más grande. El número indicado se considera en unidades de KB. El rango válido para esta opción es de 1KB hasta 64 KB. Btrieve aparta memoria para dos bloques del tamaño indicado.

Por ejemplo, si el registro más grande que se utilizará es de 1,800 bytes de largo, se debe especificar lo siguiente:

/U: 2

En este ejemplo, Btrieve aparta 2 x 2048 bytes de memoria para los buffers de compresión.

NÚMERO MÁXIMO DE ARCHIVOS DENTRO DE UNA TRANSACCIÓN (/N).

Esta opción cambia el número máximo de archivos que pueden estar abiertos y ser accesados dentro de una transacción. El valor por omisión es de 12. Los valores válidos para ésta opción es de 1 a 18. Si ésta opción es activada, también se debe indicar la opción /T. Cuando se especifica la opción /T, automáticamente se activa la opción /N con su valor por omisión que es 12.

Por ejemplo, si se desea cambiar a un valor de 16, debe indicarse lo siguiente.

/N: 16

MODO QUE PERMITE GRABAR EL DETALLE DE LOS MOVIMIENTOS DE LOS ARCHIVOS (/A).

Cuando se activa esta opción, Btrieve registra cada operación que modifique el contenido de algún archivo. Cuando ocurre una falla del sistema de cómputo, se puede utilizar la utilidad BROLLFWR al archivo dañado, para reaplicar todos los cambios que se le hicieron al archivo desde el último respaldo. Al activar la opción /A Btrieve busca en el mismo directorio el archivo BLOG.CFG, este archivo contiene el nombre de los archivos de los cuales se desea que Btrieve lleve el registro de sus cambios, sólo los archivos seleccionados en este archivo serán procesados. El archivo BLOG.CFG es de texto y se puede crear o editar con cualquier programa que opere con archivos no documento.

Para activar esta opción sólo indique lo siguiente:

/A

♦ COMO DESACTIVAR A BTRIEVE.

Una vez puesto en marcha Btrieve, toman su lugar en la memoria de la computadora el programa de manejador de registros y los buffers necesarios. Para desactivar y borrar de la memoria a Btrieve existen dos formas:

1. Utilizando el comando STOP de la utilidad BUTIL.
2. Solicitando desde la aplicación la operación 25 (*ALTO*).

Después de que se desactivó Btrieve ya no es posible acceder a sus operaciones, de intentarlo, se obtendrá el código de error 20 (*BTRIEVE NO INSTALADO*).

2.05 INSTALACIÓN Y PUESTA EN MARCHA DE LA MODALIDAD DE *BASADO EN SERVIDOR* DE BTRIEVE PARA NOVELL NETWARE.

Para instalar Btrieve en su modalidad de servidor, primero se debe tener en cuenta la versión de NetWare con que se cuenta, por que la manera en que se implementa Btrieve en redes con Novell NetWare ver 3.x (incluso ver 4.0) es diferente para las versiones 2.x. Se indicará cuando algún aspecto es propio de alguna versión específica.

En este subcapítulo se ilustra cual es el contenido y como se instala el paquete de Btrieve en la modalidad de *Basado en servidor*.

- REQUERIMIENTOS DE EQUIPO PARA REDES CON NETWARE 2.x Y 3.x.

Para utilizar Btrieve en el ambiente *basado en servidor*, se debe contar con los siguientes requerimientos mínimos.

- Una red de micro computadoras trabajando con el sistema de Novell NetWare ver 2.x en adelante.
- Al menos un servidor de archivos con memoria para poder llamar el archivo BSERVER.VAP en NetWare V.2.x y BTRIEVE.NLM y BSPXCOM.NLM en NetWare V.3.x. normalmente se requieren 2MB de memoria.
- Memoria en cada estación de trabajo para almacenar el Btrieve requester, normalmente 28KB dependiendo de la configuración.

- INSTALACIÓN DE BTRIEVE.

A diferencia de la modalidad de *Basado en cliente*, esta modalidad de Btrieve requiere un proceso de instalación, en el cual se define la configuración de diferentes opciones que en la modalidad de *Basado en cliente* se indicaban al poner en marcha Btrieve.

Como ya se ha indicado, esta modalidad se encuentra junto con los discos originales de la red, es decir que, con la compra del sistema operativo de Novell se adquiere también Btrieve, la diferencia es que únicamente se incluyen los manuales de instalación y no los del programador. En cuanto a los programas se cuenta con todo lo necesario para la instalación y puesta en marcha, pero no se incluye ninguna interfaz para desarrollar aplicaciones.

Es factible comprar aparte el manual del programador e interfaces con el fabricante o bien adquirir la modalidad de *Basado en cliente*, que se distribuye comercialmente.

Los pasos para instalar y utilizar Btrieve son los siguientes:

INSTALACIÓN

1. Instalar los programas dentro del servidor de la red.
2. Configurar las opciones de Btrieve.
3. Instalar los programas necesarios en las estaciones de trabajo.

PUESTA EN MARCHA

4. Poner en marcha el BSERVER que se instaló y configuró en el paso 1 y 2.
5. Poner en marcha el BREQUEST en o las estaciones de trabajo que utilizarán alguna aplicación que funcione con Btrieve.

• INSTALACIÓN DE LOS PROGRAMAS DENTRO DEL SERVIDOR DE LA RED Y ESTACIONES DE TRABAJO.

○ IMPLEMENTACIÓN DE NOVELL NETWARE 2.x:

En el caso de la presentación de Novell NetWare en discos de 5 1/4" de baja densidad, se incluyen 3 discos para los programas de Btrieve:

1. NetWare Btrieve VAPs.
2. NetWare BRUTILS
3. NetWare Btrieve Requesters and Utilites.

Utilice una estación de trabajo para entrar al servidor de archivos que desee, con los privilegios de SUPERVISOR.

Genere un directorio dentro del disco de la red para copiar el contenido de los 3 discos de Btrieve.

Después de realizar la copia ejecute dentro del directorio el programa BSETUP.EXE.

Este programa despliega un menú de opciones:

Change File Server Install Btrieve Remove Btrieve Save Configuration Set Configuration
--

Si se desean cambiar los valores preestablecidos por omisión para las opciones de configuración de Btrieve se debe utilizar la opción de Set Configuration o Cambiar Configuración, si éstos fueron modificados guade los cambios con la opción de Save Configuration o guardar configuración.

Sólo resta solicitar la opción de Install Btrieve o Instalar Btrieve para que este programa se copie correctamente al servidor seleccionado en la primera opción.

Las opciones de configuración que se definen en este programa son: (en el orden en que son solicitadas)

OPCIÓN	VALOR POR OMISIÓN	Memoria req. por unidad (Bytes.)
Número de archivos abiertos	20	115
Número de manejadores	60	114
Número de bloqueos	20	8
Número de transacciones	0	1046
Número de archivos por transacción	12	85
Tamaño del buffer de compresión	0KB	2*Valor
Tamaño máximo de registro	8192B	2*Valor+700
Tamaño de página máximo	4096B	0
Número máximo de sesiones	15	1296
Tiempo de actualización de la consola	3	0
Crear archivos transaccionales	No	0
Activar el modo que permite grabar el detalle de los movimientos a los archivos.	No	0

El significado de estas opciones es el mismo que el discutido en la instalación de Btrieve *Basado en cliente*, únicamente se incorporan las siguientes opciones.

- Número de manejadores

Esta opción permite especificar el número máximo de manejadores de archivo que el sistema permitirá utilizar simultáneamente. Si dos estaciones de trabajo abren un archivo simultáneamente, se utilizan dos manejadores. Para calcular el número máximo de manejadores en una red específica multiplique el número de estaciones de trabajo por el número de archivos comunitarios.

- Número de transacciones

Especifica el número máximo de transacciones que pueden estar activas en un servidor. Si se especifica un cero, ninguna aplicación podrá utilizar operaciones de transacción, de lo contrario Btrieve crea un archivo de control en el directorio de SYS:SYSTEM llamado BTRIEVE.TRN.

- Tamaño máximo de registro

Define el tamaño máximo de registro en bytes que cualquier aplicación utilizando Btrieve, podrá acceder.

- Número máximo de sesiones

Especifica el número máximo de estaciones de trabajo que pueden estar accediendo NetWare Btrieve al mismo tiempo.

- Tiempo de actualización de la consola

Controla el tiempo en que la opción STATUS Y ACTIVE del programa de utilidad B, tardará en actualizar su información.

- Crear archivos transaccionales

Define si Btrieve va a marcar la bandera de *transaccional* (atributo adicional de cualquier archivo en el servidor.) a los archivos que genere.

Después de que se utilizó la opción de instalación, el programa BSETUP copia los archivos necesarios al servidor, sólo se requiere reiniciar el servidor para que automáticamente se ponga en marcha Btrieve.

Para desactivar Btrieve de la red se debe utilizar la utilidad **B DOWN** o bien, el comando de consola **DOWN**.

Esto es con respecto al servidor.

○ INSTALACIÓN DE LOS PROGRAMAS NECESARIOS EN LAS ESTACIONES DE TRABAJO Y PUESTA EN MARCHA DEL BREQUEST.EXE.

Para trabajar bajo ambiente DOS, el archivo que se requiere sólo es uno: BREQUEST.EXE, éste puede ser copiado al disco local de la estación de trabajo o bien, ejecutar este programa a partir del disco del servidor copiándolo al directorio \PUBLIC.

Es necesario poner en marcha el BREQUEST en la estación de trabajo, antes de que ésta intente utilizar Btrieve. De lo contrario la aplicación obtendrá un código de error 20.

Para poner en marcha el BREQUEST utilice el siguiente formato:

```
<Drive> BREQUEST /R: número de drives mapeados
                /D: largo del mensaje de datos
                /S: número de servidores
```

Para indicar por ejemplo, 4 drives mapeados, 2048 bytes de largo de mensaje de datos, y 2 servidores, hay que utilizar la siguiente instrucción:

```
BREQUEST /R: 4 /D: 2048 /S: 2
```

- Número de drives mapeados. /R:

Esta opción denota el número máximo de mapas de drives que la estación podrá acceder. Si se omite esta opción se utilizará el valor por omisión que es de 3. Cada unidad incrementa el uso de la memoria requerida para el programa residente en 20 bytes.

- Largo del mensaje de datos. /D:

Esta opción define el largo máximo del registro que se utilizará en esa sesión de trabajo. El programa de REQUESTER utiliza este valor para calcular el tamaño de los buffers que necesita para transferir información entre la aplicación y BSERVER. Este valor debe ser igual al definido en la configuración del servidor para la opción de *tamaño máximo del registro*. El valor por omisión es de 2024 bytes. El rango máximo para este número es de 48 a 54KB dependiendo de otras opciones. El programa reserva un buffer del doble del valor indicado más 538 bytes.

- Número de servidores. /S:

Esta opción denota el número de servidores activos que puede tener NetWare Btrieve al mismo tiempo. El valor por omisión es de 1. El valor máximo que se puede indicar es de 8.

○ IMPLEMENTACIÓN DE NOVELL NETWARE 3.x:

Esta implementación es similar a la anterior únicamente existen diferencias en cuanto al manejo de servidores remotos y en cuanto las instrucciones que permiten instalar y poner en marcha a Btrieve. Las opciones de configuración anteriormente descritas, también se aplican en este caso.

En la presentación de Novell NetWare en discos de 3 1/2" de alta densidad, se incluye un solo disco para los programas de Btrieve:

1. NetWare Btrieve NLM V5.15.
SERVICES AND UTILITIES

INSTALACIÓN

Utilice una estación de trabajo para entrar al servidor de archivos que desee, con los privilegios de SUPERVISOR.

En este caso debe colocar el disco de Btrieve en la unidad de disco flexible de 3 1/2" y copiar el contenido del directorio \SYSTEMNLM al directorio de SYS:SYSTEM del servidor.

A partir de ese momento ya se puede poner en marcha a Btrieve con los valores de las opciones predefinidas.

Si se requiere modificar los valores por omisión, se debe realizar lo siguiente:

El programa BSETUP permite reconfigurar a Btrieve. Se debe ejecutar BSETUP a partir de la consola del servidor.

Para iniciar el programa escriba : **LOAD BSETUP.**

Este programa despliega un menú de opciones:

Save Configuration Set Configuration

Primero debe utilizar la opción de Set Configuration o Cambiar Configuración, si se desean cambiar los valores preestablecidos por omisión para las opciones de configuración de Btrieve, si estos fueron modificados se debe utilizar la opción de Save Configuration o guardar configuración.

Las opciones de configuración que se definen en este programa son las mismas que se contemplan para Btrieve en NetWare 2.x:.

Para poner en marcha a Btrieve en la red se debe indicar la instrucción **BSTART** en la consola de la red. Si se desea asegurar que siempre que se encienda el servidor, se instale automáticamente Btrieve, se debe editar el archivo **AUTOEXEC.NCF** adicionando el comando **BSTART**.

Para desactivar Btrieve de la red se debe utilizar la instrucción **BSTOP** en la consola de la red.

Esto es con respecto al servidor.

INSTALACIÓN DE LOS PROGRAMAS NECESARIOS EN LAS ESTACIONES DE TRABAJO Y PUESTA EN MARCHA DEL BREQUEST.EXE.

Para trabajar bajo ambiente DOS, el archivo que se requiere sólo es uno: **BREQUEST.EXE**, este puede ser copiado al disco local de la estación de trabajo o bien, ejecutar este programa a partir del disco del servidor copiándolo al directorio **\PUBLIC**.

Es necesario poner en marcha el **BREQUEST** en la estación de trabajo, antes de que ésta intente utilizar Btrieve. De lo contrario la aplicación obtendrá un código de error 20.

Para poner en marcha el BREQUEST utilice el siguiente formato:

**<Drive> BREQUEST /R: número de drives mapeados
/D: largo del mensaje de datos
/S: número de servidores**

Para indicar por ejemplo, 4 drives mapeados, 2048 bytes de largo de mensaje de datos y 2 servidores, hay que utilizar la siguiente instrucción:

BREQUEST /R: 4 /D: 2048 /S: 2

El significado de las opciones del BREQUEST.EXE, es el mismo que el discutido para el NetWare 2.x.

2.06 USO DEL PROGRAMA BUTIL.EXE O BUTIL.NLM COMO UTILERÍA PARA CREAR ARCHIVOS BTRIEVE.

Btrieve para DOS contiene utilerías para crear, recobrar y mantener archivos, que son de ayuda al probar y depurar las aplicaciones que utiliza Btrieve.

Las siguientes utilerías están incluidas en el diskette de programas de Btrieve:

- BUTIL utilería que contiene comandos que permiten crear y manejar archivos de datos Btrieve.
- B (La función Ejecutor Btrieve) utilería interactiva que se puede utilizar para aprender como funciona Btrieve o bien, para probar y depurar la lógica de un programa.
- BROLLFWD- utilería para recobrar operaciones de Btrieve.

INTRODUCCIÓN AL BUTIL.

La utilería BUTIL contiene una tabla completa de comandos para la creación, mantenimiento y recuperación de archivos. La siguiente sección describe como ejecutar BUTIL, y como utilizarlo para crear archivos que puedan ser accedados y manejados por Btrieve.

EJECUTANDO BUTIL

Para correr BUTIL, realice los siguientes pasos:

1. Ponga en marcha el manejador de registros de Btrieve antes de correr BUTIL.
2. Ejecute el comando BUTIL con el siguiente formato:

<drive.> BUTIL -COMANDO [Parámetros][-O<clave de acceso>]

donde:

<drive.>	Es el nombre del dispositivo de disco lógico o de otro dispositivo que contiene los archivos del programa Btrieve. Puede omitir el dispositivo lógico si Btrieve se encuentra en el drive actual y directorio de la tabla de búsqueda.
-COMANDO	Es el nombre del comando (CREATE, COPY, LOAD,etc) que desea utilizar. Se debe utilizar este comando con el guión (-) y debe haber un espacio entre BUTIL y el guión.
[Parámetros]	Representan la información que BUTIL necesita para realizar el <i>comando</i> que se eligió. Los parámetros son descritos más adelante para cada comando.
<clave de acceso>	Debe ser la clave de acceso del archivo de Btrieve o bien un asterisco (*). Si especifica un asterisco como clave, BUTIL tomará el nombre del archivo que fué indicado en los parámetros. BUTIL requiere esta clave para todos los archivos Btrieve que tengan especificada una clave de acceso.

Si se entra el comando BUTIL sin parámetros, BUTIL lista los formatos de todos los comandos.

USO DEL COMANDO -CREATE PARA GENERAR ARCHIVOS BTRIEVE.

Como ya se ha indicado anteriormente, el primer paso para poder utilizar un archivo Btrieve, es crear este archivo. La creación de un archivo Btrieve consiste en definir las características generales del archivo, el número, la posición y propiedades de cada llave, Btrieve no requiere de la información de los demás campos, únicamente la información asociada con los campos destinados a formar llaves. La definición de los campos de datos que no forman llaves se definen en el lenguaje sobre el cual se está codificando la aplicación, Btrieve transfiere los campos de datos pero no los valida. Existen dos formas de crear un archivo Btrieve:

1. Por medio de una aplicación que utilice la operación de Btrieve No. 14 *Crear*. Esta forma puede parecer compleja, puesto que todas las definiciones para el archivo, deben de ser colocadas en una posición fija dentro de una variable de tipo cadena.
2. Por medio del comando -CREATE del la utilería BUTIL.EXE, que facilita la definición de las características a través de un archivo intermedio llamado *archivo de descripción*.

En este caso nos enfocaremos al uso de la utilería BUTIL, para crear un archivo.

El formato de este comando es el siguiente:

BUTIL -CREATE (*N<ombre del nuevo archivo btrieve>*) (<Archivo de descripción>)

El *nombre del nuevo archivo btrieve* es un nombre válido para DOS, por ejemplo CLIENTES.BTR, ALUMNOS.DAT, CONTABIL.ENE, DATOS, etc.

La parte importante y lo único necesario para crear un archivo Btrieve es el archivo de descripción.

ARCHIVOS DE DESCRIPCIÓN DE LA UTILERÍA BUTIL.

Un *archivo de descripción* es un archivo secuencial (o ASCII) que contiene cierta información que BUTIL necesita para realizar sus operaciones, ya sea para el comando -CREATE, -INDEX o -SINDEX.

Estos archivos consisten en una lista de elementos que se refieren a una característica específica del archivo o de una llave del archivo. Un elemento está integrado por una palabra llave, seguida por un signo de igual (=) y un valor. Esta forma de definir un archivo Btrieve es más clara que la primera.

Las siguientes páginas contienen la explicación y el formato de cada elemento del archivo de descripción. Algunos elementos son marcados como opcionales y pueden ser omitidos del archivo si no son requeridos para la descripción del archivo o de alguna llave.

Los elementos deben guardar un orden dentro del archivo:

Primero se deben encontrar los elementos que definen al archivo en general. Después deben estar los que definen a las llaves.

La lista presentada a continuación se encuentra en el orden en que debe aparecer en el archivo, y para cada elemento se describen las siguientes características:

- El nombre del elemento
- La palabra llave y el formato correcto del elemento.
- El rango aceptable para ese elemento.
- Los comandos de BUTIL que requieren de ese elemento en su archivo de descripción.
- Una breve explicación.

Algunos de los elementos que se encuentran a continuación ya fueron ampliamente explicados anteriormente.

Largo del registro

Elemento:record=<mmm>

Rango:4-4090

Comandos: CREATE.

El elemento *largo del registro* define el largo lógico del registro en Bytes. Para registros de longitud variable, este número debe corresponder al tamaño de la porción fija del registro.

Registros de longitud variable.

Elemento:variable=<y/n>

Rango:ninguno

Comandos: CREATE.

Especifique "y" en el valor si desea indicar que el archivo a crear va a contener registros de longitud variable, de lo contrario indique "n".

Truncación de blancos (opcional)

Elemento:truncate=<y/n>

Rango:ninguno

Comandos: CREATE.

Si se especifican registros de longitud variable, es factible solicitar también que se realice la truncación de blancos en la porción variable del registro como medio de compresión de información.

Compresión de datos (opcional)

Elemento:compress=<y/n>

Rango:ninguno

Comandos: CREATE.

Especifique "y" en el valor si desea indicar que se compriman los registros de información que se inserten, de lo contrario indique "n".

Número de llaves

Elemento:key=<nn>

Rango:0-24

Comandos: CREATE.

El elemento *número de llaves* especifica el número de llaves que serán definidas en el archivo. Si se indica el valor de 0 para este elemento, la utilería genera un archivo de tipo *sólo llaves*.

Tamaño de página

Elemento:page=<nnnn>

Rango:512-4096

Comandos: CREATE.

El elemento *Tamaño de página* especifica el tamaño físico de la página para el archivo en múltiplos de 512 bytes.

Precolocación de páginas (opcional)

Elemento:allocation=<nnnn>

Rango:1-65,535

Comandos: CREATE.

El elemento *Tamaño de página* especifica el tamaño físico de la página para el archivo en múltiplos de 512 bytes.

Se reemplaza el archivo existente (opcional)

Elemento:replace=<y/n>

Rango:ninguno

Comandos: CREATE.

Especifique "y" en el valor, si desea que se advierta de la existencia de un archivo con el mismo nombre, de lo contrario indique "n" y si ya existe el archivo no se procede con la creación.

Incluir los datos en el archivo (opcional)**Elemento:**data=<y/n>**Rango:**ninguno**Comandos:** CREATE.

Este elemento determina el tipo de archivo a crear. Especifique "n" en el valor si desea indicar que el archivo será de *sólo llaves* e indique 0 en el elemento de *número de llaves*. De lo contrario indique "n" o no incluya el elemento si se desea crear un *archivo normal*.

Cantidad de espacio de separación**Elemento:**fhreshold=<10 / 20 / 30>**Rango:**ninguno**Comandos:** CREATE.

Este elemento determina la cantidad de espacio que Btrieve reservará entre registros de longitud variable para prever la expansión de éstos. El valor se especifica en porcentaje del tamaño de página de datos.

Este elemento no se debe incluir si no se definieron registros de longitud variable.

NOTA: Los siguientes 12 elementos corresponden a la definición de llaves.
--

Posición de la llave**Elemento:**position=<mmm>**Rango:**1-<largo del registro>**Comandos:** CREATE, INDEX, SINDEK.

Indica la posición inicial de la llave dentro del registro. Las llaves que se definan se pueden traslapar.

Largo de la llave**Elemento:**length=<mm>**Rango:**1-<limite del tipo de la llave>**Comandos:** CREATE, INDEX, SINDEK.

Indica el largo de la llave o del segmento de llave.

Se permiten valores de llave duplicados.**Elemento:**duplicates=<y/n>**Rango:**ninguno**Comandos:** CREATE, INDEX, SINDEK.

Este elemento determina si la llave permitirá valores duplicados en el archivo de ser así coloque "y", de lo contrario indique "n".

Se permite modificar los valores de llave.

Elemento:modificable=<y/n>

Rango:ninguno

Comandos: CREATE, INDEX, SINDEK.

Este elemento determina si la llave permitirá modificar su valor, de ser así coloque "y", de lo contrario indique "n".

Tipo de llave.

Elemento:type=<mn>

Rango:0-15

Comandos: CREATE, INDEX, SINDEK.

Este elemento indica el tipo de datos para la llave. El valor *mn* puede ser substituido por las primeras tres letras del tipo (en inglés). Por ejemplo: para cadena "str", para un entero "int", etc. Ver apéndice B.

Se ordena descendentemente. (opcional)

Elemento:descending=<y/n>

Rango:ninguno

Comandos: CREATE, INDEX, SINDEK.

Este elemento determina si la llave se ordenará descendentemente, de ser así coloque "y", de lo contrario indique "n" o no incluya el elemento.

Secuencia alternativa de ordenamiento.

Elemento:alternate=<y/n>

Rango:ninguno

Comandos: CREATE, INDEX, SINDEK.

Este elemento determina si la llave tomará otra secuencia de ordenamiento definida en un archivo, de ser así coloque "y", de lo contrario indique "n".

Llave manual. (opcional)

Elemento:manual=<y/n>

Rango:ninguno

Comandos: CREATE, INDEX, SINDEK.

Este elemento determina si la llave o el segmento de llave que se está definiendo será manual, de ser así coloque "y", de lo contrario indique "n". Si se define manual un segmento, los demás segmentos deben indicarse como manuales.

Llave nula.

Elemento: null=<y/n>

Rango: ninguno

Comandos: CREATE, INDEX, SINDEIX.

Este elemento determina si la llave o el segmento de llave que se está definiendo será nulo, de ser así coloque "y", de lo contrario indique "n". Si se define nulo un segmento, los demás segmentos deben indicarse como nulos.

Valor nulo de la llave

Elemento: value=<nn>

Rango: 1-FF hex

Comandos: CREATE, INDEX, SINDEIX.

Este elemento especifica el caracter que se considera como nulo, este valor se debe indicar únicamente si se especificó "y" en la *llave nula*. Si la llave no es nula este elemento no se debe especificar.

Llave segmentada

Elemento: segmented=<y/n>

Rango: ninguno

Comandos: CREATE, INDEX, SINDEIX.

Este elemento especifica cuando la llave que se está definiendo tiene más segmentos. Especifique "y" si la llave tiene más segmentos, especifique "n" si se está definiendo una llave no segmentada o si está definiendo el último segmento.

Nombre del archivo de la secuencia alternativa de ordenamiento

Elemento: name=<filespec>

Rango: nombre válido para DOS

Comandos: CREATE, INDEX, SINDEIX.

Este elemento indica el nombre del archivo que contiene el formato especial de la secuencia de ordenamiento para el índice que se está definiendo.

Si se especifica "n" para el elemento de *secuencia alternativa de ordenamiento* no se debe incluir este elemento. Sin embargo, si se especifica "s" para aquel elemento en más de un segmento de la llave, se puede definir sólo un archivo que será aplicado a los demás segmentos.

REGLAS PARA LOS ARCHIVOS DE DESCRIPCIÓN.

Las siguientes reglas se aplican para los archivos de descripción que utilizará la utilería BUTIL. No tomar en cuenta estas indicaciones será la causa de algún mensaje de error al trabajar con esta utilería.

- Todos los elementos y sus valores deben ser escritos en letra minúscula.
- Los segmentos deberán estar correctamente escritos y separados por espacios, tabuladores, o fin de línea (tecla Enter).
- Los elementos deben encontrarse en el orden antes indicado.
- Los grupos de elementos deben estar completos. Por ejemplo, si se especifica "null=y" para el elemento de *llave nula*, el elemento de *valor nulo de la llave* debe estar presente si no se indicó nula la llave, entonces el valor nulo no se debe incluir.
- Se deben especificar todos los elementos para las *n* llaves definidas en el elemento de *número de llaves*. Se debe repetir la información de una llave, empezando con la *posición de la llave*, para cada segmento de llave en el archivo.
- El archivo de descripción no debe contener caracteres de control de procesadores de documentos, debe ser un archivo secuencial o ASCII standard.
- La utilería BUTIL no verifica el fin del archivo, esto puede ser útil si se desea colocar comentarios al final del archivo de descripción.
- El archivo de descripción para los comandos de INDEX y SINDEIX sólo contiene la información de una sola llave segmentada o sencilla.

NOTA: Para facilitar la creación de los archivos de descripción y complementar el trabajo de una forma real, se presenta en el Apéndice E, un programa que permite crear los archivos de descripción de una forma más amigable. Este programa cuenta con ayuda en línea para cualquier interrogante que surja al introducir la información solicitada. El código debe ser ejecutado por el programa QBASIC.EXE que se encuentra entre las utilerías del D.O.S V5.0 en adelante. Por otra parte este programa contiene rutinas generales muy interesantes como el manejo del scroll, la edición de respuestas, manejo de tablas de columnas y renglones que pueden ser útiles para personas que se inician en el lenguaje BASIC.

EJEMPLO DE UN ARCHIVO DE DESCRIPCIÓN.

Como se ha indicado anteriormente en el capítulo IV de este trabajo se presenta una aplicación completa codificada en algunos lenguajes. Estos programas realizan el mantenimiento a un archivo de alumnos de la universidad, el programa lleva el registro de los siguientes datos:

CAMPO	LONGITUD FÍSICA EN Bytes.	POSICIÓN EN	TIPO DE DATO
Número secuencial de entrada al archivo.	2	0-2	Entero
Número de cuenta en la Universidad	10	3-12	Cadena
Nombre completo del alumno	40	13-52	Cadena
Dirección; calle	40	53-92	Cadena
Colonia	35	93-127	Cadena
Delegación o ciudad	25	128-152	Cadena
Estado	25	153-177	Cadena
Teléfonos	16	178-193	Cadena
Generación a la que pertenece	5	194-198	Cadena
Semestre actual	2	199-200	Entero
Tamaño lógico del archivo:		200	

Para este archivo el programa lleva el control de los siguientes índices:

No de llave	Nombre del índice	Características importantes del índice	Posición en el archivo
0	Registro	Autoincrementable	0
1	Cuenta	No acepta duplicados	3
2	Nombre	No acepta duplicados	13
3	Por sem/cuenta	Llave segmentada	1er segmento 199 2o segmento 3
4	Por sem/nombre	Llave segmentada	1er segmento 199 2o segmento 3

Las llaves número 3 y 4, son índices que tienen la información del archivo ordenada primero por el No. de semestre del alumno y dentro de la lista ordenada por semestre se encuentra ordenada por No. de cuenta y por nombre respectivamente. Esto le permite a la aplicación obtener la lista de los alumnos de primer semestre ordenada por cuenta o por nombre por ejemplo.

La llave número 0 lleva un control consecutivo de los registros insertados, al primer registro insertado se le asignará el valor de 1 automáticamente. Esto es con el fin de poder encontrar a un alumno por una llave muy corta semejante a un registro físico.

A continuación se presenta el archivo de descripción para este ejemplo. para poder ejecutar los programas que se incluyen en el trabajo, se debe contar con la estructura de la base de datos de alumnos (archivo Btrieve) que se genera con este archivo de descripción:

Archivo: ALUMNOS.DES

```
record=200 variable=n key=5 page=1024 replace=y data=y
position=1 length=2 duplicates=n modifiable=y type=aut
descending=n alternate=n manual=n null=n segment=n
position=3 length=10 duplicates=n modifiable=y type=str
descending=n alternate=n manual=n null=n segment=n
position=13 length=40 duplicates=n modifiable=y type=str
descending=n alternate=n manual=n null=n segment=n
position=199 length=2 duplicates=y modifiable=y type=int
descending=n alternate=n manual=n null=n segment=y
position=3 length=10 duplicates=y modifiable=y type=str
descending=n alternate=n manual=n null=n segment=n
position=199 length=2 duplicates=y modifiable=y type=int
descending=n alternate=n manual=n null=n segment=y
position=13 length=40 duplicates=y modifiable=y type=str
descending=n alternate=n manual=n null=n segment=n
```

Se debe escribir el contenido de la tabla dentro del archivo de descripción, con el nombre que se desee por ejemplo ALUMNOS.DES.

Sólo resta generar el archivo Btrieve y esto se debe realizar con la siguiente instrucción:

```
C:>BUTIL -CREATE ALUMNOS.BTR ALUMNOS.DES ␣
```

De esta manera se crea automáticamente el archivo Btrieve y con esto se encuentra listo para trabajar con el archivo. Recuerde que para poder ejecutar la utilidad BUTIL se requiere primero, poner en marcha Btrieve, en este caso con los parámetros por omisión, es decir que no se requiere especificar ninguna opción adicional.

OTROS COMANDOS DE LA UTILERÍA BUTIL.

COMANDO	FORMATO	USO
CLONE	butil -clone <arch. nuevo> <arch. Btrieve> [-o<clave>]	Crea un archivo vacío a partir de un archivo existente, copiando su estructura.
COPY	butil -copy <arch. fuente> <arch. destino> [-o<clave1>] [-o<clave2>]]	Copia el contenido del archivo fuente al archivo destino, aunque el archivo destino tenga características diferentes. (campos más grandes, más índices)
DROP	butil -drop <arch. Btrieve> <No. llave> [-o<clave>]	Elimina un índice suplementario, del archivo.
INDEX	butil -index <arch. Btrieve> <arch. de índice> <arch. de descrip.> [-o<clave>]	Genera un índice externo dentro del <i>arch. de índice</i> . Las características de este índice están definidas en el archivo de descripción. (Btrieve no actualiza índices externos.)
LOAD	butil -copy <arch. fuente> <arch. Btrieve> [-o<clave>]	Inserta registros dentro de un archivo Btrieve, contenidos en un archivo ASCII con el formato del archivo generado por el comando SAVE.
RECOVER	butil -recover <arch. Btrieve> <arch. destino> [-o<clave>]	Recupera la información que se encuentra en un archivo dañado.
RESET	butil -reset	Libera todos los recursos utilizados por Btrieve. Por ejemplo cuando un programa se interrumpe y deja los archivos abiertos o registros bloqueados.
SAVE	butil -save <arch. Btrieve> <arch. destino> [<i><y/n></i> <arch. índice> <No. de llave>] [-o<clave>]	Genera un archivo ASCII con la información contenida en el <i>arch. Btrieve</i> . El archivo generado es compatible con el archivo que se requiere en el comando LOAD. La información se puede generar ordenada por alguna llave del archivo o por medio de un índice externo generado con el comando INDEX.
SINDEX	butil -sindex <arch. Btrieve> <arch. de descrip.> [-o<clave>]	Crea un índice suplementario o adicional dentro de un archivo existente. El nuevo índice tomará el siguiente No. de llave.
STAT	butil -stat <arch. Btrieve> [-o<clave>]	Reporta todas las características de un archivo además de indicar el número total de registros en el archivo y en cada índice.
STOP	butil -stop	Remueve a Btrieve de la memoria y libera el espacio para que pueda ser ocupado por el sistema operativo.
VER	butil -ver	Reporta la versión del manejador de registros.

CAPÍTULO III

OPERACIONES

3.01 FORMATO Y PARÁMETROS GENERALES DE UNA OPERACIÓN.

- FORMATO Y PARÁMETROS DE LA FUNCIÓN DE BTRIEVE

Btrieve se presenta en cualquiera de los lenguajes como una función, esta función es definida por la interfaz correspondiente al lenguaje. Como toda función, también requiere un cierto número de parámetros para realizar cualquiera de las operaciones que posee, es decir, que todas las operaciones que se pueden manejar con Btrieve, son realizadas por una sola función. De manera general la función tiene la siguiente forma:

$ST\% = BTRV(\text{operación}\%, \text{block_de_posición}\$, \text{largo_del_buffer_de_datos}\%, \text{buffer_de_datos}\$, \text{número_de_llave}\%, \text{buffer_de_llave}\$)$

donde:

El identificador { $\%$ } representa a una variable entera y { $\$$ } representa una cadena de caracteres (tomando como ejemplo los identificadores de BASIC). Por razones obvias los parámetros son enviados como referencia y no como valor.

-PARÁMETROS DE OPERACIÓN

Btrieve requiere de una aplicación, cierta información en un orden determinado, para poder ejecutar una operación de mantenimiento, una aplicación utiliza los parámetros de función de Btrieve (Ver tabla 3.01).

PARÁMETRO	DESCRIPCIÓN
ST%	Es el resultado de la función y representa el valor del status de Btrieve después de haber ejecutado la operación.
oper%	Es la variable que contiene el número de operación que se desea realizar.
block_de_posición\$	Representa una cadena de caracteres de 128 bytes que contiene la información para acceder a un archivo.
largo_del_buffer_de_datos%	Representa el número de bytes que contiene o espera el <i>buffer de datos</i>
buffer_de_datos\$	Es una cadena de caracteres que normalmente contiene la información del registro de datos.
número_de_llave%	Es la variable que contiene el <i>número de llave</i> con la que se va a trabajar.
buffer_de_llave\$	Representa el valor de la llave con la que se va a operar.

Tabla 3.01
Parámetros de operación.

Una aplicación debe pasar a Btrieve los seis parámetros en cualquier llamada a la función, pero no todos necesitan tener información. Sólo los parámetros que Btrieve utilizará tendrán que ser inicializados con su valor correspondiente, esto depende de la opción a utilizar (Ver Tabla 3.06), a continuación se presenta una descripción de todos los parámetros y como los utiliza Btrieve:

- **CÓDIGO DE OPERACIÓN** (*operación%*).

El parámetro de código de operación le indica a Btrieve que operación se desea realizar (Ver Tabla 3.04 y/o 3.05). La aplicación debe indicar un código de operación válido en cada llamada a la función Btrieve. La variable que se utilice para contener el código de operación, debe ser una variable entera de dos bytes. Btrieve nunca cambia el valor del código de operación.

- **CÓDIGO DE STATUS** (*ST%*).

Cada operación de Btrieve regresa un código de status, informando a la aplicación el resultado de la operación (Ver Tabla A.01). Un status igual a cero, indica que la operación fue realizada sin ningún percance. Si el código es diferente de cero, normalmente esto indica una falla. Sin embargo, algunos códigos diferentes de cero son meramente informativos, (por ejemplo el código 60 y el 64) y asociados con el éxito de la operación.

En la mayoría de los lenguajes de programación, Btrieve es una función entera y el resultado de esta función es el código de status. En algunos lenguajes en particular son necesarios algunos requerimientos especiales ya que no manejan a Btrieve como función, sino como procedimiento

- **BLOCK DE POSICIÓN** (*block_de_posición\$*).

El parámetro de block de posición, es un block de memoria de 128 bytes que Btrieve utiliza para contener los apuntadores de posicionamiento e información necesaria para acceder a un archivo particular. Una aplicación utiliza el block de posición para identificar que archivo se desea acceder en una operación en particular.

Una aplicación debe tener un block de posición único para cada archivo de Btrieve abierto. La aplicación nunca debe escribir sobre el block de posición, sólomente va a ser inicializado por Btrieve para asignarlo a un archivo en la operación de apertura.

- **BUFFER DE DATOS** (*buffer_de_datos\$*).

El *buffer de datos* es un block de memoria que contiene el registros de datos, o información específica requerida en una operación de Btrieve.

Cuando se consulta o se escribe sobre un archivo Btrieve, el *buffer de datos* contiene el registro que la aplicación transfiere hacia el archivo. Por ejemplo, cuando una aplicación consulta un registro de algún archivo, Btrieve lee el registro del archivo y lo escribe dentro del área de memoria asignada para el *buffer de datos* para esa operación. Para otras operaciones, el *buffer de datos* contiene especificaciones de archivo, definiciones, condiciones de consulta y otra información que Btrieve necesita en orden para procesar una operación. Por ejemplo, cuando una aplicación utiliza una operación de *Crear*, ésta construye un *buffer de datos* que contiene las especificaciones del archivo que se desea crear, en el orden que Btrieve las espera. Btrieve lee el *buffer de datos* y crea el archivo de acuerdo a las especificaciones.

Btrieve reconoce el *buffer de datos* como una serie de bytes en memoria. Se puede definir el *buffer de datos*, como una variable de arreglo o una estructura.

- **LARGO DEL BUFFER DE DATOS** (*largo_del_buffer_de_datos%*).

Para cada operación que requiere el *buffer de datos*, la aplicación debe especificar el tamaño de éste buffer en bytes. Esto se requiere por las siguientes razones:

- Btrieve permite definir archivos que contienen registros de longitud variable. Se debe especificar cuantos Bytes se desean leer o escribir, para cada registro.
- Puesto que Btrieve no reconoce ninguna de las estructuras del programa, no tiene manera de saber el *largo del buffer de datos*. Esto crea la posibilidad de escribir datos sin sentido hacia los archivos, o regresar más información de la que el *buffer de datos* puede contener y sobrescribir el área de memoria contigua al *buffer de datos*.

La aplicación debe definir una variable para el *largo del buffer de datos*, como un entero de dos bytes. En todas las operaciones, Btrieve regresa en esta variable el largo del buffer que se está regresando, inclusive si este valor es cero, esto indica que no se regresó ningún dato. Por consiguiente, siempre se debe inicializar el *largo del buffer de datos* con el largo apropiado para una operación antes de que se realice una llamada a la función Btrieve.

Utilice la siguiente guía para inicializar el valor en el parámetro del largo de *buffer de datos*:

- Cuando se trate de leer o escribir a un archivo existente con registros de longitud fija, el valor debe ser igual al largo del registro definido para el archivo.

- Cuando se trate de leer o escribir a un archivo existente con registros de longitud variable, el valor debe ser igual al largo del registro definido para la porción fija del registro, más el número de bytes que se desean leer o escribir después de dicha porción.
- Cuando se utiliza cualquier otra operación, especifique el número exacto del largo del buffer requerido para esa operación. Estos requerimientos son incluidos en la discusión individual de cada operación de Btrieve en este capítulo:
- **BUFFER DE LLAVE** (*buffer_de_llave\$*).

El buffer de llave se utiliza normalmente para definir el valor de llave por ejemplo cuando se utilizan operaciones de consulta.

Se debe asegurar siempre que la variable que se está especificando es lo suficientemente grande para contener el valor requerido por la operación. De otra forma, la operación de Btrieve podrá sobrescribir información colocada a continuación del *buffer de llave*, ocasionando un problema.

- **NÚMERO DE LLAVE** (*número_de_llave%*).

El parámetro de *número de llave*, siempre es un entero con signo de dos bytes. Para la mayoría de las operaciones de Btrieve este parámetro indica cual de los índices del archivo se seguirá en una operación en particular. Para otras aplicaciones, la aplicación utiliza el parámetro de *número de llave* para especificar información tal como: el modo de apertura de un archivo, encriptación, o el disco lógico a utilizar. Btrieve nunca altera el valor de este parámetro.

Cuando se utiliza este parámetro para indicar qué índice se debe seguir, el número debe estar dentro del rango de 0 a 23, dado que Btrieve sólo permite 24 llaves ó llaves segmentadas en el archivo.

3.02 OPERACIONES DE BTRIEVE.

A continuación se describen las operaciones que una aplicación puede utilizar al trabajar con Btrieve. Para cada operación, se presenta la siguiente información:

- Nombre y descripción de la operación.
- Parámetros, y una tabla que ilustra cuales de los seis valores posibles son requeridos en esa función, y cuales son devueltos.
- Prerequisitos; las condiciones que la aplicación debe satisfacer antes de utilizar la operación.
- Procedimiento; los pasos requeridos por la operación para la inicialización de los parámetros.
- Resultado; el resultado de cualquiera de los dos casos cuando la operación fue realizada sin problemas, y cuando no lo fue.
- Posicionamiento; el efecto que tuvo la operación en la posición lógica y/o física actual dentro del archivo.

- REPASO DE LAS OPERACIONES DE BTRIEVE

Para realizar una operación de Btrieve, una aplicación deber completar los siguientes procedimientos:

1. Satisfacer cualquier requisito que la operación necesite. Por ejemplo, antes de que la aplicación pueda realizar cualquier E/S, el archivo se debe encontrar disponible utilizando la operación de *Abrir* para ese archivo.
2. Inicializar los parámetros que necesita la operación de Btrieve. Los parámetros son variables de programa o estructura de datos que corresponden en tipo y tamaño al valor particular que Btrieve espera para la operación dada.
3. Ejecutar la función Btrieve.
4. Evaluar los resultados de la función. Cada operación de Btrieve regresa un código de status. La aplicación debe verificar este código y realizar las acción apropiada. Btrieve también regresa datos u otra información en los parámetros individuales basándose en el propósito de la operación.

- OPERACIONES EN ORDEN ALFABÉTICO:

ABORTAR TRANSACCIÓN (21).

Abortar transacción, termina con la transacción, acumula, y remueve todas las operaciones realizadas desde el inicio de la transacción. Esto desbloquea todos los archivos bloqueados por la transacción. (En el siguiente capítulo se discutirá ampliamente el uso de esta operación)

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Enviar	21					
Resultado						

Prerequisitos:

Se debió utilizar la operación de inicio de transacción antes de usar esta operación.

Procedimiento:

Cambie el código de operación con el número 21. Btrieve ignora todos los demás parámetros.

Resultado:

Si ésta operación fue realizada sin ningún percance, Btrieve regresa el código de status con un valor de cero. Todas las operaciones de *Insertar*, *Actualiza* y *Borrar* utilizadas desde el comienzo de la transacción, serán removidas del archivo.

Si esta operación fue realizada sin ningún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

36 Error de Transacción.

39 Error en las Operaciones de *Fin* o *Abortar Transacción*.

Posicionamiento:

Esta operación no tiene ningún efecto sobre la posición de registros.

ABRIR (0).

Esta operación permite el acceso a un archivo. Una aplicación no puede utilizar ninguna operación hacia un archivo, hasta que éste se encuentre abierto. No es necesario que el archivo se localice en el directorio actual, siempre y cuando se especifique el directorio y/o el drive donde se encuentra. Para la modalidad *basado en servidor* el archivo se debe encontrar dentro del disco del servidor de archivos, no así en el caso de la modalidad de *basado en cliente* la cual permite trabajar inclusive desde discos flexibles.

Cuando una aplicación trata de abrir un archivo dentro de una transacción, y el archivo ya se encuentra bloqueado por otra transacción, la aplicación tendrá que esperar a que se libere el archivo (bloqueo con espera), o bien Btrieve regresa el código de error 85 *Archivo bloqueado* (bloqueo sin espera). En la modalidad *basado en servidor* es factible abrir un archivo fuera de una transacción aunque el archivo esté bloqueado, operación que no se puede realizar en la modalidad *basado en cliente*.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

- El archivo debe existir en el directorio indicado .
- Deben existir menos archivos abiertos de los que la configuración de Btrieve permite.

Procedimiento:

1. Cambie el código de operación con el número 0.
2. Coloque la especificación del archivo en el parámetro de *buffer de llave*.
3. Si el archivo tiene clave de acceso, debe ser indicado en el *buffer de datos*, asegurando que la cadena termine con un cero binario.
4. Asigne el largo de la clave de acceso en el parámetro de *largo del buffer de datos*.
5. Especifique el modo de apertura en el parámetro de *número de llave*

Detalles:

Los modos de apertura de un archivo se encuentran en la siguiente tabla:

ABRIR (0).
(Continuación.)

Modo	Descripción
-1	ACELERADO En este modo, se pierde la capacidad de recuperar datos a partir del archivo de pre-imagen, pero gana velocidad al insertar o actualizar registros. El archivo abierto en este modo queda bloqueado. El número de archivos abiertos en este modo simultáneamente por una aplicación, está limitado por el tamaño de página definido en la configuración.
-2	SÓLO-LECTURA Cuando se abre un archivo en este modo, la aplicación sólo puede utilizar operaciones de consulta, no podrá realizar actualizaciones o inserciones. Si la información en el archivo se encuentra dañada, la única manera de abrir el archivo será ésta y sólo se podrán utilizar las operaciones de consulta de escalón.
-3	VERIFICAR Este modo verifica que todas las operaciones de entrada al archivo se hayan realizado correctamente. Este modo se puede utilizar únicamente en la modalidad <i>basado en cliente</i> . Si un archivo se abre con este modo Btrieve utiliza la opción de Verify del sistema operativo en cada operación.
-4	EXCLUSIVO Este modo permite tener un archivo abierto únicamente por una sola estación de trabajo, hasta que el archivo se cierre.
OTRO	MODO NORMAL Para abrir un archivo en modo normal puede indicar, por ejemplo un cero.

Tabla 3.02
Modos de Apertura

Para acceder un archivo desde algunas versiones de BASIC, son necesarios dos pasos:

1. La aplicación debe ejecutar la instrucción de OPEN de BASIC, con el nombre de archivo "NUL", y especificar el registro con la instrucción FIELD. (El nombre de archivo NUL es un dispositivo, por lo tanto BASIC no abrirá el archivo en disco.)
2. La aplicación debe ejecutar la operación de *Abrir* de Btrieve.

Algunos compiladores actuales de BASIC no requieren de estos pasos, en el capítulo V y Apéndice D se presentan notas sobre estos dos casos.

Btrieve permite abrir hasta 255 archivos en aplicaciones desarrolladas en compiladores de BASIC, Pascal, COBOL, o C. Cuando se tienen varios archivos abiertos, Btrieve utiliza el parámetro de *Block de posición* para determinar sobre cuál archivo se va a realizar la operación.

ABRIR (0).
(Continuación.)

Aunque Btrieve permite abrir 255 archivos simultáneamente, en el caso de BASIC Intérprete y de algunos compiladores de BASIC, el máximo número de archivos que se pueden tener abiertos es de 15. En estos casos en vez de que el block de posición sea de 128 bytes, sólo se requiere el FCB para distinguir los archivos normalmente el FCB se obtiene con la instrucción:

10 FCB% = VARPTR(#1)

Donde #1 el número del archivo indicado en la instrucción de OPEN de BASIC, ver primer ejemplo del capítulo V.

Resultado:

Si esta operación fue realizada sin ningún percance, Btrieve asigna un manejador de archivo del total que está permitido y a partir de ese momento el archivo puede ser procesado.

Si Btrieve tuvo algún problema para realizar esta operación, puede regresar uno de los siguientes códigos diferentes de cero:

- 2 Error de E/S
- 46 Acceso al archivo denegado.
- 84 Registro bloqueado.
- 85 Archivo bloqueado
- 86 Tabla de archivos llena.
- 87 Tabla de manejadores de archivo llena.

Posicionamiento:

Esta operación no tiene ningún efecto sobre la posición de registros.

ACTUALIZAR (3).

Esta operación cambia la información de un registro existente y actualiza todos los cambios realizados en los campos llave.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

- El archivo debe estar abierto.
- Se debe tener establecida una posición física o lógica en el archivo en cuestión.

Procedimiento:

1. Cambie el código de operación con el número 3.
2. Pase el bloque de posición asociado al archivo.
3. Coloque el registro modificado en el *buffer de datos*.
4. Cambie el parámetro de *largo de buffer de datos*, con el largo del registro a actualizar.
5. Cambie el *número de llave* por el número que se utilizó para consultar el registro.

Resultado:

Si esta operación fue realizada sin ningún percance, Btrieve actualiza el registro. Si la información que se modificó corresponde a un segmento de llave o a una llave, Btrieve actualiza automáticamente el o los índices que intervienen en el cambio.

Si la aplicación tiene bloqueado únicamente este registro, después de realizar esta operación el registro se libera.

Si Btrieve tuvo algún problema para realizar esta operación, puede regresar uno de los siguientes códigos diferentes de cero:

- 5 Valor duplicado de la llave
- 7 Número de llave diferente
- 8 Posicionamiento inválido
- 10 Error de valor de llave modificable
- 14 Error de apertura del archivo de pre-imagen
- 15 Error de E/S en el archivo de pre-imagen
- 22 El *buffer de datos* es muy pequeño.
- 80 Conflicto

Posicionamiento:

Esta operación no tiene ningún efecto sobre la posición de registros a menos que se hayan modificado uno o más valores de llaves, en este caso se cambia la posición física y lógica por la del siguiente registro.

ALTO (25).

Esta operación desactiva el programa Btrieve o Brequest de la computadora o estación de trabajo respectivamente, liberando la memoria que ocupaba Btrieve. Después del uso de esta operación ninguna aplicación puede utilizar Btrieve, hasta que se vuelva a poner en marcha.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

Antes de utilizar esta operación Btrieve debe estar activo.

Procedimiento:

1. Cambie el código de operación con el número 25.

Resultado:

Si esta operación fue realizada sin ningún percance, sucede lo siguiente:

1. Si existe una transacción activa es abortada.
2. Libera cualquier recurso bloqueado por la aplicación en la estación de trabajo.
3. Cierra todos los archivos abiertos por la aplicación en la estación de trabajo.
4. Remueve de la memoria a Btrieve o Brequest.

Posicionamiento:

La operación de cerrar, elimina cualquier información física y lógica del archivo, puesto que éstos son cerrados.

BORRAR (4).

Remueve un registro existente dentro de una archivo Btrieve, después de esta acción, el espacio que ocupaba este registro antes de ser borrado se podrá utilizar para colocar un nuevo registro.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

- El archivo de datos debe estar abierto.
- Se debe tener la posición física o lógica del registro a borrar. Este requerimiento se cumple si la operación anterior aplicada en el archivo fue de consulta o de escalón y obtuvo la posición física o lógica deseada.

Procedimiento:

1. Cambie el código de operación con el número 4.
2. Pase el bloque de posición asignado al archivo en el que se desea borrar el registro.

Resultado:

Si esta operación fue realizada sin ningún percance, Btrieve remueve el registro del archivo, libera el bloqueo del registro (si este registro se encontraba bloqueado) y ajusta todos los índices para reflejar la ausencia de este registro.

Si al tratar de realizar la operación, Btrieve se encontró con algún percance puede regresar uno de los siguientes códigos diferentes de cero:

- 7 Número de llave diferente
- 8 Posicionamiento Inválido.
- 14 Error de apertura del archivo de pre-imagen.
- 15 Error de E/S en el archivo pre-imagen.
- 80 Conflicto.

Posicionamiento:

Esta operación elimina la información física y lógica del registro actual pero, la cambia con la del siguiente registro lógico y el registro anterior lógico no modificado.

BORRAR ÍNDICE SUPLEMENTARIO (32)

Esta operación elimina un índice suplementario existente, de un archivo Btrieve.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Enviar	32					
Resultado						

Prerequisitos:

- El archivo debe estar abierto.
- Debe existir en el archivo un índice suplementario.
- El archivo no debe tener una transacción activa.

Procedimiento:

1. Cambie el código de operación con el número 32.
2. Pase el bloque de posición asignado al archivo en el que se desea borrar el índice suplementario.
3. Coloque en el parámetro de *número de llave* el número de índice suplementario a eliminar.

Resultado:

Si esta operación se realizó satisfactoriamente, Btrieve coloca en la lista de espacio libre las páginas que ocupaba el índice eliminado. Los números de índice se decrementaron a partir del número de índice indicado.

Si esta operación fue realizada con algún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 6 Número de llave inválida.
- 41 Operación no permitida.

NOTA: Si el proceso es interrumpido mientras Btrieve está eliminando un índice, es posible acceder la información a través de otros índices. Btrieve regresa el código de error 56 (índice incompleto) si se trata de leer por medio de un índice dañado. En este caso, reutilice esta operación.

Posicionamiento:

Esta operación no tiene ningún efecto sobre la posición de registros. Sin embargo si se elimina el índice que se utilizó para establecer la posición lógica actual, esta posición queda eliminada.

CERRAR (1).

Esta operación cierra un archivo Btrieve asociado con un block de posición específico y libera cualquier bloqueo que la aplicación haya realizado para ese archivo. Un archivo sólo puede ser cerrado, si éste no está dentro de una transacción. La aplicación siempre debe utilizar esta operación cuando se ha terminado el acceso al archivo. La aplicación ya no podrá acceder al archivo hasta que éste se vuelva a abrir.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Enviar	01	01				
Resultado						

Prerequisitos:

- El archivo debe estar abierto.
- La aplicación no debe tener ninguna transacción activa.

Procedimiento:

1. Cambie el código de operación con el número 1.
2. Pase el bloque de posición asociado al archivo que se desea cerrar.

Resultado:

Si esta operación fue realizada sin ningún percance, sucede lo siguiente:

- El bloque de posición del archivo que se cerró, deja de ser válido. La aplicación puede usarlo para otro archivo o utilizar el área de memoria para otros propósitos.
- Cualquier archivo de pre-imagen asociado con la información del archivo cerrado, es borrada si ninguna otra aplicación tiene el mismo archivo abierto.

Si esta operación fue realizada con algún problema, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 3 Archivo no abierto.
- 41 Operación no permitida.

Posicionamiento:

La operación de cerrar, elimina la información física y lógica del archivo.

CONSULTA DIRECTAMENTE (23).

La operación de *Consulta Directamente*, regresa un registro utilizando la posición física en el archivo en vez de utilizar un índice. Se utiliza *Consulta Directamente* para realizar lo siguiente:

- Consultar un registro rápidamente utilizando su dirección física en lugar del valor de una llave.
- Para regresar la dirección de 4 bytes del registro, utilizar la operación de *Obtener Posición*.
- Localizar un registro específico dentro de un grupo de llaves duplicadas, esto permite ahorrar la lectura secuencial que se requiere para llegar a un registro.
- Cambiar el posicionamiento actual utilizando la operación de *Obtener Posición* seguida por la operación *Consulta Directamente* con un número de llave diferente, establece una posición lógica diferente a la original. Un subsecuente uso de la operación *Consulta el Siguiente* regresará el siguiente registro del archivo, basándose en el nuevo posicionamiento de índice.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del Buffer de datos	Buffer de Llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

- El archivo debe estar abierto.
- La aplicación debe conocer el valor de 4 bytes de la locación física del registro. Se puede obtener esta locación con la operación de *Obtener Posición*, la cual regresará la dirección física del archivo para el registro actual.

Procedimiento:

1. Cambie el código de operación con el número 23.
2. Almacene la posición de 4 bytes del registro requerido en los primeros 4 bytes del *Buffer de datos*.
3. Especifique la longitud total del *Buffer de datos*.
4. Especifique el número de llave para el cual *Btrieve* va a establecer un posicionamiento lógico, en el parámetro de número de llave.

CONSULTA DIRECTAMENTE (23).
(Continuación)

Resultado:

Si esta operación fue realizada sin ningún percance, Btrieve almacena el registro en el *Buffer de datos*, el largo del registro, en el parámetro de *largo del Buffer de datos*, y el valor de la llave del índice indicado en el *Buffer de Llave*.

Si esta operación fue realizada con algún contratiempo, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

22 El *Buffer de datos* es muy pequeño.

43 Dirección Inválida del Registro.

Posicionamiento:

La operación de *Consulta Directamente* cambia el posicionamiento definido anteriormente, estableciendo uno diferente de acuerdo al número de llave especificada. Esto no tiene efecto en la información física actual.

CONSULTA DIRECTORIO (18).

La operación de *Consulta Directorio* regresa el directorio actual de una unidad de disco especificada.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del Buffer de datos	Buffer de Llave	Número de llave
Enviar	18					1
Resultado						

Prerequisitos:

La aplicación puede utilizar una operación de *Consulta Directorio* en cualquier momento. La llave de Buffer debe ser menor de 66 caracteres de largo.

Procedimiento:

1. Cambie el código de operación con el número 18.
2. Indique el número del dispositivo de disco lógico en el parámetro *número de llave*. Especifique 1 para el drive A, 2 para B y así sucesivamente. Para que tome el drive por omisión se especifica cero.

Resultado:

Btrieve regresa el directorio actual, terminando con un cero binario, en el *buffer de llave*.

Posicionamiento:

La operación de *Consulta Directorio* no afecta a la información de ningún archivo utilizado.

CONSULTA EL ANTERIOR (7).

Esta operación regresa el registro que se encuentra en la posición lógica anterior a la actual. La operación de *Consulta el Anterior* puede ser utilizada para revisar los registros duplicados de un grupo de registros. Esta operación regresa un status de error 9 (Fin del Archivo) si la posición lógica anterior no existe.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del Buffer de datos	Buffer de Llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

- El archivo debe estar abierto.
- El archivo no puede ser del tipo sólo-datos (sin índices definidos).
- La aplicación debe tener establecida una posición lógica del mismo índice que la llave especificada en esta operación.

Procedimiento:

1. Cambie el código de operación con el número 7.
2. Pase el bloque de posición asociado al archivo que se desea consultar.
3. Guarde o utilice el valor de llave de la operación que estableció el posicionamiento actual en el *Buffer de Llave*. Pase el *Buffer de Llave* exactamente como Btrieve lo regresó en la llamada anterior. Btrieve puede necesitar la información anterior almacenada en el *Buffer de Llave* para determinar la posición actual en el archivo.
4. Especifique el mismo valor en el parámetro de número de llave utilizado en la llamada anterior. No se puede cambiar el índice al utilizar una operación de *Consulta el Anterior*.
5. Especifique el largo del *Buffer de datos*.

CONSULTA EL ANTERIOR (7). (Continuación)

Resultado:

Si esta operación fue realizada sin ningún percance, Btrieve actualiza el *Buffer de Llave* con el valor de llave del nuevo registro, regresa el registro anterior en el *Buffer de datos*, y regresa el largo del registro en el parámetro de largo del *Buffer de datos*.

Si esta operación fue realizada con algún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 3 Archivo no abierto.
- 6 Número de llave Inválido.
- 7 Número de llave Diferente.
- 8 Posicionamiento Inválido.
- 9 Fin del Archivo.
- 22 El *Buffer de datos* es muy pequeño.
- 82 Posición perdida.

Posicionamiento:

La operación de *Consulta el Anterior* establece completamente el posicionamiento lógico y físico haciendo que el registro que consultó se convierta en el registro actual.

CONSULTA EL PRIMERO (12).

La operación *Consulta el primero* regresa el primer registro del índice que corresponde a la llave especificada.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del Buffer de datos	Buffer de Llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

- El archivo debe estar abierto.
- El archivo no puede ser del tipo *sólo datos* (sin índices definidos).

Procedimiento:

1. Cambie el código de operación con el número 12.
2. Pase el bloque de posición que corresponde al archivo.
3. Indique el *número de llave* que corresponde al índice deseado.
4. Especifique la longitud del *buffer de datos*.

Resultado:

Si esta operación fue realizada sin ningún percance, Btrieve regresa el registro solicitado en el *buffer de datos*, almacena el *valor de llave* correspondiente en el *buffer de llave*, y regresa la longitud del registro en el parámetro de *largo de buffer de datos*.

Si esta operación fue realizada con algún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 3 Archivo no abierto.
- 6 Número de llave inválido.
- 9 Fin del Archivo.
- 22 El *buffer de datos* es muy pequeño.

Posicionamiento:

La operación de *Consulta el primero* establece completamente el posicionamiento lógico y físico haciendo que el registro que consultó se convierta en el registro actual. La posición lógica anterior del registro consultado será un punto más allá del principio del archivo (por lo tanto será inválida).

CONSULTA EL SIGUIENTE (6).

Esta operación regresa el registro que se encuentra en la posición lógica siguiente basándose en la llave especificada. La operación *Consulta el siguiente* puede ser utilizada para consultar un grupo de registros que tengan valores de llave duplicados. Esta operación regresa el status 9 (Fin de Archivo) si la siguiente posición lógica no existe.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del Buffer de datos	Buffer de Llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

- El archivo debe estar abierto.
- El archivo no puede ser de tipo *sólo datos* (sin índices definidos).
- La aplicación debe tener establecida una posición lógica del mismo índice que la llave especificada en esta operación.

Procedimiento:

1. Cambie el código de operación con el número 6.
2. Pase el bloque de posición que corresponde al archivo.
3. Guarde o use el *valor de llave* de la operación anterior en el *buffer de datos*. Pase el *buffer de llave* exactamente como Btrieve lo regresó en la llamada anterior, Btrieve puede necesitar la información anterior almacenada para determinar la posición actual en el archivo.
4. Mandé el *número de llave* utilizado en la llamada anterior. No se debe cambiar el índice utilizado.
5. Especifique el largo del *buffer de datos*.

CONSULTA EL SIGUIENTE (6).
(Continuación)

Resultado:

Si esta operación fue realizada sin ningún percance, Btrieve regresa el registro en el *buffer de datos*, el valor de llave para el registro en el *buffer de llave*, y el largo del registro en el parámetro de *largo del buffer de datos*.

Si esta operación fue realizada con algún inconveniente, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 3 Archivo no abierto.
- 6 Número de llave inválido.
- 7 Número de llave diferente.
- 8 Posicionamiento inválido.
- 9 Fin del Archivo.
- 22 El *buffer de datos* es muy pequeño.
- 82 Posición Perdida.

Posicionamiento:

La operación de *Consulta el siguiente* establece completamente el posicionamiento lógico y físico, haciendo que el registro que consultó se convierta en el registro actual. La siguiente posición lógica del registro consultado será un punto más allá del final del archivo (por lo tanto será inválida).

CONSULTA EL ÚLTIMO (13).

Esta operación regresa la última posición del índice que corresponde a la llave especificada. Si el valor de la llave que se encuentra al final del índice tiene valores duplicados, esta operación consulta el último del grupo.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del Buffer de datos	Buffer de Llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

- El archivo debe estar abierto.
- El archivo no puede ser de tipo *sólo datos* (sin índices definidos).

Procedimiento:

1. Cambie el código de operación con el número 13.
2. Pase el bloque de posición que corresponde al archivo.
3. Mande el largo del *buffer de datos*.
4. Especifique el *número de llave* que corresponde con el índice deseado.

Resultado:

Si la operación se realiza sin ningún percance, Btrieve regresa el registro en el *buffer de datos*, almacena el valor correspondiente de llave en el *buffer de llave*, y regresa el largo del registro en el parámetro del *largo del buffer de datos*.

Si esta operación fue realizada con algún problema, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 3 Archivo no abierto.
- 6 Número de llave inválido.
- 9 Fin del archivo.
- 22 El *buffer de datos* es muy pequeño.

Posicionamiento:

La operación de *Consulta el último* establece completamente el posicionamiento lógico y físico haciendo que el registro que consultó se convierta en el registro actual. La siguiente posición lógica del registro consultado será un punto más allá del principio del archivo (por lo tanto será inválida).

CONSULTA IGUAL A (5).

Esta operación regresa el registro en el cual el campo especificado por el *número de llave* sea igual al valor dado en el *buffer de llave*. Si la llave tiene valores duplicados, esta operación regresa el primer registro en orden cronológico del grupo con el mismo valor de llave.

Parámetros:

	Operación	Bloque de Posición	Buffer de datos	Largo del Buffer de datos	Buffer de Llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

- El archivo debe estar abierto.
- El archivo no puede ser de tipo *sólo datos* (sin índices definidos).

Procedimiento:

1. Cambie el código de operación con el número 5.
2. Pase el bloque de posición que corresponde al archivo.
3. Especifique el valor de llave deseado en el *buffer de llave*.
4. Mande el *número de llave* que corresponde al índice deseado.
5. Inicialice el *largo del buffer de datos* a un valor igual al del largo del registro que se desea consultar.

Resultado:

Si esta operación fue realizada sin ningún contratiempo, Btrieve regresa el registro en el *buffer de datos*, y el largo del registro en bytes en el parámetro de *largo del buffer de datos*.

Si esta operación fue realizada con algún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 3 Archivo no abierto.
- 4 Valor de llave no encontrado.
- 6 Número de llave inválido.
- 22 El *buffer de datos* es muy pequeño.

Posicionamiento:

Esta operación establece completamente el posicionamiento lógico y físico haciendo que el registro que consultó se convierta en el registro actual.

CONSULTA MAYOR A (8).

Esta operación regresa un registro el cual será mayor que el valor del *buffer de llave* especificado . Si la llave tiene valores duplicados, esta operación regresa el primer registro en orden cronológico del grupo con el mismo valor de llave.

Nota: si se utiliza esta operación sobre llaves descendentes, un valor mayor se refiere a un valor menor que el indicado en el *buffer de llave*.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del Buffer de datos	Buffer de Llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

- El archivo debe estar abierto.
- El archivo no puede ser de tipo *sólo datos* (sin índices definidos).

Procedimiento:

1. Cambie el código de operación con el número 8.
2. Pase el bloque de posición que corresponde al archivo.
3. Especifique el valor de llave deseado en el *buffer de llave*.
4. Mande el *número de llave* que corresponde con el índice correcto.
5. Especifique el largo del *buffer de datos*.

Resultado:

Si esta operación fue realizada sin ningún percance, Btrieve regresa el registro en el *buffer de datos*, y el largo del registro en el parámetro de *largo del buffer de datos*.

Si esta operación fue realizada con algún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 3 Archivo no abierto.
- 6 Número de llave inválido.
- 22 El *buffer de datos* es muy pequeño.

Posicionamiento:

Esta operación establece completamente el posicionamiento lógico y físico haciendo que el registro que consultó se convierta en el registro actual.

CONSULTA MAYOR O IGUAL A (9).

Esta operación regresa un registro el cual será mayor o igual que el valor del *buffer de llave* especificado. Si la llave tiene valores duplicados, esta operación regresa el primer registro en orden cronológico del grupo con el mismo valor de llave.

Nota: si se utiliza esta operación sobre llaves descendentes, un valor mayor o igual se refiere a un valor menor o igual que el indicado en el *buffer de llave*.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del Buffer de datos	Buffer de Llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

- El archivo debe estar abierto.
- El archivo no puede ser de tipo sólo-datos (sin índices definidos).

Procedimiento:

1. Cambie el código de operación con el número 9.
2. Pase el bloque de posición que corresponde al archivo.
3. Especifique el valor de llave deseado en el *buffer de llave*.
4. Mande el *número de llave* que corresponde con el índice correcto.
5. Especifique el largo del *buffer de datos*.

Resultado:

Si esta operación fue realizada sin ningún percance, Btrieve almacena el registro en el *buffer de datos*, y regresa el largo del registro en el parámetro de *largo del buffer de datos*.

Si esta operación fue realizada con algún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 3 Archivo no abierto.
- 6 Número de llave inválido.
- 22 El *buffer de datos* es muy pequeño.

Posicionamiento:

Esta operación establece completamente el posicionamiento lógico y físico haciendo que el registro que consultó se convierta en el registro actual.

CONSULTA MENOR A (10).

Esta operación regresa un registro el cual será menor que el valor del *buffer de llave* especificado. Si la llave tiene valores duplicados, esta operación regresa el último registro en orden cronológico del grupo con el mismo valor de llave.

Nota: si se utiliza esta operación sobre llaves descendentes, un valor menor se refiere a un valor mayor que el indicado en el *buffer de llave*.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del Buffer de datos	Buffer de Llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

- El archivo debe estar abierto.
- El archivo no puede ser de tipo *sólo datos* (sin índices definidos).

Procedimiento:

1. Cambie el código de operación con el número 10.
2. Pase el bloque de posición que corresponde al archivo.
3. Especifique el valor de llave deseado en el *buffer de llave*.
4. Mande el *número de llave* que corresponde con el índice correcto.
5. Especifique el largo del *buffer de datos*.

Resultado:

Si esta operación fue realizada sin ningún percance, Btrieve regresa el registro en el *buffer de datos*, el valor de llave para el registro en el *buffer de llave* y el largo del registro en el parámetro de *largo del buffer de datos*.

Si esta operación fue realizada con algún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 3 Archivo no abierto.
- 6 Número de llave inválido.
- 22 El *buffer de datos* es muy pequeño.

Posicionamiento:

Esta operación establece completamente el posicionamiento lógico y físico haciendo que el registro que consultó se convierta en el registro actual.

CONSULTA MENOR O IGUAL A (9).

Esta operación regresa un registro el cual será menor o igual que el valor del *buffer de llave* especificado. Si la llave tiene valores duplicados, esta operación regresa el último registro en orden cronológico del grupo con el mismo valor de llave.

Nota: si se utiliza esta operación sobre llaves descendentes, un valor menor se refiere a un valor mayor que el indicado en el *buffer de llave*.

Parámetros:

	Operación	Bloque de Posición	Buffer de datos	Largo del Buffer de datos	Buffer de Llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

- El archivo debe estar abierto.
- El archivo no puede ser de tipo *sólo datos* (sin índices definidos).

Procedimiento:

1. Cambie el código de operación con el número 11.
2. Pase el bloque de posición que corresponde al archivo correcto.
3. Especifique el valor de llave deseado en el *buffer de llave*.
4. Mandé el *número de llave* a la deseada tabla indexada.
5. Especifique el largo del *buffer de datos*.

Resultado:

Si esta operación fue realizada sin ningún percance, Btrieve regresa el registro en el *buffer de datos*, el valor de llave para el registro en el *buffer de llave* y regresa el largo del registro en el parámetro de *largo del buffer de datos*.

Si esta operación fue realizada con algún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 3 Archivo no abierto.
- 6 Número de llave inválido.
- 22 El *buffer de datos* es muy pequeño.

Posicionamiento:

Esta operación establece completamente el posicionamiento lógico y físico haciendo que el registro que consultó se convierta en el registro actual.

CREAR (14).

La operación de Crear permite generar un archivo dentro de una aplicación, esta realiza esencialmente la misma función que el comando CREATE de la utilidad BUTIL descrita anteriormente.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

Si se está creando un archivo vacío sobre un archivo existente, asegúrese de que el archivo existente esté cerrado antes de ejecutar esta operación.

Procedimiento:

1. Cambie el código de operación con el número 14.
2. Indique las especificaciones de archivo, las especificaciones de llave, y cualquier secuencia alternativa en el *buffer de datos*, éste se describirá posteriormente en el apartado de "Detalles".
3. Especifique el *largo del buffer de datos*.
4. Si desea que Btrieve le advierta sobre la existencia de un archivo con el mismo nombre, se debe asignar al parámetro de *número de llave* el valor de -1, de otra manera, se debe asignar el valor de 0.
5. Especifique el directorio para el archivo en el *buffer de llave*, asegúrese de que el directorio termine en blanco o con el 0 binario.

Detalles:

La siguiente sección describe en detalle como almacenar la definición del archivo Btrieve en el buffer de datos y algunas indicaciones para los demás parámetros.

CREAR (14).
(Continuación.)

- ESPECIFICACIONES EN EL PARÁMETRO DE *BUFFER DE DATOS*.

La siguiente sección describe como especificar lo siguiente:

- Especificaciones de archivo.
- Características de Llave.
- Secuencia alternativa de ordenamiento.

Btrieve supone que el buffer de datos presenta un formato como se muestra en la siguiente tabla

	DESCRIPCIÓN	LARGO
Especificaciones de Archivo	Largo del registro	2
	Tamaño de página	2
	Número de índices	2
	Espacios no utilizados	4
	Banderas de archivo	2
	Palabra Reservada	2
	Precolocar	2

	DESCRIPCIÓN	LARGO
Especificaciones de Llave (se repite según el número de segmentos de llaves)	Posición de Llave	2
	Largo de Llave	2
	Banderas de Llave	2
	Espacio no utilizado	4
	Tipo de llave (ver códigos en el anexo B, tabla B.1)	1
	Valor de llave nula	1
	Espacio Reservado	4

Estructura del buffer de datos para la operación *Crear*

Especificaciones de Archivo. Se almacenan las especificaciones de archivo en los primeros 16 bytes de el *buffer de datos*. Los bytes son numerados empezando con un cero. Se almacena la información en palabras (variables enteras) para el largo del registro, tamaño de página y el número de índices. Para crear un archivo de tipo *sólo datos*, coloque en el número de índices un cero.

Se pueden usar las áreas de "no utilizado" y "reservado" del buffer de datos, pero Btrieve no las utiliza para esta operación. Se deben inicializar las áreas reservadas con un cero binario para mantener la compatibilidad con futuras acciones de Btrieve.

CREAR (14).
(Continuación.)

Las especificaciones en la bandera de archivo se indican en bits y estos definen si el archivo permitirá registros de longitud variable, truncación de blancos, o compresión de datos, y también si Btrieve apartará espacio en disco para el archivo (precolocar). Se deben utilizar los bits más altos del byte menos representativo para indicar el espacio que queda libre entre registros de longitud variable. El byte más representativo no se utiliza.

Los bits de las banderas de archivo se encuentran numerados desde el 0 hasta el 15, el bit cero es el menos representativo. Coloque los bits de acuerdo con la siguiente descripción.

- Bit 0 = 1 Btrieve permite que el archivo contenga registros de longitud variable.
- Bit 1 = 1 Btrieve trunca espacios en blanco en los registros de longitud variable.
- Bit 2 = 1 Btrieve graba el número de páginas especificadas en el espacio de precolocar.
- Bit 3 = 1 Btrieve comprime la información que contenga el archivo.
- Bit 4 = 1 Btrieve crea un archivo de tipo sólo llave.
- Bit 6 = 1 Btrieve mantiene un 10% de espacio de separación entre páginas que contienen registros de longitud variable.
- Bit 7 = 1 Btrieve mantiene un 20% de espacio de separación entre páginas que contienen registros de longitud variable.
- Bit 6 = 1 y Bit 7 = 1 Btrieve mantiene un 30% de espacio de separación entre páginas que contienen registros de longitud variable.

La siguiente tabla muestra la representación binaria y decimal de las banderas de archivo:

VALORES	BINARIO	DECIMAL
Registros de longitud variable	00000001	1
Truncación de Blancos	00000010	2
Precolocación	00000100	4
Compresión de datos	00001000	8
Archivo de tipo <i>sólo llave</i>	00010000	16
10% de separación	01000000	64
20% de separación	10000000	128
30% de separación	11000000	192

Tabla 3.03
Valores de bandera de archivo.

CREAR (14).
(Continuación.)

Si se necesita especificar una combinación de los atributos de archivo, sume sus respectivos valores de bandera. Por ejemplo, si se requiere especificar que el archivo contendrá registros de longitud variable y que utilizará la truncación de blancos, se debe inicializar el entero que contiene las banderas con el número 3 (2+1). Btrieve ignora la truncación de blancos y la separación de páginas, si el archivo no permite registros de longitud variable (Bit 0 = 0).

Si se prende la bandera de *precolocación*, se debe colocar el número de páginas a generar en el entero de *precolocar*.

Características de Llave. Asigne las características de la llave inmediatamente después de las especificaciones de archivo. Se requieren de 16 bytes para cada segmento de llave en el archivo.

Los códigos del tipo de llave y la definición del caracter nulo ocupan un byte cada uno. Se debe almacenar la información para la posición de la llave y el largo de la llave como números enteros.

Las banderas de llave se conforman de acuerdo con la siguiente descripción:

Bit 0 = 1 La llave acepta duplicados.

Bit 1 = 1 La llave es modificable.

Bit 2 = 0 y La llave es de tipo cadena.
Bit 8 = 0

Bit 2 = 1 y La llave es binaria.
Bit 8 = 0

Bit 3 = 1 La llave tiene un valor nulo.

Bit 4 = 1 La llave es segmentada.

Bit 5 = 1 La llave es almacenada de acuerdo a una secuencia alternativa de ordenamiento.

Bit 6 = 1 La llave es almacenada en orden descendente.

Bit 7 Se debe ignorar en esta operación.

Bit 8 = 0 La llave es de tipo estándar.

CREAR (14).
(Continuación.)

Bit 8 = 1 La llave no es de tipo estándar .

Bit 9 = 1 La llave es manual.

Aunque Btrieve ignora el bit 7, en esta operación, se debe inicializar con 0 cuando se crea un archivo. Cuando se utiliza la operación de *Estado* (15), Btrieve coloca un 1 en el bit 7 si la llave corresponde a un índice suplementario.

La siguiente tabla muestra los valores binario, hexadecimal y decimal para las banderas de llave:

ATRIBUTO	BINARIO	HEXADECIMAL	DECIMAL
Duplicado	00000001	01	1
Modificable	00000010	02	2
Binario	00000100	04	4
Nulo	00001000	08	8
Segmentada	00010000	10	16
Seq. Alt. Ord.	00100000	20	32
Descendente	01000000	40	64
Suplemental	10000000	80	128
Tipo de llave no estándar.	1 00000000	100	256
Manual	10 00000000	200	512

Tabla 3.03 bis.
Valores de bandera de llave.

Asigne los mismos atributos de duplicado, modificable, manual y nulo para todos los segmentos de la misma llave. Si se especifican los atributos de nulo o manual para una llave, es factible asignar diferentes caracteres nulos para cada segmento.

Los atributos de las llaves segmentadas son; una bandera indicando que el siguiente block de llaves (en el buffer de datos) se refiere al siguiente segmento que comprende la misma llave. Además, se puede hacer cada llave segmentada, ascendente o descendente y especificar cualquier tipo de dato.

Por ejemplo, para crear un archivo con dos llaves, la primera con dos segmentos y la segunda con un segmento, utilice el cuarto bit de la bandera de llave como sigue:

- En el primer bloque (de 16 bytes) de llave, coloque el valor de 1 en el bit 4, esto indica que el siguiente bloque es un segmento de la misma llave.
- En el segundo bloque de llave, coloque el valor de 0 en el bit 4, indicando que es el último segmento de la primera llave.
- En el tercer bloque de llave, coloque el valor de 0 en el bit 4, indicando que es el primer y último segmento de la segunda llave.

CREAR (14).
(Continuación.)

Secuencia Alternativa de Ordenamiento. Se puede especificar una secuencia alternativa de ordenamiento para ordenar cualquier llave en el archivo. Sin embargo, se debe especificar sólo una secuencia alternativa de ordenamiento para el archivo. Se puede indicar que algunos segmentos o llaves simples serán ordenadas con la secuencia ASCII estándar y otros segmentos o llaves simples serán ordenados en base a la secuencia alternativa.

Sólo es posible especificar la secuencia alternativa de ordenamiento para los tipos de llave L-cadena, Z-cadena, y cadena. Si se prende la bandera de secuencia alternativa para cualquier llave o segmento de llave en el archivo, la definición de la secuencia alternativa de ordenamiento se debe colocar inmediatamente después de la última especificación de block de llave.

La secuencia alternativa consiste de nueve bytes de cabecera seguidos por 256 caracteres que indican el orden a seguir como se explicó anteriormente, en el capítulo II en el punto de Secuencia alternativa de ordenamiento.

NOTA: Si se crean múltiples archivos con diferentes secuencias alternativas, utilice un nombre diferente para cada secuencia.

- CALCULO DEL PARÁMETRO DE *LARGO DEL BUFFER DE DATOS.*

El largo del buffer de datos debe ser calculado correctamente para que el número indicado permita incluir en el buffer de datos, las especificaciones de archivo, las características de llave, y una secuencia alternativa, si es que ésta se define. No debe especificar el largo del registro del archivo en este parámetro como en las demás operaciones.

Por ejemplo, para crear un archivo que tenga dos llaves segmentadas y una secuencia alternativa, el buffer de datos debe tener un largo de 313 bytes como sigue:

Espec. Archivo	+	Espec. Llave 1	+	Espec. Llave 2	+	Secuencia Alternativa	
16	+	16	+	16	+	265	= 313

- INDICACIONES PARA EL PARÁMETRO DE *NÚMERO DE LLAVE.*

El *número de llave* puede ser utilizado de dos maneras dependiendo de si Btrieve está o no en un ambiente de NetWare.

CREAR (14). (Continuación.)

En la modalidad de basado en cliente, se puede utilizar el parámetro de número de llave para especificar el nivel de advertencia que utiliza Btrieve;

- *Si el archivo a crear ya existe y se desea reemplazar el archivo existente con el nuevo.* Coloque en el parámetro del número de llave un valor no negativo, preferiblemente cero.
- *Si no desea que Btrieve reemplace el archivo existente con el nuevo archivo,* coloque el valor de -1 en el parámetro de número de llave. Si un archivo del mismo nombre ya existe, Btrieve regresa un código diferente de cero y no genera el nuevo archivo.

Si se está utilizando Btrieve en la modalidad de basado en servidor, use el número de llave para especificar si Btrieve debe advertir de la existencia de un archivo con el mismo nombre y para definir si Btrieve respetará la opción de transaccional/no-transaccional, especificada durante el BSETUP. Si se elimina la opción del setup, Btrieve crea un archivo no transaccional, sin tomar en cuenta la configuración del setup.

En la modalidad de basado en servidor, se especifica el número de llave de las siguientes maneras:

Para preservar la opción de Setup

- Si se quiere que Btrieve genere un nuevo archivo sobre el archivo existente o si no se desea que se verifique la existencia de un archivo, coloque en el parámetro de número de llave un valor no negativo, preferentemente el cero.
- Si no desea que Btrieve genere un nuevo archivo sobre el archivo existente, coloque el valor de -1 en el parámetro del número de llave. Si un archivo con el mismo nombre ya existe, Btrieve regresa un status diferente de cero y no genera el nuevo archivo.

Para eliminar la opción de Setup

- Si se quiere que Btrieve genere un nuevo archivo sobre el archivo existente o si no se desea verificar la existencia de un archivo, coloque en el parámetro de número de llave el valor de -2.
- Si no desea que Btrieve genere un nuevo archivo sobre el archivo existente, coloque el valor de -3 en el parámetro del número de llave. Si un archivo con el mismo nombre ya existe Btrieve regresa un status diferente de cero y no genera el nuevo archivo.

CREAR (14).
(Continuación.)

Resultado:

Si esta operación fue realizada sin ningún percance, Btrieve advierte de la existencia de un archivo con el mismo nombre o bien, crea el nuevo archivo de acuerdo con las especificaciones indicadas. El nuevo archivo no contiene ningún registro. Esta operación no abre el archivo. La aplicación debe ejecutar la operación de *Abrir* antes de poder accederlo.

Si esta operación fue realizada con algún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 2 Error de E/S.
- 18 Disco lleno.
- 22 El *buffer de datos* es muy pequeño.
- 24 El tamaño de página es muy pequeño.
- 25 Error de E/S al crear un archivo.
- 26 *Número de llaves* inválido.
- 27 Posición de llave inválida.
- 28 Tamaño de registro inválido.
- 29 Tamaño de llave inválido.
- 48 Definición de secuencia alternativa inválida.
- 49 Error de tipo de llave.
- 59 Archivo ya existe.

Posicionamiento:

Esta operación no tiene ningún efecto sobre la posición de registros.

CREAR ÍNDICE SUPLEMENTARIO (31).

Adiciona un índice dentro de un archivo Btrieve, y este índice se considera como suplementario, es decir, que podrá ser borrado con la operación número 32.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Enviar	☐	☐	☐	☐		
Resultado						

Prerequisitos:

- El archivo Btrieve debe estar abierto.
- El número de segmentos de llave existentes dentro del archivo debe ser menor o igual a la siguiente fórmula:

$$24 - (\# \text{ de segmentos que se desean adicionar})$$

- Se debe estar seguro que las características de las llaves, la posición, y el largo del nuevo índice, son apropiados para el archivo en el que se desea adicionar el índice.
- No debe haber transacciones activas.

Procedimiento:

1. Cambie el código de operación con el número 31.
2. Pase el bloque de posición asociado al archivo en que se desea adicionar el nuevo índice.
3. Coloque las especificaciones de llave para el nuevo índice en el *buffer de datos*. El *buffer de datos* consiste en 16 bytes por cada segmento de un índice suplementario que se esté creando. Utilice el mismo bloque de especificación de llave que se ocupa en la operación de *Crear*.
4. Cambie el parámetro de *largo de buffer de datos* con el número de bytes que contiene el *buffer de datos*. Para un nuevo índice que no utiliza secuencia alternativa de ordenamiento, utilice la siguiente fórmula para determinar el largo correcto del buffer: $16 * (\# \text{ de segmentos})$.

Si la nueva llave tiene una secuencia alternativa, utilice la siguiente formula: $16 * (\# \text{ de segmentos}) + 265$.

CREAR ÍNDICE SUPLEMENTARIO (31). (Continuación)

Resultado:

Si esta operación fue realizada sin ningún percance, Btrieve inmediatamente adiciona el nuevo índice al archivo. El tiempo requerido para esta operación depende del total del número de registros que deben ser ordenados, el tamaño del archivo, y el tamaño del nuevo índice.

Si esta operación fue realizada sin ningún percance, el *número de llave* para este nuevo índice es el siguiente de la lista de llaves. Por ejemplo, si se tienen tres llaves para el archivo: cero, uno, dos; si se adiciona un índice suplementario, este tomará el *número de llave* tres. Se puede utilizar el nuevo índice para acceder la información, después que la operación se complete.

Si esta operación fue realizada satisfactoriamente, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 22 El *buffer de datos* es muy pequeño.
- 27 Posición de llave inválida.
- 41 Operación no permitida.
- 45 Bandera de llave inconsistente.
- 49 Error de tipo de llave.
- 56 Índice incompleto.

Posicionamiento:

Esta operación no tiene ningún efecto sobre ningún registro.

DESBLOQUEAR (27).

Esta operación libera uno o más registros bloqueados. Esto permite que otra estación de trabajo pueda bloquear el registro. Para más información sobre su utilidad ver capítulo IV y ejemplos del capítulo V en las opciones de cambios y bajas de los programas muestra, específicamente.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Enviar	☐	☐	☐	☐		☐
Resultado						

Prerequisitos:

La aplicación debe tener al menos un registro bloqueado.

Procedimiento:

Para bloquear un sólo registro se debe realizar lo siguiente:

1. Cambie el código de operación con el número 27.
2. Pase el bloque de posición asociado al archivo que contiene el registro bloqueado.
3. Cambie el valor del *número_de_llave* a un valor positivo o cero para indicar el bloqueo individual.

Para desbloquear un registro de un grupo de registros bloqueados, primero se debe obtener la posición física del registro (Ver *Obtener Posición* operación No. 22) que se desea desbloquear. Entonces utilice la operación desbloquear como sigue:

1. Cambie el código de operación con el número 27.
2. Pase el bloque de posición asociado al archivo que contiene los registros bloqueados.
3. Coloque en el *Buffer_de_datos* la posición obtenida de 4 bytes correspondiente al registro.
4. Cambie el valor del *largo_de_buffer_de_datos* con el valor de 4.
5. Cambie el valor del *número_de_llave* a -1.

Para desbloquear todo el grupo de registros bloqueados, se deben realizar los siguientes pasos:

1. Cambie el código de operación con el número 27.
2. Pase el bloque de posición asociado al archivo que contiene el grupo de registros bloqueados.
3. Cambie el valor del *número_de_llave* a -2.

Resultado:

Si esta operación fue realizada sin ningún percance, sucede lo siguiente:

- Btrieve libera todos los registros especificados por la operación.

Si esta operación fue realizada con algún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

81 Error de bloqueo.

Posicionamiento:

Esta operación no tiene ningún efecto sobre la posición de registros.

ELIMINAR CLAVE (30).

Remueve la clave de acceso que anteriormente se asignó para el archivo con la operación *Fijar Clave*.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

- El archivo Btrieve debe estar abierto.
- No debe haber transacciones activas para el archivo.

Procedimiento:

1. Cambie el código de operación con el número 30.
2. Pase el bloque de posición que identifica al archivo.

Resultado:

Después de esta operación, Btrieve no va a requerir la clave de acceso para el archivo. Si el archivo se encontraba anteriormente encriptado, Btrieve desencriptará la información durante ésta operación.

Si la operación no se realizó satisfactoriamente, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 3 Archivo no Abierto.
- 41 Operación no permitida.

Posicionamiento:

Esta operación no tiene ningún efecto sobre la posición de registros.

ESCALÓN ANTERIOR (35)

Esta operación regresa el registro que se encuentra en la posición física anterior al registro actual. Btrieve no utiliza ningún índice al consultar el registro en una operación de *escalón anterior*.

Las operaciones de Escalón se pueden utilizar inmediatamente después de cualquier operación de Consulta o Escalón, establecen el posicionamiento físico del registro.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del Buffer de datos	Buffer de Llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

- El archivo debe estar abierto.
- La aplicación debe tener establecida la posición física.

Procedimiento:

1. Cambie el código de operación con el número 35.
2. Pase el bloque de posición que corresponde al archivo.
3. Especifique el *buffer de datos* en el cual se almacenará el registro.
4. Especifique el largo del *buffer de datos*.

Resultado:

Si esta operación fue realizada sin ningún percance, Btrieve regresa en el *buffer de datos* el registro anterior a la posición física del registro actual y pone en el parámetro del *largo del buffer de datos* el número de bytes consultados en el *buffer de datos*.

Si esta operación fue realizada con algún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 3 Archivo no abierto.
- 9 Fin del archivo.
- 22 El *buffer de datos* es muy pequeño.

Posicionamiento:

La operación de *escalón anterior* no establece la posición lógica del registro consultado, sólo la posición física actual basándose en el registro recobrado.

ESCALÓN FINAL (34)

Esta operación regresa el último registro del archivo. Btrieve no utiliza ningún índice para encontrar el registro.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del Buffer de datos	Buffer de Llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

El archivo debe estar abierto.

Procedimiento:

1. Cambie el código de operación con el número 34.
2. Pase el bloque de posición que corresponde con el archivo.
3. Especifique el *buffer de datos* en el cual se almacenará el registro.
4. Coloque el parámetro *largo del buffer de datos* para el largo del *buffer de datos*.

Resultado:

Si esta operación fue realizada sin ningún percance, Btrieve regresa la última posición física del archivo en el *buffer de datos* y pone en el parámetro del *largo del buffer de datos* el número de bytes consultados.

Si esta operación fue realizada con algún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 3 Archivo no abierto.
- 9 Fin del archivo.
- 22 El *buffer de datos* es muy pequeño.

Posicionamiento:

La operación de *escalón final* destruye la posición lógica actual, sólo establece la posición física tomando como base el registro recuperado. La siguiente posición física apunta más allá del fin del archivo (por lo tanto es inválida).

ESCALÓN INICIAL (33)

Esta operación regresa el primer registro del archivo. Btrieve no utiliza ningún índice para consultar el registro.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del Buffer de datos	Buffer de Llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

El archivo debe estar abierto.

Procedimiento:

1. Cambie el código de operación con el número 33.
2. Pase el bloque de posición que corresponde con el archivo.
3. Especifique el *buffer de datos* en el cual se almacenará el registro.
4. Coloque el parámetro *largo del buffer de datos* para el largo del *buffer de datos*.

Resultado:

Si esta operación fue realizada sin ningún percance, Btrieve regresa el primer registro del archivo en el *buffer de datos* y pone en el parámetro del *largo del buffer de datos* el número de bytes consultados.

Si esta operación fue realizada con algún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 3 Archivo no abierto.
- 9 Fin del archivo.
- 22 El *buffer de datos* es muy pequeño.

Posicionamiento:

Esta operación destruye la posición lógica actual, sólo establece la posición física tomando como base el registro recuperado. La posición física anterior apunta más allá del principio del archivo (por lo tanto es inválida).

ESCALÓN SIGUIENTE (24)

Esta operación regresa el registro que se encuentra en la siguiente posición física establecida. Btrieve no utiliza ningún índice para consultar.

Si se utiliza esta operación después de abrir el archivo se obtiene el primer registro del archivo.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del Buffer de datos	Buffer de Llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

- El archivo debe estar abierto.
- La aplicación debe tener establecida una posición física

Procedimiento:

1. Cambie el código de operación con el número 24.
2. Pase el bloque de posición que corresponde con el archivo.
3. Especifique el *buffer de datos* en el cual se almacenará el registro.
4. Especifique el largo del *buffer de datos*.

Resultado:

Si ésta operación fue realizada sin ningún percance, Btrieve regresa la siguiente posición física del registro en el *buffer de datos* y pone en el parámetro del *largo del buffer de datos*, el número de bytes consultados.

Si esta operación fue realizada con algún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 3 Archivo no abierto.
- 9 Fin del archivo.
- 22 El *buffer de datos* es muy pequeño.

Posicionamiento:

La operación de *escalón siguiente* no establece la posición lógica del registro consultado, sólo la posición física actual basándose en el registro recobrado.

ESTADO (15).

Esta operación consulta las características actuales de un archivo especificadas cuando fue creado. Además regresa el número de registros en el archivo, el número de diferentes valores de llave para cada índice, el número de páginas sin uso.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Enviar						
Resultado						

Detalles:

Btrieve regresa las características del archivo en el buffer de datos en el mismo formato binario que como lo hace para una operación de *Crear*. Para las variables de cuatro bytes (número de llaves y registros), los primeros dos bytes son los menos significativos. Las áreas reservadas son ignoradas en esta operación.

Btrieve regresa el *buffer de datos* en el formato mostrado en la siguiente tabla:

Especificaciones
de Archivo

DESCRIPCION	LARGO
Largo del registro	2
Tamaño de página	2
Número de índices	2
Número de registros	4
Banderas de archivo	2
Palabra Reservada	2
Páginas no utilizadas	2

Especificaciones
de Llave

DESCRIPCION	LARGO
Posición de Llave	2
Largo de Llave	2
Banderas de Llave	2
Número de llaves	4
Tipo de llave no estándar	1
Valor de llave nula	1
Espacio Reservado	4

Estructura del buffer de datos para la operación *Estado*.

ESTADO (15).
(Continuación.)

Las banderas de archivo aparecen como se muestra a continuación:

- Bit 0 = 1 El archivo permite registros de longitud variable.
- Bit 1 = 1 Btrieve trunca los blancos en la porción variable del registro.
- Bit 2 = 1 Btrieve aparta o precoloca páginas para el archivo.
- Bit 3 = 1 Btrieve comprime los datos en el archivo.
- Bit 4 = 1 Btrieve crea el archivo como de tipo *sólo llave*.
- Bit 6 = 1 Btrieve mantiene un 10% de separación.
- Bit 7 = 1 Btrieve mantiene un 20% de separación.
- Bit 6 = 1 y Bit 7 = 1 Btrieve mantiene un 30% de separación.

Las especificaciones de llave aparecen inmediatamente después de las especificaciones de archivo y se repiten para cada segmento de llave. Btrieve coloca las banderas de llave como sigue:

- Bit 0 = 1 La llave acepta duplicados.
- Bit 1 = 1 La llave es modificable.
- Bit 2 = 0 y Bit 8 = 0 La llave es de tipo cadena estándar.
- Bit 2 = 1 y Bit 8 = 0 La llave es de tipo binario estándar.
- Bit 3 = 1 La llave tiene un valor nulo.
- Bit 4 = 1 La llave es segmentada.
- Bit 5 = 1 La llave es almacenada en base a la secuencia alternativa de ordenamiento.
- Bit 6 = 1 La llave es almacenada en orden descendente.
- Bit 7 = 1 La llave es un índice suplementario.

ESTADO (15).
(Continuación.)

Bit 8 = 1 La llave no es de tipo estándar.

Bit 9 = 1 La llave es manual.

Vea la tabla de la operación *Crear* en este capítulo, para conocer valores decimales de las banderas.

Si se regresa una secuencia alternativa de ordenamiento para cualquier llave o segmento de llave en el archivo, Btrieve regresa también, la definición de la secuencia, inmediatamente después del último bloque de especificación de llave.

Prerequisitos:

- El archivo debe estar abierto.

Procedimiento:

1. Cambie el código de operación con el número 15.
2. Pase el bloque de posición asociado al archivo.
3. Pase un *buffer de datos* lo suficientemente grande para contener las estadísticas del archivo.
4. Indique correctamente el *largo del buffer de datos*.
5. Asegurar que el *buffer de llave* permita al menos 64 caracteres.

Resultado:

Si esta operación fue realizada sin ningún percance, sucede lo siguiente:

- Btrieve regresa en el *buffer de datos*, las características del archivo e índices.

Si esta operación fue realizada con algún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

3 Archivo no abierto.

22 El *buffer de datos* es muy pequeño.

Posicionamiento:

Esta operación no tiene ningún efecto sobre la posición del archivo.

FIJAR CLAVE (29).

Esta operación asigna una clave de acceso a un archivo. Para intentar abrir el archivo el usuario debe tener esta clave de lo contrario no podrá escribir en el archivo por medio de Btrieve y más aún no se podrá ni siquiera leer, si así se indicó en esta operación. Si un archivo tiene especificada una clave de acceso, los usuarios o aplicaciones deben especificarla cada vez que requieran abrir el archivo hasta que la clave para el archivo se elimine con la operación *eliminar clave* op. No. 30. Además de indicar una clave de acceso para el archivo, también se definen los privilegios que cualquier persona tendrá sobre el archivo si no conoce la clave.

Cuando se asigna una clave, también se puede indicar que encripte la información dentro del archivo. Si se desea encriptar el archivo Btrieve realiza este proceso durante esta operación. Dependiendo el tamaño del archivo es el tiempo que tarda en finalizar.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Enviar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Resultado						

Prerequisitos:

- El archivo debe estar abierto.
- El archivo no debe tener ninguna transacción activa.
- La clave de acceso no se puede fijar si el archivo ya tiene una. (Eliminar antes)

Procedimiento:

1. Cambie el código de operación con el número 29.
2. Pase el bloque de posición asociado al archivo que se desea proteger.
3. Colocar la clave de acceso en el *buffer de datos* y también en el *buffer de llave* e indique el largo del buffer correcto. Btrieve requiere la clave en los dos parámetros para validar si son iguales, con esto se evitan errores en la especificación. La clave debe constar de hasta 8 caracteres finalizando con el cero binario.
4. Coloque en el *número de llave* el tipo de restricción en el acceso que se requiere y si la encriptación se llevará a cabo.

La siguiente tabla enlista los valores permitidos.

FIJAR CLAVE (29).
(Continuación.)

VALOR	DESCRIPCIÓN
0	Requiere de la clave para cualquier acceso. (Sin encriptación de los datos.)
1	Permite abrir el archivo sólo para leer, sin la clave. (Sin encriptación de los datos.)
2	Requiere de la clave para cualquier acceso. (Encriptando los datos.)
3	Permite abrir el archivo sólo para leer, sin la clave. (Encriptando los datos.)

Resultado:

Si esta operación fue realizada sin ningún percance, sucede lo siguiente:

- Btrieve prohíbe el acceso al archivo a partir de ese momento y encripta la información si se indicó de esa manera.
- La clave se requiere hasta que ésta sea eliminada.

Si la operación fue realizada con algún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 41 Operación no permitida.
- 50 El archivo ya tiene clave de acceso.
- 51 Clave inválida

Posicionamiento:

Esta operación no tiene ningún efecto sobre la posición del archivo.

FIJAR DIRECTORIO (17).

Esta operación cambia el directorio por omisión, es equivalente a la instrucción CD de DOS.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Enviar	Ⓜ				Ⓜ	
Resultado						

Prerequisitos:

- El drive lógico y el directorio que se indiquen deben ser válidos.

Procedimiento:

1. Cambie el código de operación con el número 1.
2. Coloque el drive lógico y el directorio sobre el cual desea trabajar por omisión terminando con un cero binario, en el *buffer de llave*. Si el drive no es indicado, se utilizará el drive por omisión actual. Si la especificación del directorio no se indica desde el directorio raíz, el dato se toma como un subdirectorio del directorio actual.

Resultado:

Si ésta operación fue realizada sin ningún percance, sucede lo siguiente:

- Btrieve realiza el cambio del directorio por omisión al directorio indicado.

Si esta operación fue realizada con algún percance, Btrieve no realiza la operación y regresa algún código diferente de cero.

Posicionamiento:

Esta operación no tiene ningún efecto sobre la posición del archivo.

FIN DE TRANSACCIÓN (19).

Esta operación completa una transacción y realiza los cambios apropiados en los archivos de datos, incluso libera los archivos bloqueados. (En el siguiente capítulo se discutirá ampliamente del uso de esta operación)

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Enviar	19					
Resultado						

Prerequisitos:

- Antes de ejecutar esta operación la aplicación debe tener una transacción activa es decir, que se utilizó la operación 19 *inicio de transacción*.

Procedimiento:

Cambie el código de operación con el número 20. Btrieve ignora todos los demás parámetros.

Resultado:

Si esta operación fue realizada sin ningún percance, todas las operaciones delimitadas por la transacción son grabadas en los archivos de datos.

Si esta operación fue realizada con algún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 38 Error en el archivo de control de transacción.
- 74 Transacción abortada automáticamente.

Posicionamiento:

Esta operación no tiene ningún efecto sobre la posición de registros.

INICIO DE TRANSACCIÓN (19).

La operación de inicio de transacción, define el principio de una transacción. Las transacciones son útiles cuando se necesitan realizar múltiples operaciones como por ejemplo una transferencia de fondos, si la transferencia no es completada en su totalidad, la base de datos quedaría inconsistente.

Encerrando un conjunto de operaciones entre inicio y fin de transacción, es posible asegurarse que Btrieve no grabe cualquiera de las operaciones que intervienen, hasta que se especifique la operación de fin de transacción.

En la modalidad *Basado en Servidor*, el número de archivos que pueden ser accedidos en una transacción, se configura dentro del rango de 0 a 18. En la modalidad de *Basada en Cliente*, este número se puede cambiar con la opción /N como se ha descrito anteriormente.

Los archivos que son operados con abrir, consultar, o escalón, dentro de una transacción, serán bloqueados hasta que se le de fin o se aborte la transacción

En el siguiente capítulo se discutirá ampliamente del uso de esta operación.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Enviar	19					
Resultado						

Prerequisitos:

- La aplicación debe terminar o abortar cualquier transacción anterior antes de que se utilice esta operación.
- Para aplicaciones que corren en la modalidad de *Basado en Cliente*, se debió especificar un archivo de control de transacción válido, al utilizar la opción /T cuando se puso en marcha Btrieve.
- Para la modalidad de *Basado en Servidor*; en el servidor que está utilizando Btrieve, debe estar configurada la opción de número de transacciones mayor a 0.

Procedimiento:

Cambie el código de operación con el número 19. Btrieve ignora todos los demás parámetros.

INICIO DE TRANSACCIÓN (19).
(continuación)

Resultado:

Si esta operación fue realizada sin ningún percance, Btrieve regresa el código de status con un valor de cero.

Si esta operación fue realizada con algún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

36 Error de transacción.

37 Una transacción está activa.

Posicionamiento:

Esta operación no tiene ningún efecto sobre la posición de registros.

INSERTAR (2).

Esta operación inserta un registro específico dentro del archivo indicado. Btrieve ajusta automáticamente todos los índices que intervienen, para reflejar la adición del nuevo registro.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

- El archivo debe estar abierto.
- El registro a insertar debe ser del tamaño apropiado, y los valores de las llaves deben considerar las características de los índices (tipo de llave, si acepta valores duplicados, etc.).

Procedimiento:

1. Cambie el código de operación con el número 2.
2. Pase el bloque de posición asociado al archivo en el que desea realizar la inserción.
3. Coloque el nuevo registro en el *buffer de datos*.
4. Especifique el *largo del buffer*. Este valor debe ser al menos el tamaño de la porción fija del registro.
5. Indique el *número de llave* que Btrieve utilizará para establecer su Posicionamiento.

Resultado:

Si esta operación fue realizada sin ningún percance, sucede lo siguiente:

- Btrieve inserta el registro dentro del archivo indicado y ajusta automáticamente todos los índices que intervienen, para reflejar la adición del nuevo registro.

Si esta operación fue realizada con algún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 2 Error de E/S.
- 3 Archivo no abierto.
- 5 Valor de llave duplicado.
- 14 Error al abrir el archivo de pre-imagen.
- 15 Error de E/S en el archivo de pre-imagen.
- 18 Disco lleno.

INSERTAR (2).
(Continuación.)

21 *Buffer de llave* muy corto.

22 El *buffer de datos* es muy pequeño.

Posicionamiento:

Esta operación establece completamente la posición lógica y física del registro, tomando para la posición lógica la llave indicada en el parámetro de *número de llave*.

OBTENER POSICIÓN (17).

Esta operación regresa la posición física del registro actual en una dirección en 4-bytes. Esta operación no se realiza si no se ha establecido una posición física. Mientras la aplicación tenga la dirección de cualquier registro, es factible obtener un registro con la operación 23 *Consultar directamente*.

Btrieve no realiza ninguna E/S sobre el disco al utilizar esta operación.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Enviar						
Resultado						

Prerequisitos:

- El archivo se debe encontrar abierto.
- La aplicación debe tener establecida una posición física.

Procedimiento:

1. Cambie el código de operación con el número 22.
2. Pase el bloque de posición asociado al archivo sobre el que desea obtener la posición.
3. Pase el *buffer de datos* con el espacio suficiente para contener la dirección de 4 bytes.
4. Cambie el *largo del buffer de datos* con un valor al menos igual a 4.

Resultado:

Si esta operación fue realizada sin ningún percance, sucede lo siguiente:

- Btrieve regresa la dirección del registro actual dentro del *buffer de datos*. La posición contenida en 4 bytes es un valor binario donde la primer palabra es la más significativa, éste valor indica el offset del registro dentro del archivo.

Si esta operación fue realizada con algún percance, Btrieve puede regresar uno de los siguientes códigos diferentes de cero:

- 8 Posicionamiento inválido.

Posicionamiento:

Esta operación no tiene ningún efecto sobre la posición del archivo.

RESTAURAR (30).

Libera todos los recursos ocupados por una estación de trabajo. Esta operación aborta cualquier transacción activa en la estación, desbloquea todos los registros y cierra todos los archivos.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Enviar	[E]				[E]	[E]
Resultado						

Prerequisitos:

Se debe poner en marcha a Btrieve antes de utilizar esta operación.

Procedimiento:

1. Cambie el código de operación con el número 28.
2. Si la aplicación está utilizando la modalidad de *Basada en cliente*, o si se desea restaurar los recursos de una sola estación, cambie el parámetro de *número de llave* a un valor positivo o cero.
Si la aplicación desea restaurar los recursos de otra estación de trabajo, en la modalidad *Basada en Servidor*, cambie el valor del parámetro *número de llave* igual con -1.
3. Si va a restaurar otra estación de trabajo, coloque el *número de conexión* de la estación que se desea restaurar en los primeros dos bytes del *buffer de llave*, de otra forma, este parámetro no es requerido.

Resultado:

Si la operación se realizó sin ningún percance, Btrieve realiza las siguientes acciones en el orden que se especifica a continuación:

1. Aborta cualquier transacción activa en la estación de trabajo especificada.
2. Libera todos los bloqueos de la estación de trabajo.
3. Cierra todos los archivos de la estación de trabajo.

Si la operación no se realizó satisfactoriamente, Btrieve regresa algún código de error.

Posicionamiento:

Esta operación destruye las posiciones físicas y lógicas puesto que se cierran los archivos.

VERSIÓN (26).

Esta operación regresa la versión del Btrieve o Brequest y el número de revisión.

Parámetros:

	Operación	Block de Posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Enviar	26			5		
Resultado			1	1		

Prerequisitos:

Se debe poner en marcha a Btrieve o Brequest antes de utilizar esta operación.

Procedimiento:

1. Cambie el código de operación con el número 26.
2. Pase un *buffer de datos* con un tamaño de al menos 5 bytes.
3. Cambie el *largo del buffer de datos* con un valor de al menos 5.

Resultado:

Si la operación se realizó sin ningún percance, Btrieve regresa la información en el *Buffer de datos* en el formato siguiente:

TAMAÑO	DESCRIPCIÓN
2	Entero que contiene el número de la versión.
2	Entero que contiene el número de revisión
1	Carácter que contiene la modalidad de Btrieve. "N" para <i>basado en servidor y basado en cliente</i> , "P" para OS/2, y "W" para MS-Windows.

Si la operación no se realizó satisfactoriamente, Btrieve regresa algún código de error.

Posicionamiento:

Esta operación no tiene ningún efecto en la posición dentro de ningún archivo.

TABLA DE OPERACIONES DE BTRIEVE ORDENADA POR CÓDIGO

OPERACIÓN	CÓDIGO	DESCRIPCIÓN
Abrir	0	Habilita el acceso a un archivo
Cerrar	1	Deshabilita el acceso a un archivo
Insertar	2	Inserta un nuevo registro dentro de un archivo
Actualizar	3	Actualiza un registro en el archivo
Borrar	4	Elimina el registro actual del archivo
Consulta Igual a	5	Regresa el registro cuyo valor de llave sea igual al valor de llave especificado
Consulta el Siguiente	6	Regresa el registro que sigue al actual en el orden del índice especificado.
Consulta el Anterior	7	Regresa el registro anterior al actual en el orden del índice especificado.
Consulta Mayor a	8	Regresa el registro cuyo valor de llave sea mayor al valor especificado
Consulta Mayor o igual a	9	Regresa el registro cuyo valor de llave sea mayor o igual al valor especificado
Consulta Menor a	10	Regresa el registro cuyo valor de llave sea menor al valor especificado
Consulta Menor o igual a	11	Regresa el registro cuyo valor de llave sea menor o igual al valor especificado
Consulta el Primero	12	Regresa el primer registro que contenga el índice
Consulta el Último	13	Regresa el último registro que contenga el índice
Crear	14	Crea un archivo de Btrieve con las características que se le especifiquen.
Estado	15	Muestra las características de un archivo y sus índices, así como el número de registros que contiene
Fijar Directorio	17	Cambia el Directorio actual
Consulta Directorio	18	Obtiene el directorio actual
Inicio de Transacción	19	Marca el inicio de una serie fija de operaciones lógicas
Fin de Transacción	20	Marca el final de una serie fija de operaciones lógicas
Abortar Transacción	21	Elimina las operaciones realizadas durante una transacción en proceso.
Obtener Posición	22	Regresa la posición física del registro actual
Consulta Directamente	23	Regresa el registro de una posición física especificada
Escalón Siguiente	24	Regresa el registro que sigue del actual en el orden físico del archivo
Alto	25	Desactiva Btrieve de la estación de Trabajo
Versión	26	Regresa la versión de Btrieve
Desbloquear	27	Desbloquea un registro o un conjunto de registros
Restaurar	28	Libera todos los recursos utilizados por una estación de trabajo
Fijar Clave	29	Asigna una clave de acceso a un archivo
Eliminar Clave	30	Elimina la clave de acceso de un archivo
Crear índice suplementario	31	Crea un índice suplementario
Borrar índice suplementario	32	Elimina un índice suplementario
Escalón inicial	33	Regresa el primer registro físico del archivo

Continuación...

OPERACIÓN	CÓDIGO	DESCRIPCIÓN
Escalón final	34	Regresa el último registro físico del archivo
Escalón anterior	35	Regresa el registro físico anterior al registro actual del archivo
Consultar el Siguiete extendido	36	Regresa el registro que sigue del actual en el orden de la tabla de índice, en un archivo que se encuentra separado.
Consultar Anterior extendido	37	Regresa el registro anterior del actual en el orden de la tabla de índice, en un archivo que se encuentra separado.
Consultar sólo Llave	+50	Detecta la presencia de un valor de llave en el archivo, sólo busca en el índice y no extrae el registro de datos.
Bloqueos	+100	Bloqueo con espera de un sólo registro
	+200	Bloqueo sin espera de un sólo registro
	+300	Bloqueo con espera de múltiples registros
	+400	Bloqueo sin espera de múltiples registros

Tabla 3.04
Operaciones ordenadas por código.

TABLA DE OPERACIONES DE BTRIEVE ALFABÉTICO POR OPERACIÓN

OPERACIÓN	CÓDIGO	DESCRIPCIÓN
Abortar Transacción	21	Elimina las operaciones realizadas durante una transacción en proceso.
Abrir	0	Habilita el acceso a un archivo
Actualizar	3	Actualiza un registro en el archivo
Alto	25	Desactiva Btrieve de la estación de Trabajo
Borrar	4	Elimina el registro actual del archivo
Borrar índice suplementario	32	Elimina un índice suplementario
Cerrar	1	Deshabilita el acceso a un archivo
Consulta Directamente	23	Regresa el registro de una posición física especificada
Consulta Directorio	18	Obtiene el directorio actual
Consulta el Anterior	7	Regresa el registro anterior al actual en el orden del índice especificado.
Consulta el Primero	12	Regresa el primer registro que contenga el índice
Consulta el Siguiente	6	Regresa el registro que sigue al actual en el orden del índice especificado.
Consulta el Último	13	Regresa el último registro que contenga el índice
Consulta Igual a	5	Regresa el registro cuyo valor de llave sea igual al valor de llave especificado
Consulta Mayor a	8	Regresa el registro cuyo valor de llave sea mayor al valor especificado
Consulta Mayor o igual a	9	Regresa el registro cuyo valor de llave sea mayor o igual al valor especificado
Consulta Menor a	10	Regresa el registro cuyo valor de llave sea menor al valor especificado
Consulta Menor o igual a	11	Regresa el registro cuyo valor de llave sea menor o igual al valor especificado
Crear	14	Crea un archivo de Btrieve con las características que se le especificuen.
Crear índice suplementario	31	Crea un índice suplementario
Desbloquear	27	Desbloquea un registro o un conjunto de registros
Eliminar Clave	30	Elimina la clave de acceso de un archivo
Escalón anterior	35	Regresa el registro físico anterior al registro actual del archivo
Escalón Final	34	Regresa el último registro físico del archivo
Escalón Inicial	33	Regresa el primer registro físico del archivo
Escalón Siguiente	24	Regresa el registro que sigue del actual en el orden físico del archivo

Continuación...

OPERACIÓN	CÓDI GO	DESCRIPCIÓN
Estado	15	Muestra las características de un archivo y sus índices, así como el número de registros que contiene
Fijar Clave	29	Asigna una clave de acceso a un archivo
Fijar Directorio	17	Cambia el Directorio actual
Fin de Transacción	20	Marca el final de una serie fija de operaciones lógicas
Inicio de Transacción	19	Marca el inicio de una serie fija de operaciones lógicas
Insertar	2	Inserta un nuevo registro dentro de un archivo
Obtener Posición	22	Regresa la posición física del registro actual
Restaurar	28	Libera todos los recursos utilizados por una estación de trabajo
Versión	26	Regresa la versión de Btrieve
Consultar solo Llave	+50	Detecta la presencia de un valor de llave en el archivo, sólo busca en el índice y no extrae el registro de datos.
Bloqueos	+100	Bloqueo con espera de un sólo registro
	+200	Bloqueo sin espera de un sólo registro
	+300	Bloqueo con espera de múltiples registros
	+400	Bloqueo sin espera de múltiples registros

Tabla 3.05
Operaciones ordenadas por nombre.

TABLA DE LOS PARAMETROS REQUERIDOS POR OPERACION.

Enviar parámetro
Resultado

<input type="checkbox"/>
<input type="checkbox"/>

OPERACIÓN	Código	Opera- ción	Blok de posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Abortar Transacción	21	<input type="checkbox"/>					
Abrir	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Actualizar	3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Alto	25	<input type="checkbox"/>					
Borrar	4	<input type="checkbox"/>	<input type="checkbox"/>				
Borrar índice suplementario	32	<input type="checkbox"/>	<input type="checkbox"/>				
Cerrar	1	<input type="checkbox"/>	<input type="checkbox"/>				
Consulta Directamente	23	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Consulta Directorio	18	<input type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>
Consulta el Anterior	7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Consulta el Primero	12	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Consulta el Siguiente	6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Consulta el Último	13	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Consulta Igual a	5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Consulta Mayor a	8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Consulta Mayor o igual a	9	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Consulta Menor a	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Consulta Menor o igual a	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Crear	14	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Continuación ...

OPERACIÓN	Código	Operación	Block de posición	Buffer de datos	Largo del buffer de datos	Buffer de llave	Número de llave
Crear índice suplementario	31	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Desbloquear	27	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
Eliminar Clave	30	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
Escalón anterior	35	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
Escalón Final	34	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Escalón Inicial	33	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Escalón Siguiente	24	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Estado	15	<input checked="" type="checkbox"/>					
Fijar Clave	29	<input checked="" type="checkbox"/>					
Fijar Directorio	17	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>	
Fin de Transacción	20	<input checked="" type="checkbox"/>					
Inicio de Transacción	19	<input checked="" type="checkbox"/>					
Insertar	2	<input checked="" type="checkbox"/>					
Obtener Posición	22	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Restaurar	28	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Versión	26	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Tabla 3.06
Parámetros requeridos
por operación.

CLASIFICACIÓN DE LAS OPERACIONES:

Las operaciones de Btrieve las podemos clasificar según su función o uso en seis tipos de operaciones: operaciones de archivo e índices, de control de procesos y seguridad, de consulta, de mantenimiento de registros, informativas y misceláneas.

-OPERACIONES DE ARCHIVO E ÍNDICES

Abrir	0
Cerrar	1
Crear	14
Borrar índice suplementario	32
Crear índice suplementario	31

- OPERACIONES DE MANTENIMIENTO DE REGISTROS.

Actualizar	3
Borrar	4
Insertar	2

- OPERACIONES DE CONSULTA.

Consultas directas y por índice. (Establecen el posicionamiento físico y lógico)

Consulta Directamente	23
Consulta el Anterior	7
Consulta el Primero	12
Consulta el Siguiente	6
Consulta el Último	13
Consulta Igual a	5
Consulta Mayor a	8
Consulta Mayor o igual a	9
Consulta Menor a	10
Consulta Menor o igual a	11

Consultas secuenciales de la totalidad del archivo sin orden.

(Establecen el posicionamiento físico)

Escalón anterior	35
Escalón Final	34
Escalón Inicial	33
Escalón Siguiente	24

- OPERACIONES PARA EL CONTROL DE PROCESOS Y SEGURIDAD.

Abortar Transacción	21
Desbloquear	27
Eliminar Clave	30
Fijar Clave	29
Fin de Transacción	20
Inicio de Transacción	19
Bloqueos	+100, +200, +300, +400

- OPERACIONES INFORMATIVAS.

Estado	15
Obtener posición	22
Versión	26

- OPERACIONES VARIAS.

Alto	25
Fijar directorio	17
Restaurar	28
Consultar sólo llave	+50

3.03 DESCRIPCIÓN DEL PROGRAMA DE UTILERIA B.EXE.

Esta utilería se incluye en el paquete de Btrieve en la modalidad de *basado en cliente*, pero funciona perfectamente en la modalidad *basado en servidor*.

El programa B.EXE permite ejecutar cualquier operación de Btrieve de forma individual e interactiva. Este programa realiza las operaciones basándose en los valores que se especifiquen en cada uno de los parámetros que se solicitan. Este programa es útil para aprender como opera Btrieve, probar la lógica de un programa, o depurar un programa.

Para ejecutar esta utilería, indique en el sistema operativo lo siguiente:

B.1

Quando se ejecuta la utilería, aparece en la pantalla un grupo de preguntas que corresponden a los parámetros que requiere Btrieve, también una lista de todas las operaciones con su número y nombre (en inglés), y un lugar donde informa el código de error actual.

Para realizar cualquier operación, se deben inicializar o llenar los parámetros que utiliza la función en particular. Al inicio de este capítulo se presentan los parámetros que se necesitan para cada operación.

Después de llenar los parámetros correspondientes para la operación que se desea ejecutar, sólo resta oprimir la tecla [F1], para que la utilería realice la función y regrese la información y el código de error como respuesta, quedando listo para otra operación. Se pueden solicitar tantas operaciones como se desee. Para finalizar con la utilería se deben cerrar los archivos que se abrieron, y oprimir la tecla [Esc] (Escape).

A continuación se presentan dos tablas que enlistan los parámetros que se solicitan y algunas consideraciones para éstos, y otra tabla donde se enlistan las teclas que se pueden utilizar para facilitar el uso de este programa.

Tabla de preguntas de la utilidad B.EXE

PREGUNTA	DESCRIPCIÓN Y DETALLES
Function	Indique el No. de operación que se desea realizar. La lista aparece en la parte inferior de la pantalla.
Key Path	Indique el número de llave.
Position Block	Indique el número de identificación del archivo que se desea acceder. La utilidad asigna un número consecutivo en el momento de la apertura. Se pueden abrir con esta utilidad hasta 10 archivos, comienza con el número de identificación del 0 hasta el 9.
Status Code (es un dato)	En este caso no se puede escribir nada, la utilidad regresa el No. de error que se presentó después de ejecutar la operación, de no existir alguna anomalía se regresa un cero.
Data Buffer Length	Especifique el valor correcto para el largo del buffer de datos.
Data Buffer	Indique el contenido del buffer de datos requerido por la operación, en este campo sólo se permiten caracteres imprimibles, en las operaciones de consulta sólo estos caracteres se presentan.
Key Buffer	Coloque el contenido del buffer de llave, requerido por la operación, en este campo sólo se permiten caracteres imprimibles, en las operaciones de consulta sólo se presentan estos caracteres.

Tabla con las teclas de función para esta utilidad.

TECLA	FUNCIÓN
[Esc]	Salida de la utilidad.
[F1]	Ejecuta la operación indicada con los valores capturados.
[Home]	Se posiciona el cursor sobre la primera pregunta.
[End]	Se posiciona el cursor sobre la última pregunta.
[↑]	Se posiciona el cursor sobre la pregunta anterior.
[↓] [Tab]	Se posiciona el cursor sobre la siguiente pregunta.
[←]	Se posiciona el cursor sobre el caracter anterior.
[→]	Se posiciona el cursor sobre el siguiente caracter.
[Ctrl-Home]	Se posiciona el cursor sobre el inicio del campo.
[Ctrl-End]	Se posiciona el cursor sobre el final del campo.
[Delete] ó	Borra un caracter.
[BackSpace]	
[Ins]	Cambia al modo de inserción de caracteres.

CAPÍTULO IV

HERRAMIENTAS PARA EL CONTROL DE PROCESOS Y USO EN RED

4.01 RECUPERACIÓN AUTOMÁTICA DE INFORMACIÓN Y CONTROL DE PROCESOS POR MEDIO DE TRANSACCIONES.

Btrieve provee de tres métodos para implementar la recuperación automática de información.

- Pre-imagen
- Pre-imagen directa
- Control de transacción

- **PRE-IMAGEN.**

Al indicar esta cualidad (ver capítulo II), Btrieve crea un archivo que es como una imagen del archivo real, este archivo siempre tiene el mismo nombre que el archivo original, la diferencia radica en la extensión del archivo de pre-imagen, la cual siempre es .PRE. Por ejemplo C:\DATOS\ALUMNOS.BTR tendrá como imagen el archivo C:\DATOS\ALUMNOS.PRE. En este método de seguridad, el archivo de imagen se encuentra dentro del mismo directorio que el archivo original.

En forma concreta, este método utiliza el archivo de imagen para actualizar en él las modificaciones que se realicen en el archivo original, estas modificaciones son guardadas en el archivo de imagen antes que sean colocadas en el archivo original.

Si existiera alguna falla en el sistema o de corriente en las computadoras, es posible que la información dentro del archivo original quede corrupta. En el momento en que el archivo dañado sea abierto por primera vez, después de una terminación anormal del sistema, automáticamente se restaurará el problema cuando Btrieve verifique el contenido del archivo de imagen.

- **PRE-IMAGEN DIRECTA.**

Normalmente, Btrieve crea el archivo de pre-imagen en el mismo disco, cuando se utiliza este método, Btrieve genera el archivo pre-imagen en otro disco. Este disco debe contener los mismos directorios que los asignados para los archivos originales por ejemplo en el caso del archivo localizado en C:\DATOS\ALUMNOS.BTR su archivo de imagen se encontrará en F:\DATOS\ALUMNOS.PRE si la unidad de disco que se definió fue la F:.

- CONTROL DE TRANSACCIÓN.

Es posible obtener un nivel más alto de recuperación de información que utilizando los métodos de pre-imagen, por medio de la definición de transacciones lógicas dentro de la algún programa de aplicación.

Este método de recuperación de información es muy útil, cuando se desea mucha seguridad al realizar procesos de escritura en archivos, cuando es preferible que el proceso no se concluya a que se concluya parcialmente por algún error o bien cuando el proceso sea susceptible de ser abortado.

Para comenzar una transacción lógica primero se define el inicio de la transacción (Oper. 19), esto quiere decir que todas las instrucciones a partir de este inicio formarán parte de la transacción hasta que se encuentre la operación de fin de transacción (Oper. 20). Las instrucciones que se encuentran dentro de una transacción son operaciones de actualizaciones en archivos.

Si el proceso que se realiza dentro de la transacción es interrumpido por un error en la aplicación o en la computadora (falla de corriente por ejemplo), las operaciones que se alcanzaron a terminar serán borradas automáticamente en el momento de abrir el archivo después de la falla, el borrar las operaciones es con el fin de que, se pueda repetir el proceso nuevamente desde el principio.

Si el programa de aplicación encuentra la operación de fin de transacción, entonces la transacción se considera como concluida. Es posible solicitar con una operación si la transacción se aborta (Oper. 21), esto ocurre solamente si la aplicación lo decide, es decir, se dió una condición por la cual el proceso no debe o no puede continuar. Como por ejemplo que el usuario no quiera grabar los cambios realizados.

Dependiendo la modalidad en que se esté ocupando Btrieve, se especifica de una u otra forma que se desea utilizar este método de recuperación de información.

4.02 FACTIBILIDAD DEL MULTIACCESO Y USO DE BLOQUEOS.

- INTRODUCCIÓN AL MULTIACCESO.

El multiacceso a un archivo se efectúa cuando dos o más computadoras tienen abierto un archivo de un disco común. Esto sucede en caso de contar con una red de computadoras donde una o más de éstas tienen la función de servidoras de archivo (File server), estas computadoras ponen a la disposición de las demás la información contenida en su disco duro.

Btrieve provee algunos métodos para resolver conflictos que pueden ocurrir cuando dos o más aplicaciones desean acceder los mismos registros de un archivo Btrieve. Estos métodos son:

- Bloqueo por transacción.
- Concurrencia pasiva.
- Detección de bloqueos muertos.
- Bloqueo de registros.

Es posible utilizar cualquiera de estos métodos de manera individual o combinada dentro de una sola aplicación.

● BLOQUEO POR TRANSACCIÓN.

Cuando se inicia una transacción dentro de un archivo, no es posible que otra aplicación inicie otra con este archivo, es decir que el archivo se encuentra momentáneamente bloqueado para intentar iniciar una transacción con él. Existen dos tipos de bloqueos; los bloqueos con espera y los bloqueos sin espera.

Los bloqueos con espera se refieren a que, si una aplicación inició una transacción de bloqueo con espera, en el momento que otra aplicación intente iniciar una transacción con el mismo archivo, la ejecución de la aplicación se detiene hasta que la transacción que está ejecutando se concluya.

Los bloqueos sin espera se refieren a que si una aplicación inició una transacción con bloqueo sin espera, en el momento que otra aplicación intente iniciar una transacción con el mismo archivo, la aplicación recibirá inmediatamente un código indicando que el archivo está ocupado.

- **CONCURRENCIA PASIVA.**

La concurrencia pasiva es el nombre que se le da a una forma de trabajo o método de trabajo que admite el multiacceso a uno o más archivos. Este método no requiere utilizar transacciones, ni bloqueos para prevenir conflictos, simplemente verifica en el momento de escribir si no hay algún impedimento para hacerlo.

Si el registro leído por alguna estación de trabajo fue modificado por otra, entre el momento en que fue leído y en el momento en que se intenta actualizarlo, Btrieve regresa un código de advertencia, para que la aplicación decida si abortar la inserción o volver a leer el registro para actualizarlo nuevamente.

- **DETECCIÓN DE BLOQUEOS MUERTOS.**

La detección de bloqueos muertos, se refiere a una situación específica en la que una estación de trabajo espera a que se libere uno o más de los registros bloqueados por otra estación de trabajo y esta segunda espera que se liberen uno o más registros de la primer estación, en ese momento Btrieve advierte el problema. Si no fuera advertido las dos computadoras quedarían en espera indefinida.

- **BLOQUEO DE REGISTROS.**

Siempre que una aplicación consulta un registro de un archivo determinado, es posible especificar si se desea bloquear ese registro. Otras estaciones de trabajo pueden leer registros bloqueados dentro del mismo archivo, pero no pueden actualizar o volver a bloquear estos registros hasta que éstos sean liberados por la estación de trabajo que solicitó el bloqueo inicialmente.

- USO DE BLOQUEOS.

Los bloqueos permiten controlar el acceso a registros o archivos. Btrieve no tiene un número específico de operación para realizar un bloqueo. Lo que se debe realizar para solicitar el bloqueo de un registro es sumar una constante a determinada operación para que al ejecutarse se realice el bloqueo, o bien, como ya se ha indicado antes, las operaciones que se encuentran dentro de una transacción realizan el bloqueo automáticamente. Btrieve utiliza exclusivamente bloqueos de escritura, esto quiere decir que sólo se le permite modificar registros bloqueados a la estación de trabajo que realizó dichos bloqueos, las demás estaciones sólo tendrán acceso de lectura sobre esos registros. Existen dos diferentes niveles de bloqueo: Bloqueos de Registros y Bloqueos de Archivos.

- **Bloqueos de Registros.**

Se puede solicitar este nivel de bloqueo cuando se extrae un registro a través de las operaciones de consulta o escalón, adicionándole una constante al número de operación correspondiente. El bloqueo se realizará siempre y cuando el registro o archivo a utilizar no se encuentre bloqueado por otra estación de trabajo. Al mantener un registro bloqueado se previene cualquier operación de actualización y/o borrado al registro, solicitada por otra estación de trabajo. Sin embargo es posible consultar dichos registros por operaciones de consulta o escalón que no intenten bloquear el registro. Btrieve tiene una operación explícita para liberar los registros bloqueados (*desbloquear* Oper. 27) . Adicionalmente, algunas operaciones (*restaurar, cerrar, borrar* y en el caso de bloqueos sencillos *actualizar*) realizan implícitamente la liberación del o los registros que la aplicación mantiene bloqueados.

Btrieve reconoce dos tipos de bloqueos de registros:

- **Bloqueo Sencillo de Registro:** Se debe utilizar el bloqueo sencillo de registro cuando se necesita bloquear solamente un registro a la vez. Cuando se utiliza una operación para recuperar un registro solicitando que éste quede bloqueado, automáticamente se libera el registro que se encuentra actualmente bloqueado (si existe alguno) en el archivo.

Las siguientes instrucciones liberan un registro bloqueado de manera sencilla: *Desbloquear* (27), *Borrar* (4), o *Actualizar* (3).

- **Bloqueo Múltiple de Registros:** Se debe utilizar el bloqueo múltiple de registros cuando se necesita bloquear más de un registro en el mismo archivo al mismo tiempo. Cuando se utiliza alguna operación solicitando el bloqueo múltiple de registros, Btrieve bloquea el registro pero no libera el o los registros que ya se encuentran bloqueados en el archivo.

Para liberar algún registro en especial o liberar todos los registros bloqueados en el archivo se puede utilizar cualquiera de las siguientes operaciones: *Desbloquear* que puede liberar un registro específico o todos; *Borrar* para liberar el registro actual.

Sin embargo la operación de *Actualizar* no libera el registro que se actualizó, en este tipo de bloqueo.

No es posible utilizar los dos tipos de bloqueos en el mismo archivo solicitados por la misma aplicación al mismo tiempo. De intentar ésto, Btrieve regresa el código de error número 93 (tipo de bloqueo incompatible) si un tipo de bloqueo se encuentra activo en el momento que se intenta solicitar el otro tipo. Btrieve regresa el código de error número 81 (error de bloqueo) cuando se intenta bloquear más registros que el número máximo configurado.

- Bloqueo de Archivos.

Se puede solicitar el bloqueo de un archivo cuando éste se abre dentro de una transacción o cuando se utiliza un archivo que ya se encuentra abierto dentro de una transacción y las operaciones de consulta y/o escalón, solicitan el bloqueo. El bloqueo del archivo se realiza siempre y cuando el archivo o un registro dentro de él no se encuentren actualmente bloqueados por otra aplicación. El mantener un archivo bloqueado, ocasiona que cualquier otra aplicación no pueda realizar ninguna otra operación de actualización, de borrado o de inserción sobre el archivo. Sin embargo, estas aplicaciones pueden leer cualquier información del archivo, siempre y cuando no se encuentre dentro de una transacción.

NOTA: En la modalidad de *basado en cliente* no es posible abrir un archivo que se encuentra bloqueado. En la modalidad de *basado en servidor*, si es posible.

Btrieve no tiene ninguna operación específica para liberar un archivo. Las operaciones de *fin de transacción*, *abortar transacción*, o *restaurar implícitamente* liberan todos los archivos bloqueados por una transacción.

- TIPOS DE BLOQUEO

Existen dos opciones de bloqueo para los dos niveles descritos anteriormente, los bloqueos con espera y los bloqueos sin espera:

- Bloqueos con espera.

Si se solicita el bloqueo con espera de un archivo o de un registro y éste ya se encuentra bloqueado por otra aplicación, Btrieve no regresa el control a la aplicación hasta que el bloqueo sea liberado. Para especificar el bloqueo con espera, se debe sumar al número de la operación un número 100 para bloqueo sencillo o 300 para un bloqueo múltiple. Por omisión las operaciones de apertura, de consulta y de escalón que se encuentran dentro de una transacción son de la opción con espera.

NOTA: Esta opción de bloqueo puede originar un "bloqueo muerto", esto sucede cuando una aplicación espera que se liberen recursos de otra aplicación que a su vez está esperando que la primera aplicación libere otros recursos, se podría pensar que las dos aplicaciones quedarían indefinidamente en espera, pero no es así, en este caso se regresa un código de error 78.

- **Bloqueos sin espera.**

Si se solicita el bloqueo sin espera de un archivo o de un registro y éste ya se encuentra bloqueado por otra aplicación, inmediatamente Btrieve regresa el código de error 84 (registro bloqueado) o el 85 (archivo bloqueado). Para especificar el bloqueo sin espera se debe sumar 200 para bloqueo sencillo o 400 para bloqueo múltiple. Para especificar el bloqueo sin espera para los movimientos de apertura, de consulta o de escalón que se realizan dentro de una transacción, se debe sumar el número 200 o 400 a la operación de *Inicio de Transacción*.

Condiciones adicionales para las operaciones en que se utilizan los bloqueos.

Cuando se solicita un bloqueo a una operación específica, se contemplan las condiciones de uso de la operación y además se debe considerar lo siguiente:

Parámetros:

Con excepción del código de la operación, los parámetros para la operación en la que se solicita el bloqueo, son exactamente los mismos que los definidos en el Capítulo III

Prerequisitos:

Los prerequisites que se necesitan cuando se solicita una operación con bloqueo, son los mismos que corresponden a la operación, además de los anteriormente descritos en este capítulo.

Procedimiento:

Para especificar un bloqueo, la aplicación debe sumar determinado valor al número de la operación de consulta, escalón o inicio de transacción según corresponda. La siguiente tabla ilustra los valores para cada opción y tipo de bloqueo:

Valor	Descripción
+100	Bloqueo sencillo con espera.
+200	Bloqueo sencillo sin espera.
+300	Bloqueo múltiple con espera.
+400	Bloqueo múltiple sin espera.

Ejemplos:

1. Para especificar la operación de *consulta igual a* (5) solicitando el bloqueo sencillo con espera (+100), en vez de utilizar el número de operación 5 se cambia por el número 105 que es igual a la suma de 5 + 100. Esta operación busca el valor de llave especificado en el archivo correspondiente y si encuentra la llave dentro del archivo regresa en el *buffer de datos* el contenido del registro como es usual pero el registro queda bloqueado.
2. Para indicar la misma operación anterior solicitando el bloqueo múltiple con espera (+300), el código de operación se cambia por el 305 (5 + 300) .
3. Para utilizar la operación de *escalón final* (34) solicitando un bloqueo sencillo sin espera (+200), se cambia el código de operación por 234 (34 + 200).

Resultado:

Si la solicitud de bloqueo se realiza sin ningún percance, el registro o el archivo bloqueado no se podrá bloquear por otra aplicación hasta que se libere explícitamente o implícitamente.

Si la solicitud de bloqueo se realiza con algún percance, Btrieve puede regresar alguno de los siguientes códigos de error diferentes de cero:

- 78 Detección de bloqueos muertos.
- 81 Error de bloqueo.
- 84 Registro bloqueado.
- 85 Archivo bloqueado.
- 93 Tipo de bloqueo incompatible.

Posicionamiento.

La solicitud de bloqueo no tiene ningún efecto adicional sobre la posición actual del archivo, más que el definido por la operación en sí.

- APLICACIÓN DE BLOQUEOS

El ejemplo más común de un bloqueo es el que se implementa en el programa muestra desarrollado en el siguiente capítulo.

El programa muestra permite realizar el mantenimiento a un archivo (altas, bajas, cambios, consultas, catálogo). El proceso de "bajas" y el proceso de "cambios" son los que pueden llegar a tener un conflicto si el archivo es utilizado por dos o más estaciones de trabajo. Por ejemplo:

Si consideramos dos estaciones de trabajo que utilizan el programa, si el archivo cuenta con tres registros de datos, y las estaciones de trabajo realizan las siguientes operaciones "al mismo tiempo":

ESTACIÓN 1	ESTACIÓN 2
Lectura del registro 2 en la opción de cambios.	Lectura del mismo registro en la opción de bajas
Modificación de algunos datos	Baja del registro
Guarda el registro en el archivo	Fin
Error al actualizar. (Concurrencia pasiva)	
"Conflicto"	

Este es el tipo de conflictos más comunes y sencillos de identificar, para poder solucionar este tipo de problemas se requiere el uso de los bloqueos de registros de la siguiente manera:

ESTACIÓN 1	ESTACIÓN 2
Lectura del registro 2 en la opción de cambios, solicitando bloqueo.	Lectura del mismo registro en la opción de bajas, solicitando bloqueo.
Modificación de algunos datos espera o advertencia
Guarda el registro en el archivo espera o advertencia
Fin	Baja del registro
	Fin

La estación que debe esperar, depende del momento exacto en que se solicitó la operación, considerando que una computadora personal no realiza acciones al mismo tiempo, una de las dos operaciones se atiende primero.

4.03 DESCRIPCIÓN DEL PROGRAMA DE UTILERÍA BROLLFWD.EXE

La función de la utilería BROLLFWD.EXE, es recuperar los cambios realizados a un archivo entre el momento del último respaldo y una falla del sistema de computo. Esta utilería puede recuperar la información siempre y cuando se halla completado lo siguiente:

- Especificar la opción /A cuando se pone en marcha Btrieve.
- El archivo BLOG.CFG debe contener la lista de archivos que son susceptibles a ser recuperados por medio de esta utilería.
- Contar con un respaldo de los archivos que se especifican en el archivo BLOG.CFG, antes de iniciar el trabajo con los mismos.

Cuando se solicita la opción /A, Btrieve graba todas las operaciones que modifican el archivo. Si ocurre una falla en la computadora que interrumpa los procesos de actualización, entonces se puede recurrir a esta utilería sobre la copia de respaldo para que los cambios realizados antes de la falla, se reapliquen al respaldo, recuperando así toda la información posible.

- CARACTERÍSTICAS DEL ARCHIVO BLOG.CFG.

Para habilitar el seguimiento de las operaciones a uno o más archivos, primero se debe solicitar la opción /A al poner en marcha a Btrieve. Después se debe especificar un archivo en particular asociado a cada archivo de datos que va a tener el seguimiento. El archivo asociado se puede encontrar en el mismo volumen o en cualquier otro volumen del mismo servidor de archivos.

Para especificar que archivos serán contemplados en el seguimiento, se debe crear un directorio llamado "BLOG" colocado en el directorio raíz del volumen donde se encuentran los archivos de datos. En este directorio se debe crear un archivo de texto (formato ASCII estándar) llamado "BLOG.CFG", dentro de este archivo se debe escribir la lista de archivos que tendrán el seguimiento en ese volumen en particular. Cada línea en el archivo puede contener hasta 256 caracteres, es factible colocar varias especificaciones en una misma línea separándolas por uno o más espacios.

Cada especificación de archivos locales (para Btrieve *Basado en Cliente*), debe conservar el siguiente formato general:

```
\ <directorio> \ <Nombre del archivo Btrieve>  
[=\ <directorio> \ <Nombre del archivo con el seguimiento>]
```

La unidad de disco no es necesaria.

Cada especificación de archivos que se encuentran dentro de una red (para Btrieve *Basado en Servidor*), debe conservar el siguiente formato general:

```
\<directorio> \<Nombre del archivo Btrieve>  
[ = [ \ \<Nombre del servidor> \<volumen>: ]  
\<directorio> \<Nombre del archivo con el seguimiento>]
```

No es necesario especificar el servidor y el volumen donde se encuentra el archivo de datos. Si se desea que el archivo del seguimiento se coloque en un lugar diferente al del archivo de datos, se debe indicar el nombre del servidor y volumen, de lo contrario, si los dos archivos se van a encontrar en el mismo servidor y mismo volumen, éstos datos no son necesarios.

Si el nombre del archivo con el seguimiento no se especifica, en el momento que el archivo de datos se abra, Btrieve asigna el mismo nombre del archivo de datos, sólo que con la extensión ".LOG". Por ejemplo si se desea el seguimiento para el archivo CLIENTES.DAT que se encuentra en el directorio de DATOS, y el archivo asociado no se especificó, el archivo asignado será: \DATOS\CLIENTES.LOG, en este caso el archivo con el seguimiento y el archivo de datos comparten el mismo directorio.

Cada operación que modifique el archivo Btrieve, será grabada al archivo con el seguimiento. Las operaciones subsecuentes serán adicionadas al final del archivo de seguimiento, hasta que este archivo sea borrado. Por esta razón es importante que el administrador de la red realice operaciones periódicas de respaldo para poder borrar estos archivos.

- COMO EJECUTAR LA UTILERÍA.

La utilería BROLLFWR utiliza las funciones de Btrieve para grabar en los archivos con el seguimiento. Se puede recuperar uno o más archivos en el momento de ejecutarse.

Antes de ejecutar la utilería se debe de contar con los respaldos de los archivos a trabajar. Después de aplicar esta utilería a un archivo, éste debe ser respaldado y el archivo con el seguimiento se debe borrar, antes de utilizar el archivo recuperado.

El formato general de la utilería es el siguiente:

```
BROLLFWR {<Archivo Btrieve> | @<lista de archivos> | /A } [/L] [/O:<clave de acceso>]
```

Se debe especificar una de las dos opciones encerradas entre { }.

Las opciones son las siguientes:

- <Archivo Btrieve>** Es el nombre del archivo a recuperar, si se ocupa esta opción el archivo asociado, es tomado del archivo **BLOG.CFG**.
- @ <lista de archivos>** Esta opción se utiliza para recuperar múltiples archivos. Los nombres de los archivos deben estar separados por espacios.
- /A** Si se indica esta opción automáticamente se recuperaran todos los archivos que se encuentran especificados en el archivo **BLOG.CFG**.
- /L** Este parámetro opcional, indica que las operaciones a realizar sólo serán consultadas en pantalla, el proceso no es grabado.
- /O** Este parámetro opcional, permite especificar la clave de acceso del o los archivos. Este parámetro sólo se indica si el archivo a trabajar tiene clave de acceso. Si se indica una lista de archivos, se asume que todos tienen la misma clave de acceso, si la utilería encuentra un archivo que tiene una clave diferente a la especificada, la utilería preguntará por la clave del archivo específico.

ALTAS

La pantalla de altas es la siguiente:

```

PROGRAMA MUESTRA, TRABAJA UTILIZANDO EL MANEJADOR DE BASES DE DATOS BRIEVE
AGENDA DE ALUMNOS DE M.A.C. EN BASIC INTERPRETE
ALTAS

[1] CUENTA      :[8683974-5 ]
[2] NOMBRE     :[ALICIA GARCIA MENDOZA
[3] DIRECCION  :[CONSTANTINO 263-B-6
[4] COLONIA   :[VALLEJO
[5] DEL O CIU :[GUSTAVO A. MADERO ]
[6] ESTADO    :[MEXICO D.F. ]
[7] TELEFONOS :[5-39-82-88 ]
[8] GENERACION:[89-93]
[9] SEMESTRE  :[88]

DESEA DAR DE ALTA ESTOS DATOS (S o N)?
EL NOMBRE Y/O LA CUENTA YA EXISTEN. ALTA INVALIDA...
    
```

En esta pantalla se preguntan los siguientes datos:

Número	Campo	Largo	Restricciones
1	Número de Cuenta	10	no debe ser menor a 9 caracteres ni mayor a 10
2	Nombre	40	no debe ser menor de 11 caracteres ni exceder de 40 caracteres
3	Dirección	40	no debe exceder de 40 caracteres
4	Colonia	35	no debe exceder de 35 caracteres
5	Delegación o Ciudad	25	no debe exceder de 25 caracteres
6	Estado	25	no debe exceder de 25 caracteres
7	Teléfonos	16	no debe exceder de 16 caracteres.
8	Generación	5	debe darse con el formato: AI-AF donde AI = Año Inicial y AF = Año final.
9	Semestre	2	no debe ser menor a 0 ni mayor a 15

Al terminar de teclear el valor de un dato se oprime [Enter] para pasar al siguiente, al concluir con todos aparece la pregunta: "DESEA DAR DE ALTA ESTOS DATOS (S o N)? Presione N si los datos no son correctos, el programa no los almacena y regresa a preguntar nuevos datos, presione S si los datos son correctos, el programa los almacena, les asigna un número consecutivo de REGISTRO y regresa a preguntar nuevos datos. Si se presiona otra tecla por error el programa no realiza ningún movimiento y espera hasta que se teclee S o N.

NOTA: Es muy importante que se respete la longitud de los campos llave, ya que para localizar un registro el programa se basa en la extensión, por ejemplo, si el campo es de 1 a 8 caracteres de largo lo toma como Registro, si es de 9 o 10 caracteres lo toma como Cuenta y si es de más de 10 caracteres lo toma como Nombre.

Para regresar al menú principal, teclee [Enter] cuando se pida la Cuenta o el Nombre sin escribir ningún dato.

BAJAS

La pantalla de Bajas es la siguiente:

```

PROGRAMA MUESTRA, TRABAJA UTILIZANDO EL MANEJADOR DE BASES DE DATOS BTRIEVE
AGENDA DE ALUMNOS DE M.A.C. EN BASIC INTERPRETE
          BAJAS
-----
NOMBRE, CTA o REG:[8683974-5]                REGISTRO: [ 6 ]
[1] CUENTA      :[8683974-5 ]
[2] NOMBRE      :[ALICIA GARCIA MENDOZA      ]
[3] DIRECCION   :[CONSTANTINO 263-B-6        ]
[4] COLONIA     :[VALLEJO                      ]
[5] DEL O CIU   :[GUSTAVO A. MADERO          ]
[6] ESTADO      :[MEXICO D.F.                 ]
[7] TELEFONOS   :[5-39-82-88                 ]
[8] GENERACION  :[89-93]
[9] SEMESTRE    :[ 8 ]

DESEA DARLO DE BAJA (S o N)?
  
```

Al entrar se pide el nombre, número de cuenta o registro a dar de baja, se escribe y se presiona [Enter], si no encuentra al alumno aparece el mensaje "NO SE ENCONTRÓ EL ALUMNO, PULSE [ENTER] P. CONT.", se presiona [Enter] y el programa regresa a preguntar otro Nombre.

Si encuentra al alumno, aparecen los datos de éste y la pregunta "DESEA DARLO DE BAJA (S o N)?", si presiona N el programa respeta los datos y vuelve a preguntar otro Nombre, si se presiona cualquier otra tecla el programa borra los datos del alumno y regresa a preguntar otro Nombre.

Para regresar al menú principal, teclee [Enter] cuando se pida el Nombre sin escribir ningún dato.

CAMBIOS

La pantalla de cambios es la siguiente:

```
PROGRAMA MUESTRA, TRABAJA UTILIZANDO EL MANEJADOR DE BASES DE DATOS BTRIEVE
AGENDA DE ALUMNOS DE M.A.C. EN BASIC INTERPRETE
                                CAMBIOS
-----
NOMBRE, CTA o REG:[ALICIA GARCIA MENDOZA          REGISTRO: [ 6 ]
[1] CUENTA      :[0683974-5 ]
[2] NOMBRE      :[ALICIA GARCIA MENDOZA          ]
[3] DIRECCION   :[CONSTANTINO 263-B-6           ]
[4] COLONIA    :[VALLEJO                        ]
[5] DEL O CIU   :[GUSTAVO A. MADERO             ]
[6] ESTADO     :[MEXICO D.F.                   ]
[7] TELEFONOS  :[5-39-02-00                    ]
[8] GENERACION :[89-93                          ]
[9] SEMESTRE   :[08                             ]

No. DE DATO A MODIFICAR:
```

Al entrar se pide el nombre, número de cuenta ó registro a modificar, se escribe y se presiona [Enter], si no encuentra al alumno aparece el mensaje "NO SE ENCONTRÓ EL ALUMNO, PULSE [ENTER] P. CONT.", se presiona [Enter] y el programa regresa a preguntar otro Nombre.

Si encuentra al alumno, aparecen los datos de éste y la pregunta "ESTA CORRECTA LA INFORMACIÓN (S o N)?", si presiona S el programa vuelve a preguntar otro Nombre, si se presiona cualquier otra tecla aparecerá la pregunta "NO. DE DATO A MODIFICAR", que solicita el número del 1 al 9 ó [Enter], si se presiona un número válido se pregunta por el dato, escríbalo y presione [Enter]; al recibir la sentencia [Enter], haya o no preguntado el dato, el programa pregunta de nuevo si está correcta la información.

Para regresar al menú principal, teclee [Enter] cuando se pida el nombre sin escribir ningún dato.

CONSULTA

La pantalla de consulta es la siguiente:

```
PROGRAMA MUESTRA, TRABAJA UTILIZANDO EL MANEJADOR DE BASES DE DATOS BTRIEVE
AGENDA DE ALUMNOS DE M.A.C. EN BASIC INTERPRETE
CONSULTA
REGISTRO: ( 6 )
]
NOMBRE, CTA o REG:IG
[1] CUENTA :IB683974-5 ]
[2] NOMBRE :IALICIA GARCIA MEMDOZA ]
[3] DIRECCION :ICONSTANTINO 263-B-6 ]
[4] COLONIA :IVALLEJO ]
[5] DEL O CIU :IGUSTAVO A. MADERO ]
[6] ESTADO :IMEXICO D.F. ]
[7] TELEFONOS :IS-39-82-88 ]
[8] GENERACION:(89-93)
[9] SEMESTRE :I 8 ]

DESEA SEGUIR CONSULTANDO (S o N)?
```

Al entrar se pide el nombre, número de cuenta ó registro a consultar, se escribe y se presiona [Enter], si no encuentra al alumno aparece el mensaje "NO SE ENCONTRO EL ALUMNO, PULSE [ENTER] P: CONT.", se presiona [Enter] y el programa regresa a preguntar otro Nombre.

Si encuentra al alumno, aparecen los datos de éste y la pregunta "DESEA SEGUIR CONSULTANDO (S o N)?", si presiona N el programa vuelve al menú principal, si se presiona cualquier otra tecla el programa vuelve a preguntar otro Nombre a consultar.

Para regresar al menú principal, teclee [Enter] cuando se pida el nombre sin escribir ningún dato.

CONSULTA POR SEMESTRE

Las pantallas de consulta por semestre son las siguientes:

PROGRAMA MUESTRA, TRABAJA UTILIZANDO EL MANEJADOR DE BASES DE DATOS BTRIEVE
AGENDA DE ALUMNOS DE M.A.C. EN BASIC INTERPRETE
CONSULTA EN PANTALLA POR SEMESTRE/CUENTA o SEMESTRE/NOMBRE

INDIQUE EL SEMESTRE A CONSULTAR:00
CONSULTA POR 1)CUENTA o 2)NOMBRE (1 o 2)?

PROGRAMA MUESTRA, TRABAJA UTILIZANDO EL MANEJADOR DE BASES DE DATOS BTRIEVE
AGENDA DE ALUMNOS DE M.A.C. EN BASIC INTERPRETE
CONSULTA EN PANTALLA POR SEMESTRE/CUENTA o SEMESTRE/NOMBRE

REG	CUENTA	NOMBRE	TELEFONO	SEM
6	B683974-5	ALICIA GARCIA MENDOZA	5-39-82-88	8
REG	NOMBRE	CUENTA	TELEFONO	SEM
6	ALICIA GARCIA MENDOZA	B683974-5	5-39-82-88	8

PULSE CUALQUIER TECLA PARA CONTINUAR ...

Al entrar se pide el Semestre a Consultar, este puede ser del rango del 1 al 15, si se teclea un valor inválido el programa vuelve a pedir el Semestre, si es válido pregunta por el modo de consulta, si se desea que la consulta esté ordenada por número de cuenta se presiona 1, si se desea que la consulta esté ordenada por nombre se presiona 2, si se presiona cualquier otro caracter, el programa vuelve a preguntar el modo de consulta.

Si se teclea un valor correcto el programa muestra la información ordenada del modo indicado y aparece el mensaje "PULSE CUALQUIER TECLA PARA CONTINUAR", en caso de que toda la información se pueda desplegar en una pantalla, al pulsar cualquier tecla regresa a la pregunta de Indique el semestre, si la información no se puede desplegar en una pantalla, al pulsar cualquier letra aparece la continuación de la lista de consulta y así sucesivamente hasta que regrese a pedir el Semestre.

Para regresar al menú principal, teclee [Enter] cuando se pida el Semestre sin escribir ningún dato.

CATÁLOGOS

La pantalla de catálogos es la siguiente:

```

PROGRAMA MUESTRA, TRABAJA UTILIZANDO EL MANEJADOR DE BASES DE DATOS BTRIEVE
AGENDA DE ALUMNOS DE M.A.C. EN BASIC INTERPRETE
CATALOGOS IMPRESOS
-----
1) POR REGISTRO
2) POR CUENTA
3) POR NOMBRE
4) POR SEMESTRE/CUENTA
5) POR SEMESTRE/NOMBRE

OPCION ==>

ESTA LISTA SU IMPRESORA (S O N)?
    
```

Al entrar se pide el tipo de catálogo que se desea imprimir, estos pueden ser:

Numero	Tipo
1	Ordenado por Registro.
2	Ordenado por Número de Cuenta.
3	Ordenado por Nombre.
4	Ordenado por Semestre/Cuenta.
5	Ordenado por Semestre/Nombre.

Si se tecllea un caracter inválido el programa vuelve a preguntar el tipo de catálogo, si se tecllea un caracter válido (1 - 5) el programa muestra la pregunta "ESTA LISTA SU IMPRESORA (S o N)?", si se presiona N el programa regresa al menú principal sin emitir el reporte, si se presiona S el programa emite el reporte ordenado como le fue indicado y al terminar de mandar la información a la impresora vuelve al menú principal.

5.02 INTERFAZ CON BASIC INTÉRPRETE Y EL PROGRAMA EJEMPLO CODIFICADO.

Este lenguaje de programación, tiene algunas restricciones importantes si lo comparamos con lenguajes más actuales, por ejemplo: el pequeño espacio de memoria de trabajo que es de 64KB, o el límite de las cadenas que es de 255 caracteres. Sin embargo, este lenguaje es muy práctico y común, además de que sirve como base para Basic compilado, donde muchas de las restricciones de Basic intérprete ya no se encuentran, por ésta razón se incluye el código para este lenguaje.

Para llamar a Btrieve desde Basic intérprete, se debe poner en marcha a la interfaz de Basic con los parámetros apropiados, de lo contrario la aplicación no se ejecuta correctamente.

- LA INTERFAZ CON BASIC INTÉRPRETE.

La interfaz con Basic intérprete es una sub-rutina en lenguaje ensamblador, llamada BASXBTRV.EXE, que debe ser ejecutada después de poner en marcha a Btrieve, esta interfaz queda en memoria también, y se elimina sólo al apagar la computadora.

La sub-rutina de interfaz BASXBTRV escribe sobre la pantalla o archivo un número, este número corresponde a la dirección del segmento en notación decimal, que corresponde a la dirección de la sub-rutina. Este número debe de ser recuperado por la aplicación y utilizarlo como parámetro en la instrucción DEF SEG, después de realizar esto, la aplicación puede utilizar la instrucción CALL, para utilizar las operaciones de Btrieve, como se muestra en el programa ejemplo.

El formato de la interfaz es el siguiente:

```
BASXBTRV <archivo>
```

El parámetro <archivo>, corresponde al drive, directorio, nombre y/o extensión del archivo en el que la interfaz colocará la dirección del segmento que la aplicación requiere. En ambiente red, se debe tener cuidado de seleccionar estos nombres, puesto que si se utiliza el mismo para cada estación de trabajo, la aplicación puede estar recuperando un número que no le corresponde.

Ejemplo:

```
C:>BASXBTRV SEG.DIR
```

Después que la interfaz con Basic se instale en memoria y escriba la dirección dentro del archivo SEG.DIR, aparece en pantalla lo siguiente:

```
"Btrieve BASIC interface loaded at segment XXXXX."
```

La aplicación podrá recuperar la dirección del segmento y establecer el contacto con la interfaz, con las siguientes instrucciones:

```
30 OPEN "SEG.DIR" FOR INPUT AS #1
40 INPUT #1, DIR%
50 DEF SEG = DIR%
```

- PONIENDO EN MARCHA BASIC INTÉRPRETE.

Normalmente, BASIC asume que se utilizarán registros de hasta 128 bytes para cualquier archivo que se abra. Para acceder un archivo Btrieve con un registro lógico mayor que 128 bytes, se debe incluir el parámetro de tamaño de archivo, especificando el tamaño necesario. Por ejemplo:

BASIC /S:512

- LLAMANDO A BTRIEVE DESDE BASIC.

Como este lenguaje no tiene manera de definir registros, más que utilizando la instrucción FIELD, es necesario seguir cuatro pasos para abrir un archivo Btrieve desde Basic.

1. Utilizar la instrucción OPEN de Basic con el nombre de archivo "NUL". Ejemplo OPEN "NUL" AS #2
2. Definir los campos del archivo #2 (en este ejemplo). Ejemplo:
FIELD #2,2 AS REG\$,34 AS NOM\$, 30 AS DIR\$, 20 AS TELS
3. Determinar el FCB del archivo, el cual se utiliza como block de posición. Ejemplo:
FCB% = VARPTR (#2).
De esta forma el FCB del archivo #2, se coloca en la variable FCB%, esta variable está asociada con el registro del archivo y es la que se manda como parámetro en la función Btrieve en vez del buffer de datos.
4. Utilizar la función BTRV con la operación 0 *Abrir*, como se describe en el capítulo III.
CALL BTRV (ABRIR%, ST%, FCB%, LAR.BUFFER%, BUF.LLAVE\$, NO.LLAVE%)

Como ya se ha comentado Btrieve requiere de un block de posición para determinar sobre cual archivo se va a realizar la operación; en el caso de BASIC, esto no es posible. Para substituir el block de posición la función Btrieve requiere del FCB (File Control Block), para poder manipular el registro (buffer de datos).

Todas las llamadas a Btrieve desde Basic deben tener el siguiente formato:

CALL BTRV(*Operación%*, *Código de error%*, *FCB%*, *Tamaño del buffer de datos%*,
Buffer de llave\$, Número de llave%)

Consideraciones especiales en los parámetros:

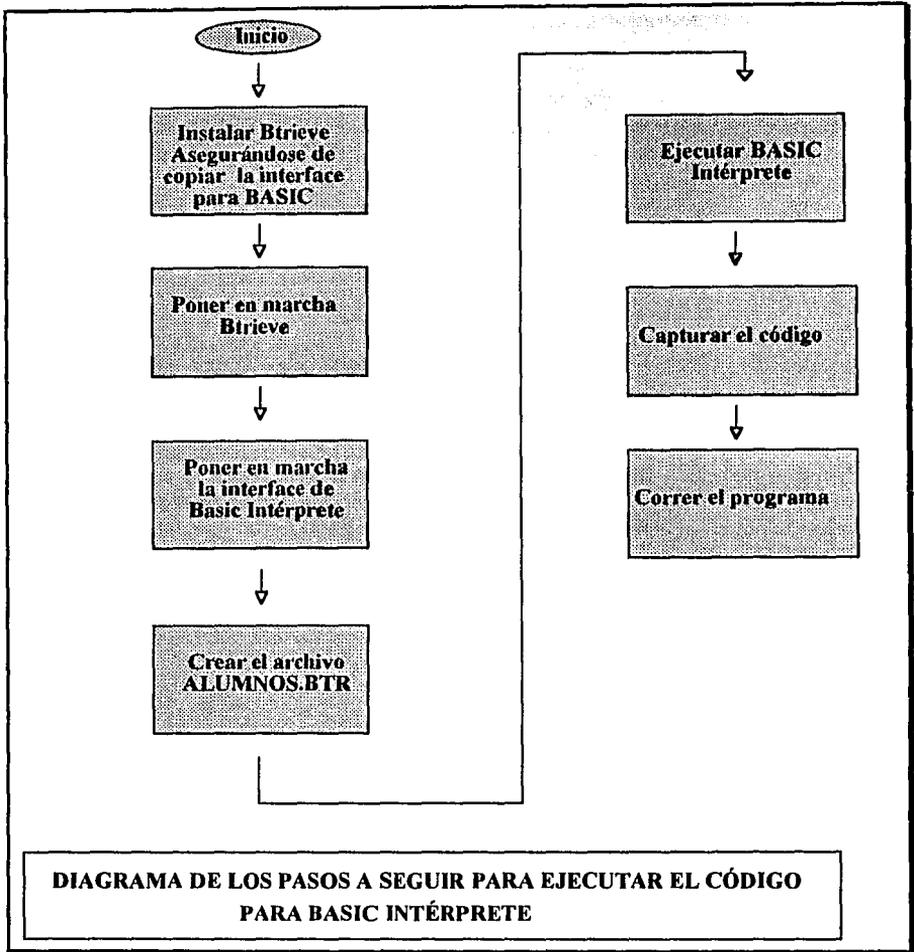
Código de error. Este parámetro corresponde a una variable entera. La variable que se indique es la que Btrieve ocupará para regresar el código de error (de existir alguno).

FCB. Este parámetro corresponde a una variable entera y contiene el FCB antes descrito del archivo con el que se desea operar. Esta variable se relaciona con la información contenida en el registro definido para el archivo con la instrucción FIELD. Este parámetro sustituye al *buffer de datos* anteriormente mencionado.

Para poder ejecutar este programa desde Basic intérprete se deben realizar los siguientes pasos:

1. **Instalar Btrieve (Ver. subcapítulo 2.04 ó 2.05, según el caso). Asegurando que la interfaz para BASIC intérprete sea copiada (archivo: BASXBTRV.EXE)**
2. **Poner en marcha a Btrieve (Ver subcapítulo 2.04 ó 2.05, según el caso). Como el archivo Btrieve que se define en el paso 4, no tiene especificaciones demandantes, se puede poner en marcha Btrieve sin indicar ninguna opción especial. Ejemplo:**
BTRIEVE.↓
3. **Poner en marcha la interfaz de Basic intérprete con Btrieve, dentro del directorio donde el código será capturado:**
BASXBTRV SEG.DIR.↓
4. **Crear el archivo ALUMNOS.BTR como se indica al final del capítulo II, en el Subcapítulo 2.06 tema: "-EJEMPLO DE UN ARCHIVO DE DESCRIPCIÓN".**
BUTIL -CREATE ALUMNOS.BTR ALUMNOS.DES ↓
5. **Ejecutar Basic intérprete:**
BASIC /S:255 ↓
6. **Capturar el código que se presenta y correr el programa.**

A continuación se presenta el diagrama de los pasos a seguir y el código completo del programa muestra, para ejecutarlo se debe utilizar la instrucción RUN de BASIC intérprete:




```

1180 LOCATE 12,17:LINE INPUT TEL2$
1190 LOCATE 13,17:LINE INPUT GEN2$
1200 LOCATE 14,17:LINE INPUT SEM2$
1210 IF VAL(SEM2$)<1 OR VAL(SEM2$)>15 THEN BEEP:GOTO 1200
1220 LOCATE 23,1:PRINT "DESEA DAR DE ALTA ESTOS DATOS (S o N)?":R$=INPUT$(1)
1230 IF R$="N" OR R$="n" THEN 500
1240 IF R$<>"S" AND R$<>"s" THEN BEEP:GOTO 1220
1250 '*** INSERTA EL REG
1300 LSET REG$=MKI$(0)
1310 LSET CTA$=CTA2$
1320 LSET NOM$=NOM2$
1330 LSET DIR$=DIR2$
1340 LSET COL$=COL2$
1350 LSET DEL$=DEL2$
1360 LSET EST$=EST2$
1365 LSET TEL$=TEL2$
1370 LSET GEN$=GEN2$
1380 LSET SEM$=MKI$(VAL(SEM2$))
1400 NO.LLAVE%=0:BUF.LLAVE$=REG$
1410 CALL BTRV(INSERT%,ST%,FCB%,LAR.BUFFER%,BUF.LLAVE$,NO.LLAVE%)
1415 IF ST%=5 THEN LOCATE 23,1:PRINT "EL NOMBRE Y/O LA CUENTA YA EXISTEN. ALTA
INVALIDA...":BEEP:R$=INPUT$(1):GOTO 1000
1420 IF ST%>0 THEN 30000 'Control de errores inesperados
1430 GOTO 1000
1499 '*****
1500 RR$="BAJAS":GOSUB 3800
1510 GOSUB 8000 'Pone datos en pantalla
1520 LOCATE 5,1:PRINT "NOMBRE, CTA o REG:[";SPC(40);"]";
1530 LOCATE 5,20:LINE INPUT R$:IF R$="" THEN 500
1535 LLAVE$=R$
1540 LR%=LEN(R$):IF LR%>40 THEN 1500
1550 IF LR%<=5 THEN NO.LLAVE%=0:R$=MKI$(VAL(R$)): GOTO 1580 'POR REGISTRO
1560 IF LR%<=10 THEN NO.LLAVE%=1:R$=R$+SPACE$(10-LR%):GOTO 1580 'POR CUENTA
1570 NO.LLAVE%=2:R$=R$+SPACE$(40-LR%) 'POR NOMBRE
1580 '**** Consulta con bloqueo sin espera +200
1590 BUF.LLAVE$=R$:LAR.BUFFER%=200:OP%=CONSULTI%+BLOQUEO%
1600 CALL BTRV(OP%,ST%,FCB%,LAR.BUFFER%,BUF.LLAVE$,NO.LLAVE%)
1605 IF ST%=84 THEN LOCATE 23,1:BEEP:PRINT "REGISTRO BLOQUEADO POR OTRA
COMPUTADORA DE LA RED. PULSE ENTER.":R$=INPUT$(1):GOTO 1500
1610 IF ST%=0 THEN 1640
1620 LOCATE 23,1:PRINT "NO SE ENCONTRO EL ALUMNO, PULSE [ENTER] P. CONT.";
1630 R$=INPUT$(1):GOTO 1500
1640 GOSUB 8500 'Pone informaci3n en pantalla
1650 LOCATE 23,1:PRINT "DESEA DARLO DE BAJA (S o N)?":R$=INPUT$(1)
1660 IF R$="N" OR R$="n" THEN GOSUB 1800:GOTO 1500 'Desbloqueo
1670 '*****
1680 CALL BTRV(BORRAR%,ST%,FCB%,CERO%,NADA$,CERO%)
1690 IF ST%<>0 THEN 30000
1700 GOTO 1500
1800 'Liberaci3n del registro bloqueado porque no se di3 de baja.
1805 LAR.BUFFER%=200
1810 CALL BTRV(DESBLOQ%,ST%,FCB%,LAR.BUFFER%,NADA$,NO.LLAVE%)
1820 RETURN
1999 '*****
2000 RR$="CAMBIOS":GOSUB 3800
2005 CAMB%=0
2010 GOSUB 8000 'Pone datos en pantalla
2020 LOCATE 5,1:PRINT "NOMBRE, CTA o REG:[";SPC(40);"]";
2030 LOCATE 5,20:LINE INPUT R$:IF R$="" THEN 500
2040 LR%=LEN(R$):IF LR%>40 THEN 2000
2050 IF LR%<=5 THEN NO.LLAVE%=0:R$=MKI$(VAL(R$)): GOTO 2080 'POR REGISTRO
2060 IF LR%<=10 THEN NO.LLAVE%=1:R$=R$+SPACE$(10-LR%):GOTO 2080 'POR CUENTA
2070 NO.LLAVE%=2:R$=R$+SPACE$(40-LR%) 'POR NOMBRE
2080 '**** Consulta con bloqueo sin espera +200
2090 BUF.LLAVE$=R$:LAR.BUFFER%=200:OP%=CONSULTI%+BLOQUEO%
2100 CALL BTRV(OP%,ST%,FCB%,LAR.BUFFER%,BUF.LLAVE$,NO.LLAVE%)

```

```

2105 IF ST%=84 THEN LOCATE 23,1:BEEP:PRINT "REGISTRO BLOQUEADO POR OTRA
COMPUTADORA DE LA RED. PULSE ENTER.";R$=INPUT$(1):GOTO 2000
2110 IF ST%=0 THEN 2140
2120 LOCATE 23,1:PRINT "NO SE ENCONTRO EL ALUMNO, PULSE [ENTER] P. CONT.";
2130 R$=INPUT$(1):GOTO 2000
2140 GOSUB 8500 'Pone informaci3n en pantalla
2150 LOCATE 23,1:PRINT "ESTA CORRECTA LA INFORMACION (S O N)?":R$=INPUT$(1)
2160 IF R$="S" OR R$="s" THEN 2300 ELSE LOCATE 23,1:PRINT SPACES(79);
2170 LOCATE 23,1:PRINT "No. DE DATO A MODIFICAR:";R$=INPUT$(1)
2175 LOCATE 23,1:PRINT SPACES(79);
2180 CC%=VAL(R$):IF CC%<1 THEN 2150
2185 ***** Cambia el dato seleccionado
2190 CAMB%=-1:ON CC% GOTO 2210,2220,2230,2240,2250,2260,2270,2280,2290
2210 LOCATE 6,17:PRINT SPC(10)::LOCATE ,17:LINE INPUT CTA2$
2215 IF CTA2$="" OR LEN(CTA2$)<9 THEN 2210
2217 GOTO 2170
2220 LOCATE 7,17:PRINT SPC(40)::LOCATE ,17:LINE INPUT NOM2$
2225 IF NOM2$="" OR LEN(NOM2$)<=10 THEN 2220
2227 GOTO 2170
2230 LOCATE 8,17:PRINT SPC(40)::LOCATE ,17:LINE INPUT DIR2$:GOTO 2170
2240 LOCATE 9,17:PRINT SPC(35)::LOCATE ,17:LINE INPUT COL2$:GOTO 2170
2250 LOCATE 10,17:PRINT SPC(25)::LOCATE ,17:LINE INPUT DEL2$:GOTO 2170
2260 LOCATE 11,17:PRINT SPC(25)::LOCATE ,17:LINE INPUT EST2$:GOTO 2170
2270 LOCATE 12,17:PRINT SPC(16)::LOCATE ,17:LINE INPUT TEL2$:GOTO 2170
2280 LOCATE 13,17:PRINT SPC(05)::LOCATE ,17:LINE INPUT GEN2$:GOTO 2170
2290 LOCATE 14,17:PRINT SPC(02)::LOCATE ,17:LINE INPUT SEM2$
2295 IF VAL(SEM2$)<1 OR VAL(SEM2$)>15 THEN BEEP:GOTO 2290
2297 GOTO 2170
2300 'Actualiza los cambios **
2310 IF NOT CAMB% THEN GOSUB 1800:GOTO 2000 'Desbloqueo
2330 LSET CTA$=CTA2$
2340 LSET NOM$=NOM2$
2350 LSET DIR$=DIR2$
2360 LSET COL$=COL2$
2370 LSET DEL$=DEL2$
2380 LSET EST$=EST2$
2390 LSET TEL$=TEL2$
2400 LSET GEN$=GEN2$
2410 LSET SEM$=MKI$(VAL(SEM2$))
2420 LAR.BUFFER%=200
2430 CALL BTRV(ACTUALIZ%,ST%,FCB%,LAR.BUFFER%,NADA$,NO.LLAVE%)
2440 IF ST%>0 THEN 30000
2450 GOTO 2000
2499 *****
2500 RR$="CONSULTA":GOSUB 3800
2510 GOSUB 8000 'Pone datos en pantalla
2520 LOCATE 5,1:PRINT "NOMBRE, CTA o REG:[";SPC(40);"]";
2530 LOCATE 5,20:LINE INPUT R$:IF R$="" THEN 500
2540 LR%=LEN(R$):IF LR%>40 THEN 2500
2550 IF LR%<=5 THEN NO.LLAVE%=0:R$=MKI$(VAL(R$)): GOTO 2600 'POR REGISTRO
2560 IF LR%<=10 THEN NO.LLAVE%=1:R$=R$+SPACES(10-LR%):GOTO 2600 'POR CUENTA
2570 NO.LLAVE%=2:R$=R$+SPACES(40-LR%) 'POR NOMBRE
2600 **** Consulta igual a BUF.LLAVES%
2610 BUF.LLAVES%=R$:LAR.BUFFER%=200
2620 CALL BTRV(CONSULTI%,ST%,FCB%,LAR.BUFFER%,BUF.LLAVE$,NO.LLAVE%)
2630 IF ST%=0 THEN 2700
2640 LOCATE 23,1:PRINT "NO SE ENCONTRO EL ALUMNO, PULSE [ENTER] P. CONT.";
2650 R$=INPUT$(1):GOTO 2500
2700 GOSUB 8500 'Pone informaci3n en pantalla
2830 LOCATE 23,1:PRINT "DESEA SEGUIR CONSULTANDO (S O N)?":R$=INPUT$(1)
2840 IF R$="N" OR R$="n" THEN 500 ELSE 2500
2850 *****
3000 RR$="CATALOGOS IMPRESOS":GOSUB 3800
3010 LOCATE 7,1
3020 PRINT "1) POR REGISTRO"
3030 PRINT "2) POR CUENTA"

```

```

3040 PRINT "3) POR NOMBRE"
3050 PRINT "4) POR SEMESTRE/CUENTA"
3060 PRINT "5) POR SEMESTRE/NOMBRE":PRINT
3070 PRINT " OPCION ==>";R$=INPUT$(1)
3080 OPC=VAL(R$):IF R$="" THEN 500
3090 IF OPC<1 OR OPC>5 THEN BEEP:GOTO 3010
3100 GOSUB 3800
3110 LOCATE 23,1:PRINT "ESTA LISTA SU IMPRESORA (S O N)?":R$=INPUT$(1)
3120 IF R$<>"N" AND R$<>"n" AND R$<>"S" AND R$<>"s" THEN BEEP:GOTO 3110
3130 IF R$="N" OR R$="n" THEN 500
3140 LC%=0:PAG%=0:LOCATE 23,1:PRINT "IMPRESION EN PROCESO . . . ";
3150 LAR.BUFFER%=200:NO.LLAVE%=OPC%-1:BUF.LLAVE$=SPACE$(40)
3160 CALL BTRV(C.PRIMER%,ST%,FCB%,LAR.BUFFER%,BUF.LLAVE$,NO.LLAVE%)
3170 WHILE ST%=0
3180 IF LC%=0 OR LC%>=60 THEN GOSUB 10000
3190 LPRINT USING DET$,CVI(REG$);CTA$;NOM$;CVI(SEM$);GEN$;TEL$
3280 LC%=LC%+1
3290 LAR.BUFFER%=200
3300 CALL BTRV(C.SIGUI%,ST%,FCB%,LAR.BUFFER%,BUF.LLAVE$,NO.LLAVE%)
3310 WEND
3320 IF LC%<>0 THEN LPRINT CHR$(12)
3330 GOTO 500
3499 '***** Fin normal del programa. Cierra archivo.
3500 CALL BTRV(CERRAR%,ST%,FCB%,CERO$,NADA$,CERO%)
3505 IF ST%<>0 THEN 30000
3510 CLS:END
3799 '***** Rutina de encabezado.
3800 CLS:SI$="PROGRAMA MUESTRA TRABAJA UTILIZANDO EL MANEJADOR DE BASES DE DATOS
DE BTREIVE":LOCATE 1,FND$(SI$):PRINT SI$;
3805 NP$="AGENDA DE ALUMNOS DE M.A.C. EN BASIC INTERPRETE":LOCATE 2,FND$(NP$)
:PRINT NP$:PRINT STRING$(80," ")
3810 COLOR 0,7:LOCATE 3,FND$(RR$):PRINT RR$;:COLOR 7,0
3820 RETURN
4999 '*****
5000 RR$="CONSULTA EN PANTALLA POR SEMESTRE/CUENTA o SEMESTRE/NOMBRE":
GOSUB 3800
5010 LOCATE 10,1
5020 PRINT "INDIQUE EL SEMESTRE A CONSULTAR:":LINE INPUT NSEM$
5025 IF NSEM$="" THEN 500
5030 NSEM%=VAL(NSEM$):IF NSEM%<1 OR NSEM%>15 THEN BEEP:GOTO 5020
5040 PRINT "CONSULTA POR 1)CUENTA 0 2)NOMBRE (1 o 2)? ";:ORDEN$=INPUT$(1)
5050 ORDEN%=VAL(ORDEN$):IF ORDEN%<>1 AND ORDEN%<>2 THEN BEEP:GOTO 5040
5060 GOSUB 3800
5070 LOCATE 5,1:PRINT TITP$(ORDEN%):PRINT STRING$(80,"-")
5075 '***** Uso de las llaves segmentadas
5080 BUF.LLAVE$=MKI$(NSEM%)+SPACE$(40):LAR.BUFFER%=200:NO.LLAVE%=2+ORDEN%
5090 CALL BTRV(C.MAYOR%,ST%,FCB%,LAR.BUFFER%,BUF.LLAVE$,NO.LLAVE%)
5100 IF ST%=9 OR ST%=4 OR CVI(SEM$)<>NSEM% THEN LOCATE 23,1:PRINT "NO EXISTEN
ALUMNOS EN EL SEMESTRE INDICADO...";R$=INPUT$(1):GOTO 5000
5110 IF ST%<>0 THEN 30000
5120 IF ORDEN%=1 THEN PRINT USING
DETP$(ORDEN%);CVI(REG$);CTA$;NOM$;TEL$;CVI(SEM$)
5130 IF ORDEN%=2 THEN PRINT USING
DETP$(ORDEN%);CVI(REG$);NOM$;CTA$;TEL$;CVI(SEM$)
5135 R%=7
5140 WHILE ST%=0 AND CVI(SEM$)=NSEM%
5150 LAR.BUFFER%=200
5160 CALL BTRV(C.SIGUI%,ST%,FCB%,LAR.BUFFER%,BUF.LLAVE$,NO.LLAVE%)
5170 IF ST%=9 OR ST%=4 OR CVI(SEM$)<>NSEM% THEN 5200
5180 IF ST%<>0 THEN 30000
5190 IF ORDEN%=1 THEN PRINT USING DETP$(ORDEN%);CVI(REG$);CTA$;NOM$;TEL$;CVI(SEM$)
5195 IF ORDEN%=2 THEN PRINT USING DETP$(ORDEN%);CVI(REG$);NOM$;CTA$;TEL$;CVI(SEM$)
5197 R%=R%+1:IF R%=21 THEN GOSUB 5300:FOR I%=7 TO 21:LOCATE I%,1:
PRINT SPACE$(79);:NEXT I$:LOCATE 7,1:R%=7
5200 WEND
5210 GOSUB 5300
5220 GOTO 5000

```

```

5299 '***** Rutina de pausa
5300 LOCATE 23,1:PRINT "PULSE CUALQUIER TECLA PARA CONTINUAR ...";W$=INPUT$(1)
5320 LOCATE 23,1:PRINT SPACE$(79);RETURN
7999 '***** Rutina que pone las preguntas en pantalla (pone_preg)
8000 LOCATE 6,1
8010 PRINT "[1] CUENTA      : [";SPC(10);"]"
8020 PRINT "[2] NOMBRE      : [";SPC(40);"]"
8030 PRINT "[3] DIRECCION  : [";SPC(40);"]"
8040 PRINT "[4] COLONIA    : [";SPC(35);"]"
8050 PRINT "[5] DEL O CIU   : [";SPC(25);"]"
8060 PRINT "[6] ESTADO    : [";SPC(25);"]"
8070 PRINT "[7] TELEFONOS  : [";SPC(16);"]"
8080 PRINT "[8] GENERACION: [";SPC(5);"]"
8090 PRINT "[9] SEMESTRE  : [";SPC(2);"]"
8100 RETURN
8499 '***** Rutina que pone los datos del reg. en la pantalla (pone_datos)
8500 LOCATE 6,17:PRINT CTA$:CTA2$=CTA$
8510 LOCATE 7,17:PRINT NOM$:NOM2$=NOM$
8520 LOCATE 8,17:PRINT DIR$:DIR2$=DIR$
8530 LOCATE 9,17:PRINT COL$:COL2$=COL$
8540 LOCATE 10,17:PRINT DEL$:DEL2$=DEL$
8550 LOCATE 11,17:PRINT EST$:EST2$=EST$
8560 LOCATE 12,17:PRINT TEL$:TEL2$=TEL$
8570 LOCATE 13,17:PRINT GEN$:GEN2$=GEN$
8580 LOCATE 14,17:PRINT USING "###";CVI(SEM$):SEM2$=STR$(CVI(SEM$))
8590 LOCATE 4,60:PRINT USING "REGISTRO: [#####]";CVI(REG$);
8600 RETURN
9999 SAVE "BTRV":END
10000 '***** Rutina para el ENCABEZADO DEL REPORTE IMPRESO
10010 IF LC%>=60 THEN LPRINT CHR$(12):LC%=0
10030 LPRINT SPACE$(FND$(SI$));SI$
10040 LPRINT SPACE$(FND$(NP$));NP$
10050 IF OPC%=1 THEN RR$="REPORTE ORDENADO POR No. DE REGISTRO"
10060 IF OPC%=2 THEN RR$="REPORTE ORDENADO POR No. DE CUENTA"
10070 IF OPC%=3 THEN RR$="REPORTE ORDENADO POR NOMBRE"
10080 IF OPC%=4 THEN RR$="REPORTE ORDENADO POR No. DE SEMESTRE Y CUENTA"
10090 IF OPC%=5 THEN RR$="REPORTE ORDENADO POR No. DE SEMESTRE Y NOMBRE"
10100 LPRINT SPACE$(FND$(RR$));RR$
10105 PAG%=PAG%+1:LPRINT USING "PAG:[###]";PAG%
10110 LPRINT STRING$(80,"-")
10120 LPRINT TIT$
10130 LPRINT STRING$(80,"-")
10140 LC%=7:RETURN
29999 '***** Rutina para la manipulaci3n de errores inesperados.
30000 BEEP:LOCATE 24,1:PRINT USING "ERROR DE BTRV No. ### PULSE ENTER PARA
CONTINUAR ";ST$;R$=INPUT$(1)
30010 IF ST%<>20 THEN CALL BTRV(RESTAU$,ST$,CERO$,CERO$,NADAS,CERO$)'Restaurar
30020 CLS:END 'Salida inesperada del programa.

```

Nota: Los renglones que aparentemente no cuentan con número de línea, forman parte de la línea superior.

5.03 MODIFICACIONES AL CÓDIGO DEL PROGRAMA REALIZADO EN BASIC INTÉRPRETE, PARA COMPILARLO EN Microsoft QuickBASIC VERSIÓN 2.0 Y 4.5.

En este subcapítulo se describen las modificaciones necesarias al código presentado en el subcapítulo anterior, con el fin de que el programa pueda ser compilado.

El compilador de lenguaje BASIC; Microsoft QuickBASIC V2.0, ejecuta casi cualquier código desarrollado para BASIC intérprete, se puede decir, que en cierta medida es compatible.

Por otro lado la versión más actual es Microsoft QuickBASIC V4.5, tiene más problemas (en funciones CALL) para ejecutar un código desarrollado para BASIC intérprete. Sin embargo ofrece un BASIC modificado que permite modular los programas que es muy diferente en comparación con las versiones anteriores. Esta última versión permite abrir un archivo Btrieve por medio del uso de las instrucciones OPEN y FIELD, como ya se indicó antes, pero también permite la forma directa como se realiza en compiladores de lenguaje C o Pascal, utilizando variables de tipo registro, puesto que ya utiliza el block de parámetros y no el RCB como el Basic intérprete y la versión 2.0 de éste compilador.

- MODIFICACIONES AL CÓDIGO DE BASIC INTÉRPRETE PARA COMPILARLO EN Microsoft QuickBASIC V2.0.

En este caso sólo se deben de hacer 2 cambios sencillos al código.

1. Eliminar del código las líneas de la 100 a la 140 y la línea 9999.
2. Omitir la puesta en marcha de la interfaz para BASIC intérprete. En este caso la interfaz es: BASXBTRB.OBJ.

Esta versión del compilador permite ejecutar un programa en memoria o bien, crear el archivo ejecutable, a continuación se describen los pasos para cada opción:

Para poder ejecutar en memoria este programa desde Microsoft QuickBASIC versión 2.0, se deben realizar los siguientes pasos.

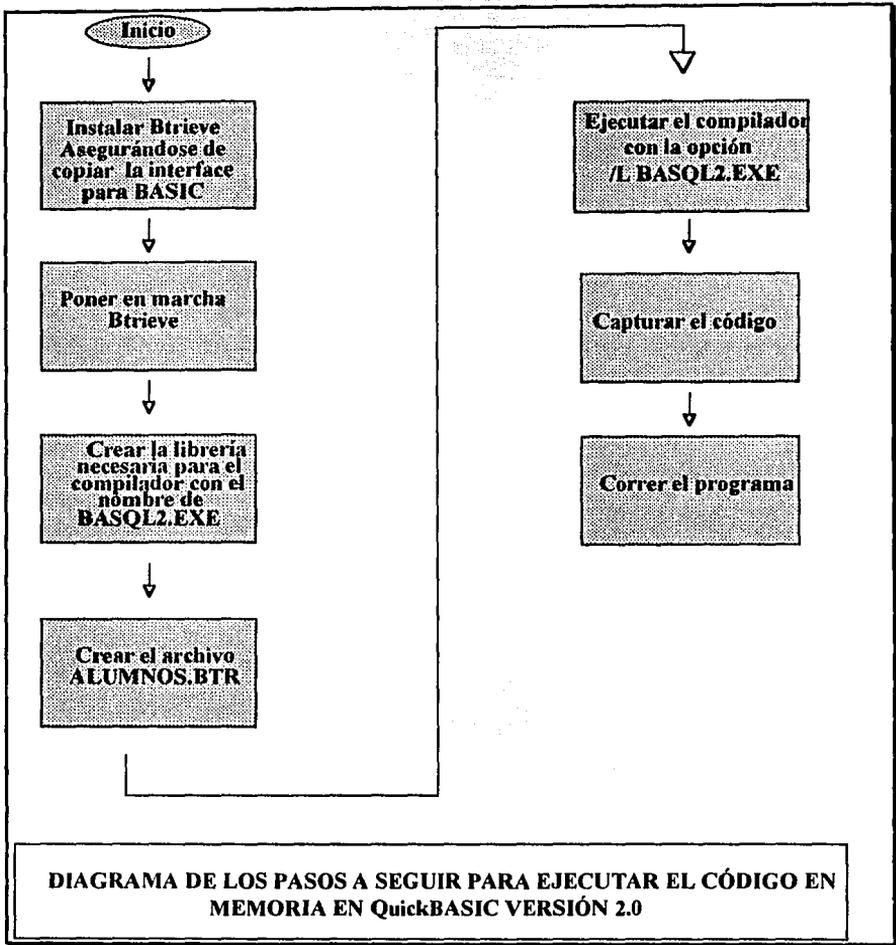
1. **Instalar Btrieve (Ver. subcapítulo 2.04 ó 2.05, según el caso). Asegurando que la interfaz para BASIC sea copiada (archivo: BASXBTRV.OBJ)**
2. **Poner en marcha a Btrieve (Ver subcapítulo 2.04 ó 2.05, según el caso). Como el archivo Btrieve que se define en el paso 4, no tiene especificaciones demandantes, se puede poner en marcha Btrieve sin indicar ninguna opción especial. Ejemplo: BTRIEVE_**

3. Crear la librería necesaria para el compilador con el nombre de archivo: **BASQL2.EXE**
`BUILDLIB BASXBTRV.OBJ, BASQL2.EXE; ↵`

Nota: el programa que crea la librería (**BUILDLIB.EXE**), se encuentra en los discos del compilador. La interfaz **BASXBTRV.OBJ**, se encuentra en los discos de Btrieve.

4. Crear el archivo **ALUMNOS.BTR** como se indica al final del capítulo II, en el Subcapítulo 2.06 tema: **"-EJEMPLO DE UN ARCHIVO DE DESCRIPCIÓN"**.
`BUTIL -CREATE ALUMNOS.BTR ALUMNOS.DES ↵`
5. Ejecutar el compilador con la opción **/L BASQL2.EXE:**
`QB /L BASQL2.EXE ↵`
6. Capturar el código que se presenta en el subcapítulo 5.02, con las modificaciones mencionadas en este subcapítulo. Y correr el programa (^R).

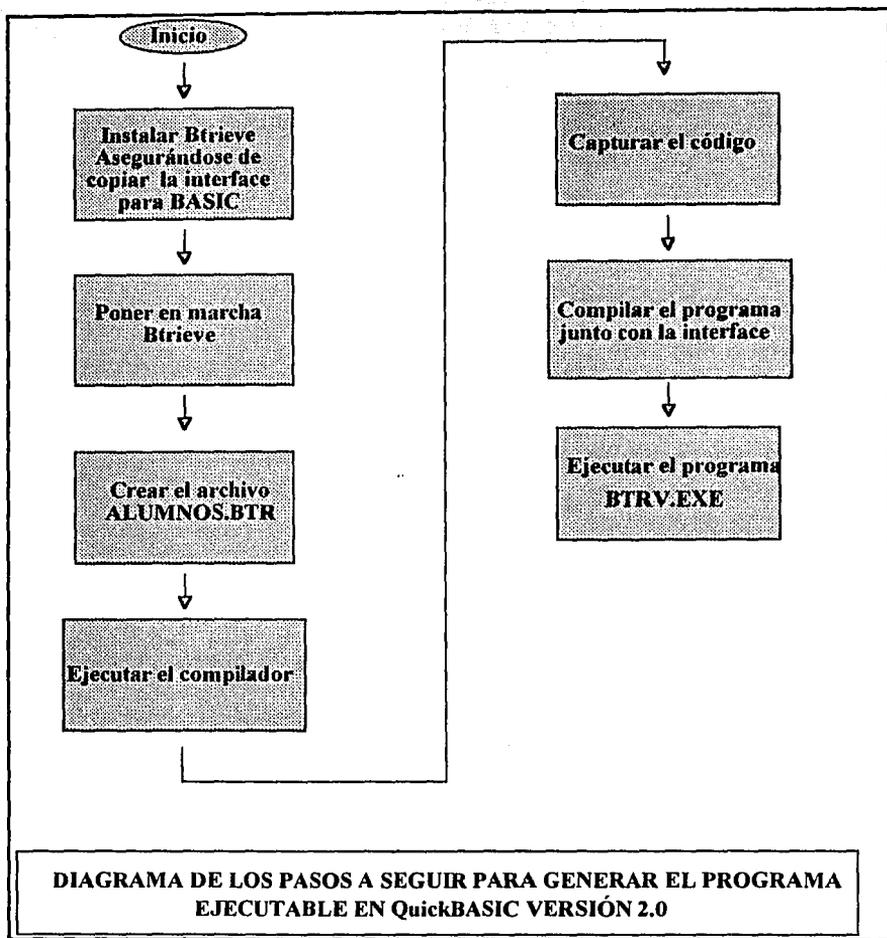
A continuación se presenta el diagrama de los pasos a seguir:



Para poder generar el programa ejecutable desde Microsoft QuickBASIC Versión 2.0, se deben realizar los siguientes pasos.

1. Instalar Btrieve (Ver. subcapítulo 2.04 ó 2.05, según el caso). Asegurando que la interfaz para BASIC sea copiada (archivo: BASXBTRV.OBJ)
 2. Poner en marcha a Btrieve (Ver subcapítulo 2.04 ó 2.05, según el caso). Como el archivo Btrieve que se define en el paso 3, no tiene especificaciones demandantes, se puede poner en marcha Btrieve sin indicar ninguna opción especial. Ejemplo:
BTRIEVE.┘
 3. Crear el archivo ALUMNOS.BTR como se indica al final del capítulo II, en el Subcapítulo 2.06 tema: "-EJEMPLO DE UN ARCHIVO DE DESCRIPCIÓN".
BUTIL -CREATE ALUMNOS.BTR ALUMNOS.DES ┘
 4. Ejecutar el compilador
QB BTRV.BAS ┘
 5. Capturar el código que se presenta en el subcapítulo 5.02, con las modificaciones mencionadas en este subcapítulo. Grabar el programa y salir a DOS.
 6. Compilar el programa junto con la interfaz:
QB BTRV.BAS;
LINK BTRV.OBJ+BASXBTRV.OBJ;
- Nota: Estos pasos generan el programa ejecutable BTRV.EXE que requiere de la presencia del archivo del compilador BRUN20.EXE. Para que el programa ejecutable corra de manera independiente utilice la opción /O al compilar el programa.
7. Ejecutar el programa BTRV.EXE.
BTRV.┘

A continuación se presenta el diagrama de los pasos a seguir:



- MODIFICACIONES AL CÓDIGO DE BASIC INTÉRPRETE PARA COMPILARLO EN Microsoft QuickBASIC V4.5.

En este caso se debe:

1. Eliminar del código las líneas de la 100 a la 140 y la línea 9999.
2. Omitir la puesta en marcha de la interfaz para BASIC intérprete. En este caso la interfaz es: QB6RBTRB.OBJ.

Esta versión del compilador permite ejecutar el programa en memoria para depurarlo y también generar el programa ejecutable. A continuación se describen los pasos para utilizar este compilador:

1. **Instalar Btrieve (Ver. subcapítulo 2.04 ó 2.05, según el caso). Asegurando que la interfaz para el compilador sea copiada (archivo:QB6RBTRV.OBJ)**
2. **Poner en marcha a Btrieve (Ver subcapítulo 2.04 ó 2.05, según el caso). Como el archivo Btrieve que se define en el paso 4, no tiene especificaciones demandantes, se puede poner en marcha Btrieve sin indicar ninguna opción especial. Ejemplo: BTRIEVE.┘**
3. **Crear las librerías necesarias para el compilador con los nombres de archivo BTRIEVE.QLB (para correr el programa en memoria), y BTRIEVE.LIB (se requiere al generar el programa ejecutable)**
LINK /Q QB6RBTRV.OBJ, BTRIEVE.QLB,,BQLB45.LIB;┘
LIB BTRIEVE.LIB+QB6RBTRV.OBJ;┘

Nota: Los programas que crean las librerías (LINK.EXE, LIB.EXE Y BQLB45.LIB), se encuentran en los discos del compilador. La interfaz QB6RBTRV.OBJ, se encuentra en los discos de Btrieve.

4. **Crear el archivo ALUMNOS.BTR como se indica al final del capítulo II, en el Subcapítulo 2.06 tema: "-EJEMPLO DE UN ARCHIVO DE DESCRIPCIÓN".**
BUTIL -CREATE ALUMNOS.BTR ALUMNOS.DES ┘
5. **Ejecutar el compilador con la opción /L BTRIEVE:**
QB /L BTRIEVE.QLB ┘
6. **Capturar el código que se presenta en el subcapítulo 5.02, con las modificaciones mencionadas en este subcapítulo, más las modificaciones que se presentan a continuación:**

```

310 AL% = SADD(REG%) 'Define el registro a utilizar (Buffer de Datos)
340 POS% = SPACE$(128)'variable para con tener el Block de posición.
400 CALL BTRV(ABRIR%, st%, POS%, AL%, LAR.BUFFER%, BUF.LLAVES, NO.LLAVE%)
1410 CALL BTRV(INSERT%, st%, POS%, AL%, LAR.BUFFER%, BUF.LLAVES, NO.LLAVE%)
1600 CALL BTRV(OP%, st%, POS%, AL%, LAR.BUFFER%, BUF.LLAVES, NO.LLAVE%)
1680 CALL BTRV(BORRAR%, st%, POS%, CER0%, CER0%, NADAS, CER0%)
1810 CALL BTRV(DESBLQ%, st%, POS%, LAR.BUFFER%, NADAS, NO.LLAVE%)
2100 CALL BTRV(OP%, st%, POS%, AL%, LAR.BUFFER%, BUF.LLAVES, NO.LLAVE%)
2430 CALL BTRV(ACTUALIZ%, st%, POS%, AL%, LAR.BUFFER%, NADAS, NO.LLAVE%)
2620 CALL BTRV(CONSULTI%, st%, POS%, AL%, LAR.BUFFER%, BUF.LLAVES, NO.LLAVE%)
3160 CALL BTRV(C.PRIMER%, st%, POS%, AL%, LAR.BUFFER%, BUF.LLAVES, NO.LLAVE%)
3300 CALL BTRV(C.SIGUI%, st%, POS%, AL%, LAR.BUFFER%, BUF.LLAVES, NO.LLAVE%)
3500 CALL BTRV(CERRAR%, st%, POS%, CER0%, CER0%, NADAS, CER0%)
5090 CALL BTRV(C.MAYOR%, st%, POS%, AL%, LAR.BUFFER%, BUF.LLAVES, NO.LLAVE%)
5160 CALL BTRV(C.SIGUI%, st%, POS%, AL%, LAR.BUFFER%, BUF.LLAVES, NO.LLAVE%)
30010 IF ert<20 THEN CALL BTRV(RESTAU%,st%,NADAS,CER0%,CER0%,NADAS,CER0%)

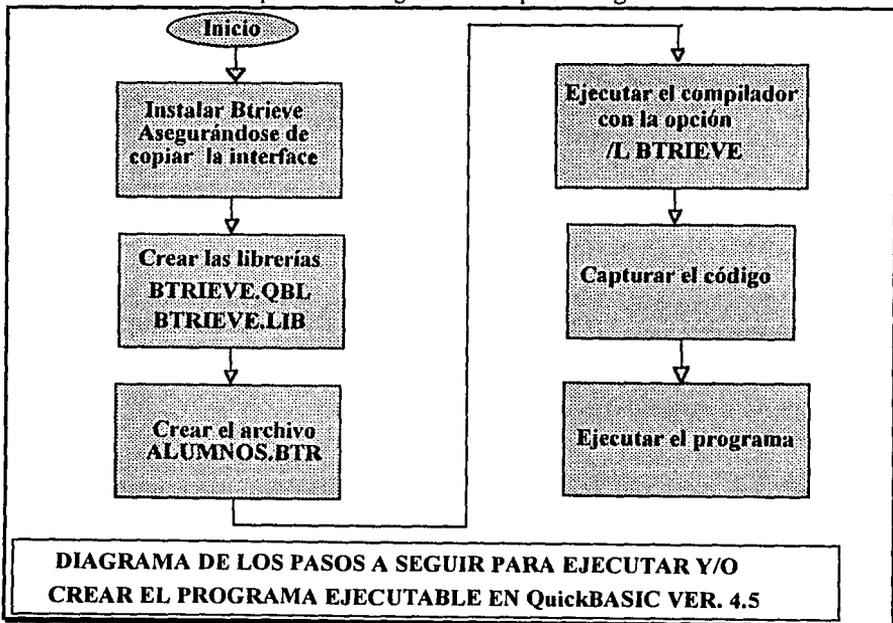
```

Las líneas anteriores substituyen a las del código original.

En la interfaz con este compilador, la función Btrieve cambia; donde el buffer de datos es representado por la dirección de la variable que contiene la información correspondiente, la dirección puede ser la del primer campo definido por la instrucción FIELD, o bien, la dirección de una variable tipo registro que se obtiene con la instrucción VARPTR(REG), ver notas iniciales de la interfaz QB6RBTRV.OBJ del apéndice D.

Después de realizar las correcciones al código, sólo resta correr el programa, o bien, generar el programa ejecutable.

A continuación se presenta el diagrama de los pasos a seguir:



5.04 INTERFAZ CON TURBO C VERSIÓN 2.0 Y EL PROGRAMA EJEMPLO CODIFICADO.

La interfaz con Turbo C v2.0, es mucho más directa que la descrita con BASIC. En este caso la interfaz se implementa como un archivo de cabecera, (como el `STDIO.H` por ejemplo), el nombre del archivo que contiene la interfaz es `TURCBTRV.C`. A diferencia de las interfaces para BASIC, en este caso el lenguaje permite desarrollar completamente la interfaz con instrucciones propias de TURBO C, es decir, que la interfaz `TURCBTRV.C` forma parte de la aplicación, por medio de la instrucción:
`# include "TURCBTRV.C"`.

Existe una consideración importante en este caso; para definir cadenas en el lenguaje C se requiere de un byte de control que determina el fin de la cadena. El código del programa muestra, coloca dentro del archivo `Btrieve`, el carácter de control para cada una de las cadenas del registro, por lo tanto el largo lógico y físico de éste se modifica a comparación de la definición del archivo para BASIC. Originalmente el tamaño del registro era de 200 bytes, en el caso de este ejemplo el tamaño es de 208, puesto que en el registro existen 8 cadenas. `Btrieve` previene este "problema", para lo cual define un tipo cadena especial que es el Z-CADENA (consultar el apéndice B). Todas estas consideraciones deben ser reflejadas en el archivo `Btrieve` previsto para BASIC.

A continuación se presentan las modificaciones del archivo `Btrieve ALUMNOS.BTR`. El nombre del archivo `Btrieve` que requiere el código desarrollado para TURBO C se llama `ALUM_C.BTR`.

Archivo: `ALUM_C.DES`

```
record=208 variable=n key=5 page=1024 replace=y data=y
position=1 length=2 duplicates=n modifiable=y type=aut
descending=n alternate=n manual=n null=n segment=n
position=3 length=11 duplicates=n modifiable=y type=zst
descending=n alternate=n manual=n null=n segment=n
position=14 length=41 duplicates=n modifiable=y type=zst
descending=n alternate=n manual=n null=n segment=n
position=207 length=2 duplicates=y modifiable=y type=int
descending=n alternate=n manual=n null=n segment=y
position=3 length=11 duplicates=y modifiable=y type=zst
descending=n alternate=n manual=n null=n segment=n
position=207 length=2 duplicates=y modifiable=y type=int
descending=n alternate=n manual=n null=n segment=y
position=14 length=41 duplicates=y modifiable=y type=zst
descending=n alternate=n manual=n null=n segment=n
```

Se debe escribir el contenido de la tabla dentro del archivo de descripción, con el nombre `ALUM_C.DES`.

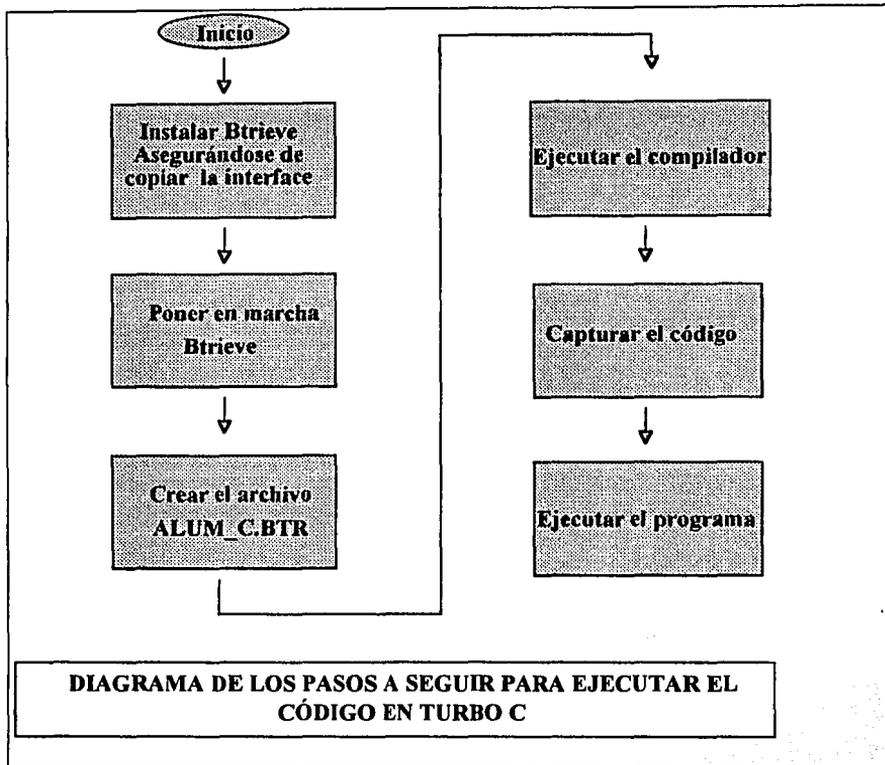
Sólo resta generar el archivo Btrieve y esto se debe realizar con la siguiente instrucción:

```
BUTIL -CREATE ALUM_C.BTR ALUM_C.DES ␣
```

Para poder ejecutar el programa se deben seguir los siguientes pasos:

1. **Instalar Btrieve (Ver. subcapítulo 2.04 ó 2.05, según el caso). Asegurando que la interfaz para el compilador sea copiada (archivo:TURCBTRV.C)**
2. **Poner en marcha a Btrieve (Ver subcapítulo 2.04 ó 2.05, según el caso). Como el archivo Btrieve que se define en el paso 3, no tiene especificaciones demandantes, se puede poner en marcha Btrieve sin indicar ninguna opción especial. Ejemplo: BTRIEVE.␣**
3. **Crear el archivo ALUM_C.BTR como se indicó anteriormente**
BUTIL -CREATE ALUM_C.BTR ALUM_C.DES ␣
4. **Ejecutar el compilador:**
TC ␣
5. **Capturar el código que se presenta a continuación y correr el programa:**

A continuación se presenta el diagrama de los pasos a seguir y el código completo del programa ejemplo:



```

/* EL ARCHIVO turcbtrv.c ES LA INTERFAZ CON BTRIEVE.
ESTE PROGRAMA REQUIERE QUE EL ARCHIVO ALUM_C.BTR
SE ENCUENTRE EN EL MISMO DIRECTORIO. */

# include "turcbtrv.c"

# include "stdio.h"
# include "conio.h"

const
abrir =0, cerrar=1, insert =2, restau =28, consulti =5, borrar =4,
actualiz=3, c mayor=8, c sigui=6, c_primer=12, bloqueo=200, desbloq=27,
true =-1, false =0, cero =0;

/* SE ADICIONA UN CARACTER A TODAS LAS CADENAS PARA GRABAR EN EL ARCHIVO
EL IDENTIFICADOR DE FIN DE CADENA */
struct regis
{
int reg;
char cta[10+1];
char nom[40+1];
char dir[40+1];
char col[35+1];
char del[25+1];
char est[25+1];
char tel[16+1];
char gen[05+1];
int sem;
} al;

char archivo[256],r[256],rr[256],rc;
char tit[256][2];

char op,rc;

char buf_llave[40+1];

struct segment{
int sem;
char lla[40+1];
} llave34;

int i,n,k,j,seme,orden,lr,lc,pag,cc;
char ib,nb,jb,camb;

int no_llave,st,lar_buffer;

char nada[128];

char pos[128];

main()
{
strcpy(tit[1],"REG CUENTA NOMBRE DEL ALUMNO TELEFONO SEM");
strcpy(tit[2],"REG NOMBRE DEL ALUMNO CUENTA TELEFONO SEM");

/* APERTURA DEL ARCHIVO */
clrscr();
nada[0]=0;

lar_buffer=208;
no_llave=0;
strcpy(archivo,"ALUM C. BTR");
st = BTRV(abrir,pos,nada,&lar_buffer,archivo,0);
if (st!=0) error_exit();

```

```

/* MENU */
menu:
encabezado("MENU");
do {
gotoxy(1,23);
color(0,7);
cputs("1)ALTAS| 2)BAJAS| 3)CAMBIOS| 4)CONSULTAS| 5)CONS. X SEM.| 6)CATALOGO| 7)FIN =>");
color(7,0);
op=getche();
} while (!(op=='1' && op!='7'));

switch (op) {

case '1': altas:
encabezado("ALTAS");
pone_preg();

do {
gotoxy(17, 6);gets(al.cta);
}while (strlen(al.cta)<9 && al.cta[0]!='0');
if (al.cta[0]=='0') break; /* SALIDA DE ALTAS */
do {
gotoxy(17, 7);gets(al.nom);
}while (strlen(al.nom)<10 && al.nom[0]!='0');
if (al.nom[0]=='0') break; /* SALIDA DE ALTAS */
gotoxy(17, 8);gets(al.dir);
gotoxy(17, 9);gets(al.col);
gotoxy(17,10);gets(al.del);
gotoxy(17,11);gets(al.est);
gotoxy(17,12);gets(al.tel);
gotoxy(17,13);gets(al.gen);
do {
gotoxy(17,14);cscanf("%d",&al.sem);
}while ((al.sem<1 || al.sem>15));

do {
gotoxy(1,23);
cputs("DESEA DAR DE ALTA ESTOS DATOS (S O N)?");
rc=getche();rc=upcase(rc);
}while (rc!='N' && rc!='S');
if (rc=='N') goto altas;

/* INSERTA EL REGISTRO */
al.reg=0;
lar_buffer=208;
no_llave=0;
st=BTRV(insert,pos,&al,&lar_buffer,&al.reg,no_llave);
if (st==5) {
gotoxy(1,23);
cputs("EL NOMBRE Y/O LA CUENTA YA EXISTEN. ALTA INVALIDA...");
putch(7); /* BEEP */
rc=getch();
} else
if (st!=0) error_exit(st);
goto altas;

case '2': bajas:
do {
encabezado("BAJAS");
pone_preg();
gotoxy(1,5);cprintf("NOMBRE, CTA o REG:[%40s]", "");
gotoxy(20,5);
gets(r);
lr=strlen(r);
}while (lr>40);
if (r[0]=='0') break; /* SALIDA DE BAJAS */

```

```

no_llave=0;

if (lr<=5) (no_llave=0;
            i=atoi(r);
            r[0]='&i); /*COLOCA EL VALOR DEL ENTERO EN LA CADENA*/
            r[1]='&i+1);
            ) else
            if (lr<=10) no_llave=1;
            else
            no_llave=2;

lar_buffer=208; /* CONSULTA Y BLOQUEA (SIN ESPERA) */
st=BTRV(consulti+bloqueo,pos,&al,&lar_buffer,r,no_llave);
r[0]=0;

if (st==84) {
    gotoxy(1,23);
    cputs("REGISTRO BLOQUEADO POR OTRA COMPUTADORA DE LA RED. Pulse [Enter].");
    putch(7);
    rc=getch();
} else
if (st==4) {
    gotoxy(1,23);
    cputs("NO SE ENCONTRO EL ALUMNO, PULSE [ENTER] P. CONT.");
    putch(7);
    rc=getch();
} else
if (st!=0) error_exit(st);
else {
    pone_datos();
    do (gotoxy(1,23);
        cputs("DESEA DARLO DE BAJA (S o N)?");
        rc=getche();rc=upcase(rc);
        )while (rc!='N' && rc!='S');

    if (rc=='N') {
        lar_buffer=208;
        st=BTRV(desblog,pos,&al,&lar_buffer,nada,no_llave);
    } else st=BTRV(borrar,pos,&al,&cerro,nada,cerro);
}

goto bajas;

case '3': cambios:
camb=false;
do {
    encabezado("CAMBIOS");
    pone_preg();
    gotoxy(1,5);cprintf("NOMBRE, CTA o REG:[%40s]", "");
    gotoxy(20,5);
    gets(r);
    lr=strlen(r);
}while (lr>40);
if (r[0]==0) break; /* SALIDA DE CAMBIOS */
no_llave=0;
strcpy(buf_llave,r);
if (lr<=5) {no_llave=0;
            i=atoi(buf_llave);
            buf_llave[0]='&i);
            buf_llave[1]='&i+1);
            } else
            if (lr<=10) no_llave=1;
            else
            no_llave=2;

lar_buffer=208;
st=BTRV(consulti+bloqueo,pos,&al,&lar_buffer,buf_llave,no_llave);

```

```

if (st==84) {
    gotoxy(1,23);
    cputs("REGISTRO BLOQUEADO POR OTRA COMPUTADORA DE LA RED. Pulse [Enter].");
    putch(7);
    rc=getch();
} else
{
    if (st==4) {
        gotoxy(1,23);
        cputs("NO SE ENCONTRO EL ALUMNO, PULSE [ENTER] P. CONT.");
        putch(7);
        rc=getch();
    } else
    {
        if (st!=0) error_exit(st);
        else {
            pone_datos();
            do {
                do {
                    gotoxy(1,23);
                    cputs("ESTA CORRECTA LA INFORMACION (S o N)?");
                    rc=getche();rc=upcase(rc);
                }while (rc!='N' && rc!='S');

                if (rc=='N') do {
                    gotoxy(1,23);stri(79,' ');
                    gotoxy(1,23);
                    cputs("No. DE DATO A MODIFICAR:");rc=getche();
                    cc=rc-48;if (cc<0) cc=0;
                    if (cc>0) camb=true;

                    switch (cc) {
                        case 1: do {
                            gotoxy(17, 6);stri(10,' ');
                            gotoxy(17, 6);gets(al.cta);
                        } while (strlen(al.cta)<9);
                            break;

                        case 2: do {
                            gotoxy(17, 7);stri(40,' ');
                            gotoxy(17, 7);gets(al.nom);
                        }while (strlen(al.nom)<=10);
                            break;

                        case 3: gotoxy(17, 8);stri(40,' ');
                            gotoxy(17, 8);gets(al.dir);
                            break;

                        case 4: gotoxy(17, 9);stri(35,' ');
                            gotoxy(17, 9);gets(al.col);
                            break;

                        case 5: gotoxy(17,10);stri(25,' ');
                            gotoxy(17,10);gets(al.del);
                            break;

                        case 6: gotoxy(17,11);stri(25,' ');
                            gotoxy(17,11);gets(al.est);
                            break;

                        case 7: gotoxy(17,12);stri(16,' ');
                            gotoxy(17,12);gets(al.tel);
                            break;

                        case 8: gotoxy(17,13);stri(15,' ');
                            gotoxy(17,13);gets(al.gen);
                            break;

                        case 9: do {

```

```

        gotoxy(17,14);stri(2,' ');
        gotoxy(17,14);gets(r);al.sem=atoi(r);
        } while(al.sem<1 && al.sem>15);
        break;
    } /*switch*/
} while (cc>0);
} while (rc!='S');
} /* else */

if (camb) { /* ACTUALIZA */
    lar_buffer=208;
    st=BTRV(actualiz,pos,&al,&lar_buffer,buf_llave,no_llave);
    if (st!=0) error_exit(st);
} else {
    lar_buffer=208;
    st=BTRV(desbloq,pos,&al,&lar_buffer,nada,no_llave);
}
goto cambios;

case '4': consulta:
do {
    encabezado("CONSULTAS");
    pone_preg();
    gotoxy(1,5);cprintf("NOMBRE, CTA o REG:[%40s]","");
    gotoxy(20,5);
    gets(r);lr=strlen(r);
}while (lr>40);
if (r[0]==0) break; /* SALIDA DE CONSULTA */

no_llave=0;
if (lr<=5) (no_llave=0;
    i=atoi(r);
    r[0]='&i);
    r[1]='&i+1);
    } else
    if (lr<=10) no_llave=1;
    else
    no_llave=2;

    lar_buffer=208;
    st=BTRV(consulti,pos,&al,&lar_buffer,r,no_llave);
    r[0]=0;
    rc='N'; /* no se desea seguir consultando */
    if (st==4) {
        gotoxy(1,23);
        cprintf("%s","NO SE ENCONTRO EL ALUMNO, PULSE [ENTER] P. CONT.");
        putchar(7);
        rc=getch();
    } else if (st!=0) error_exit();
    else {
        pone_datos();
        do {
            gotoxy(1,23);
            cputs("DESEA SEGUIR CONSULTANDO (S o N)?");
            rc=getche();rc=upcase(rc);
        }while (rc!='N' && rc!='S');
    } /* else */

    if (rc=='S') goto consulta;
    break; /* SALIDA DE CONSULTA */

case '5': consultas:
strcpy(rr,"CONSULTA EN PANTALLA POR SEMESTRE/CUENTA o SEMESTRE/NOMBRE");
encabezado(rr);
do {
    gotoxy(1,10);cputs("INDIQUE EL SEMESTRE A CONSULTAR:");
    gets(r);

```

```

    seme=atoi(r);
}while (seme<0 || seme>15);
if (seme==0) break;
do {
    gotoxy(1,11);
    cputs("1]CUENTA 0 2]NOMBRE (1 o 2)?");
    gets(r);orden=atoi(r);
}while (orden<1 || orden>2);

encabezado(rr);
gotoxy(1,5);
cputs(tit[orden]);gotoxy(1,6);
stri(80,'-');

/* USO DE LAS LLAVES SEMENTADAS */
llave34;sem=seme;llave34;lla[0]=0;
lar_buffer=208;no_llave=2+orden;
st=BTRV(c_mayor,pos,&al,&lar_buffer,&llave34,no_llave);

if (st==9 || st==4 || al.sem!=seme) {
    gotoxy(1,23);
    cputs("NO EXISTEN ALUMNOS EN EL SEMESTRE INDICADO ...");
    putch(7);
    rc=getch();
}
else {
    if (st!=0) error_exit();
    if (orden==1) cprintf("%5d %-10s %-40s %-16s %2d\r\n",al.reg,al.cta,al.nom,al.tel,al.sem);
    else cprintf("%5d %-40s %-10s %-16s %2d\r\n",al.reg,al.nom,al.cta,al.tel,al.sem);
    n=7;
    while (st==0 && al.sem==seme) {
        lar_buffer=208;
        st=BTRV(c_sigui,pos,&al,&lar_buffer,&llave34,no_llave);
        if (st!=0 && st!=9 && st!=4) error_exit();
        if (st==0 && al.sem==seme) {
            if (orden==1) cprintf("%5d %-10s %-40s %-16s %2d\r\n",al.reg,al.cta,al.nom,al.tel,al.sem);
            else cprintf("%5d %-40s %-10s %-16s %2d\r\n",al.reg,al.nom,al.cta,al.tel,al.sem);
            n++;
            if (n==21) { pausa();
                for(i=7;i<=21;i++){
                    gotoxy(1,ib);stri(80,' ');
                    n=7;gotoxy(1,7);
                }
            }
        }
    }
}; /*if*/
} /*while*/
pausa();
} /*else*/
goto consultas;

case '6': strcpy(rr,"CATALOGOS IMPRESOS");
encabezado(rr);
do {
    gotoxy(1,7);
    cputs("1) POR REGISTRO\r\n");
    cputs("2) POR CUENTA\r\n");
    cputs("3) POR NOMBRE\r\n");
    cputs("4) POR SEMESTRE/CUENTA\r\n");
    cputs("5) POR SEMESTRE/NOMBRE\r\n");
    cputs(" OPCION ==>");rc=getche();
    orden=rc-48;if (orden<0) orden=0;
}while (orden<0 || orden>5);
if (orden==0) break;
encabezado(rr);
do {

```

```

gotoxy(1,23);
cputs("ESTA LISTA SU IMPRESORA (S o N)?");
rc=getche();rc=upcase(rc);
}while (rc!='N' && rc!='S');
if (rc=='N') break; /* SALIDA DE CATALOGOS */

lc=0;pag=0;
gotoxy(1,23);cputs("IMPRESION EN PROCESO ... ");

lar_buffer=208;no_llave=orden-1;buf_llave[0]=0;
st=BTRV(c_primer,pos,sal,&lar_buffer,buf_llave,no_llave);

while (st==0) {
if (lc==0 || lc>=60) {
if (lc>=60) {fprintf(stdprn,"%c",12);lc=0;}
strcpy(r,"Programa Muestra, Trabaja Utilizando el Manejador de Bases de Datos de BTRIEVE");
strip(40-strlen(r)/2,' ');fprintf(stdprn,"%s\n",r);
strcpy(r,"AGENDA DE ALUMNOS DE M.A.C., EN TURBO C:V2.0");
strip(40-strlen(r)/2,' ');fprintf(stdprn,"%s\n",r);
if (orden==1) strcpy(rr,"No. DE REGISTRO");
if (orden==2) strcpy(rr,"No. DE CUENTA");
if (orden==3) strcpy(rr,"NOMBRE");
if (orden==4) strcpy(rr,"No. DE SEMESTRE Y CUENTA");
if (orden==5) strcpy(rr,"No. DE SEMESTRE Y NOMBRE");
strcpy(r,"REPORTE ORDENADO POR ");
strcat(r,rr);
strip(40-strlen(r)/2,' ');fprintf(stdprn,"%s\n",r);
pag++;fprintf(stdprn,"PAG:[%3d]\n",pag);
strip(80,'-');fprintf(stdprn,"\n");
fprintf(stdprn,"REG CUENTA NOMBRE SEM GEN TELEFONO\n");
strip(80,'-');fprintf(stdprn,"\n");
lc=7;
}
fprintf(stdprn,"%5d %-10s %-37s %2d %-5s %-16sn" al.reg.al.cta.al.nom.al.sem.al.gen.al.tel);
lc++;
lar_buffer=208;
st=BTRV(c_sigui,pos,sal,&lar_buffer,buf_llave,no_llave);
}; /*while*/
if (lc>0) fprintf(stdprn,"%c",12);
break; /* SALIDA DE CATALOGOS */
} /* switch */

if (op!='7') goto menu; /* REGRESO AL MENU PRINCIPAL*/

/* FIN DEL PROGRAMA */
st = BTRV(cerrar,pos,nada,&cero,nada,0);
if (st!=0) error_exit();
}

```

```
/* RUTINAS COMPLEMENTARIAS */
```

```
error_exit()
```

```
{
gotoxy(1,24);
printf("%s%4d%s", "ERROR EN BTRIEVE No. ", st, " PULSE ENTER PARA CONTINUAR.");
putch(7);
rc=getch();
if (st!=20) st = BTRV(restau,nada,nada,&cero,nada,0);
exit(0);
}
```

```
encabezado(rr)
```

```
char rr[256];
{
clrscr();
centra(1,"PROGRAMA MUESTRA. TRABAJA UTILIZANDO EL MANEJADOR DE BASES DE DATOS DE BTRIEVE");
centra(2,"AGENDA DE ALUMNOS DE M.A.C., EN TURBO C V2.0");
gotoxy(1,3);stri(80,'-');
color(0,7);centra(3,rr);color(7,0);
}
```

```
color(x,y)
```

```
char x,y;
{
textcolor(x);
textbackground(y);
}
```

```
centra(y,r)
```

```
char y,r[256];
{
if (y<=0) y=wherey();
gotoxy(40-stlen(r)/2,y);
cprintf("%s",r);
}
```

```
stri(n,c)
```

```
char n,c;
{
register int i;
for(i=1;i<=n;i++,putch(c));
}
```

```
strip(n,c)
```

```
char n,c;
{
register int i;
for(i=1;i<=n;i++,fprintf(stdprn,"%c",c));
}
```

```
pone_preg()
```

```
{
gotoxy(1,6);
cprintf("[1] CUENTA : [%10s] \n \r", "");
cprintf("[2] NOMBRE : [%40s] \n \r", "");
cprintf("[3] DIRECCION : [%40s] \n \r", "");
cprintf("[4] COLONIA : [%35s] \n \r", "");
cprintf("[5] DEL O CIU : [%25s] \n \r", "");
cprintf("[6] ESTADO : [%25s] \n \r", "");
cprintf("[7] TELEFONOS : [%16s] \n \r", "");
cprintf("[8] GENERACION: [%05s] \n \r", "");
cprintf("[9] SEMESTRE : [%02s] \n \r", "");
}
```

```

uppercase(c)
char c;
{
if (c>='a' && c<='z') c=c-32;
return c;
}

pone_datos()
{
gotoxy(17, 6);puts(al.cta);
gotoxy(17, 7);puts(al.nom);
gotoxy(17, 8);puts(al.dir);
gotoxy(17, 9);puts(al.col);
gotoxy(17,10);puts(al.del);
gotoxy(17,11);puts(al.est);
gotoxy(17,12);puts(al.tel);
gotoxy(17,13);puts(al.gen);
gotoxy(17,14);cprintf("%2d",al.sem);
gotoxy(60, 4);cprintf("REGISTRO: [%5d]",al.reg);
}

pausa()
{
gotoxy(1,23);
cputs("PULSE [ENTER] PARA CONTINUAR ...");
rc=getch();
gotoxy(1,23);
stri(80,' ');
}

```

5.05 INTERFAZ CON TURBO PASCAL VERSIÓN 5.0 Y EL PROGRAMA EJEMPLO CODIFICADO.

La interfaz con Turbo PASCAL v5.0, es mucho más directa que la descrita con BASIC. En este caso la interfaz se implementa como una "unidad", (como la unidad CTR, por ejemplo) , el nombre del archivo que contiene la interfaz es TUR5BTRV.TPU, el código fuente de la unidad es TUR5BTRV.PAS. A diferencia de las interfaces para BASIC, en este caso el lenguaje permite desarrollar completamente la interfaz con instrucciones propias de TURBO PASCAL, es decir, que la interfaz TUR5BTRV.PAS forma parte de la aplicación, por medio de la instrucción:

uses TUR5BTRV;

Existe una consideración importante en este caso; para definir cadenas en el Turbo PASCAL, se requiere de un byte de control que se encuentra al inicio, y es el que determina el número de caracteres que contiene la cadena. El código del programa muestra, coloca dentro del archivo Btrieve, el caracter de control para cada una de las cadenas del registro, por lo tanto el largo lógico y físico de éste se modifica en comparación con la definición del archivo para BASIC. Originalmente el tamaño del registro es de 200 bytes, en el caso de este ejemplo el tamaño es de 208, puesto que en el registro existen 8 cadenas. Btrieve prevee este "problema", para lo cual define un tipo cadena especial que es el L-CADENA. Todas estas consideraciones deben ser reflejadas en el archivo Btrieve previsto para BASIC.

A continuación se presentan las modificaciones del archivo Btrieve ALUMNOS.BTR. El nombre del archivo Btrieve que requiere el código desarrollado para TURBO PASCAL se llama ALUM_PAS.BTR.

Archivo: ALUM_PAS.DES

```
record=208 variable=n key=5 page=1024 replace=y data=y
position=1 length=2 duplicates=n modifiable=y type=aut
descending=n alternate=n manual=n null=n segment=n
position=3 length=11 duplicates=n modifiable=y type=lst
descending=n alternate=n manual=n null=n segment=n
position=14 length=41 duplicates=n modifiable=y type=lst
descending=n alternate=n manual=n null=n segment=n
position=207 length=2 duplicates=y modifiable=y type=int
descending=n alternate=n manual=n null=n segment=y
position=3 length=11 duplicates=y modifiable=y type=lst
descending=n alternate=n manual=n null=n segment=n
position=207 length=2 duplicates=y modifiable=y type=int
descending=n alternate=n manual=n null=n segment=y
position=14 length=41 duplicates=y modifiable=y type=lst
descending=n alternate=n manual=n null=n segment=n
```

Se debe escribir el contenido de la tabla dentro del archivo de descripción, con el nombre ALUM_PAS.DES.

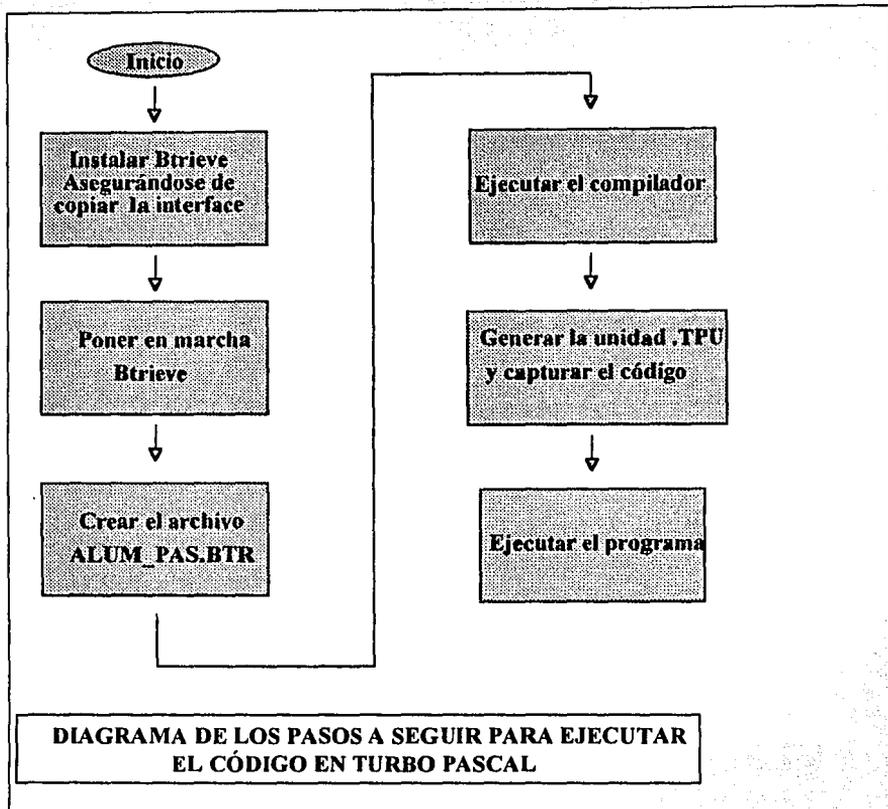
Sólo resta generar el archivo Btrieve y esto se debe realizar con la siguiente instrucción:

```
BUTIL -CREATE ALUM_PAS.BTR ALUM_PAS.DES ↵
```

Para poder ejecutar el programa se deben seguir los siguientes pasos:

1. **Instalar Btrieve (Ver. subcapítulo 2.04 ó 2.05, según el caso). Asegurando que la interfaz para el compilador sea copiada (archivo:TUR5BTRV.PAS)**
2. **Poner en marcha a Btrieve (Ver subcapítulo 2.04 ó 2.05, según el caso). Como el archivo Btrieve que se define en el paso 3, no tiene especificaciones demandantes, se puede poner en marcha Btrieve sin indicar ninguna opción especial. Ejemplo: BTRIEVE↵**
3. **Crear el archivo ALUM_PAS.BTR como se indicó anteriormente**
BUTIL -CREATE ALUM_PAS.BTR ALUM_PAS.DES ↵
4. **Ejecutar el compilador:**
TURBO ↵
5. **Generar la unidad TUR5BTRV.TPU**
6. **Capturar el código que se presenta a continuación y correr el programa:**

A continuación se presenta el diagrama de los pasos a seguir y el código completo del programa ejemplo:



```

{ El archivo TUR5BTRV.TPU es la interfaz con Btrieve.
  Este programa requiere en el mismo directorio el
  archivo ALUM_PAS.BTR.
}

program btrv_PAS (input, output);

uses crt, tur5btrv, printer;

label
  fin_altas, fin_consulta, fin_bajas, fin_cambios;

const
  abrir    =0; cerrar =1; insert =2; restau  =28; consulti =5; borrar  =4;
  actualiz=3; c_mayor=8; c_sigui=6; c_primer=12; bloqueo=200; desbloq=27;

var

{ El tamaño real de una cadena en Turbo pascal es su dimensión + uno. Ya que
  ocupa un byte para contar el número de caracteres en la cadena. Este byte de
  control será grabado junto con los datos en el archivo.}

  al : record
    reg: integer;
    cta: string[10];
    nom: string[40];
    dir: string[40];
    col: string[35];
    del: string[25];
    est: string[25];
    tel: string[16];
    gen: string[05];
    sem: integer;
end;

{El largo total del registro es de 208}

archivo, r, rr: string;
tit: array[1..2] of string;

op, rc: char;
cero: integer;

buf_llave: string[40];
llave34: record
  sem: integer;
  lla: string[40];
end;

i, n, k, j, sem, orden, lr, lc, pag, cc: integer;
ib, nb, jb: byte;
camb: boolean;

no_llave, st, lar_buffer: integer;
nada: string;

pos: array[1..128] of char;

function stri(n: byte; c: char): string;
var
  i: integer;
  cad: string;
begin
  cad:='';
  for i:=1 to n do cad:=cad+c;

```

```

stri:=cad;
end;
procedure pausa;
begin
gotoxy(1,23);
write('PULSE [ENTER] PARA CONTINUAR ...');
readln;
gotoxy(1,23);write(stri(80,' '));
end;

procedure error_halt(st:integer);
begin
gotoxy(1,24);
write(#7,'ERROR DE BTRIEVE No. ',st:4,' PULSE ENTER PARA CONTINUAR. ');
readln;
if st<>20 then st:=BTRV(restau,nada,nada,cero,nada,0);
halt(0);
end;

procedure centra(y:integer;r:string);
begin
if y<=0 then y:=wherey;
gotoxy(40-length(r) div 2,y);
write(r);
end;

procedure color(x,y:byte);
begin
textcolor(x);
textbackground(y);
end;

procedure encabezado(rr:string);
begin
clrscr;
centra(1,'PROGRAMA MUESTRA, TRABAJA UTILIZANDO EL MANEJADOR DE BASES DE DATOS DE BTRIEVE);
centra(2,'AGENDA DE ALUMNOS DE M.A.C., EN TURBO PASCAL V5.0');
gotoxy(1,3);write(stri(80,'-'));
color(0,7);centra(3,rr);color(7,0);
end;

procedure pone_preg;
begin
gotoxy(1,6);
writeln(' [1] CUENTA      :[' ,stri(10,' '),']');
writeln(' [2] NOMBRE     :[' ,stri(40,' '),']');
writeln(' [3] DIRECCION  :[' ,stri(40,' '),']');
writeln(' [4] COLONIA   :[' ,stri(35,' '),']');
writeln(' [5] DEL O CIU :[' ,stri(25,' '),']');
writeln(' [6] ESTADO    :[' ,stri(25,' '),']');
writeln(' [7] TELEFONOS :[' ,stri(16,' '),']');
writeln(' [8] GENERACION: [' ,stri(05,' '),']');
writeln(' [9] SEMESTRE  :[' ,stri(02,' '),']');
end;

procedure pone_datos;
begin
gotoxy(17,6);writeln(al.cta);
gotoxy(17,7);writeln(al.nom);
gotoxy(17,8);writeln(al.dir);
gotoxy(17,9);writeln(al.col);
gotoxy(17,10);writeln(al.del);
gotoxy(17,11);writeln(al.est);
gotoxy(17,12);writeln(al.tel);
gotoxy(17,13);writeln(al.gen);
gotoxy(17,14);writeln(al.sem);

```

```

gotoxy(60, 4);writeln('REGISTRO: [' ,al.reg:5, ']');
end;
(***** inicio del programa principal *****)
begin
ti[1]:=REG CUENTA NOMBRE DEL ALUMNO TELEFONO SEM;
ti[2]:=REG NOMBRE DEL ALUMNO CUENTA TELEFONO SEM;
nada:='';

(Apertura del archivo)
clrscr;
lar_buffer:=208;
no_llave:=0;
archivo:='ALUM.PAS.BTR ';
st:=BTRV(abrir,pos,nada,lar_buffer,archivo[1],0);
if st<>0 then error_halt(st);

(menu)
repeat
encabezado('MENU');
repeat
gotoxy(1,23);
color(0,7);
write('1)ALTAS 2)BAJAS 3)CAMBIOS 4)CONSULTAS 5)CONS.X SEM 6)CATALOGO 7)FIN =>');
color(7,0);
readln(op);
until (op>='1') and (op<='7');

case op of
(***** A L T A S *****)
'1':begin
repeat
rc:='x';
encabezado('ALTAS');
pone_preg;

repeat
gotoxy(17, 6);readln(al.cta);if al.cta='' then goto fin_altas;
until length(al.cta)>=9;
repeat
gotoxy(17, 7);readln(al.nom);if al.nom='' then goto fin_altas;
until length(al.nom)>10;
gotoxy(17, 8);readln(al.dir);
gotoxy(17, 9);readln(al.col);
gotoxy(17,10);readln(al.del);
gotoxy(17,11);readln(al.est);
gotoxy(17,12);readln(al.tel);
gotoxy(17,13);readln(al.gen);

repeat
gotoxy(17,14);readln(r);val(r,al.sem,i);
until (al.sem>=1) and (al.sem<=15);
repeat
gotoxy(1,23);
write('DESEA DAR DE ALTA ESTOS DATOS (S O N)?');
readln(rc);rc:=upcase(rc);
until (rc='N') or (rc='S');
if rc='S' then
begin
al.reg:=0;
lar_buffer:=208;
no_llave:=0;
st:=BTRV(insert,pos,al,lar_buffer,al.reg,no_llave);
if (st=5) then
begin
gotoxy(1,23);

```

```

write('EL NOMBRE Y/O LA CUENTA YA EXISTEN. ALTA INVALIDA...','#7);
readln;
end else
  if st<>0 then error_halt(st);
end; (if)
fin altas;;
until (rc='x');
end;
{***** B A J A S *****}
'2':begin
  repeat (bajas)
  repeat (pregunta)
  rc='x';
  encabezado('BAJAS');
  pone_preg;
  gotoxy(1,5);write('NOMBRE, CTA o REG:[' ,STRI(40,' '),']');
  gotoxy(20,5);
  readln(r);
  lr=length(r);
  until lr<=40;
  if r='' then goto fin_bajas;
  no_llave:=0;
  if lr<=5 then begin
    no_llave:=0;
    val(r,i,n);
    move(i,r,2);
  end else
    if lr<=10 then no_llave:=1
    else
      no_llave:=2;
    end;
  end;
  lar_buffer:=208;
  st:=BTRV(consulti+bloqueo,pos,al,lar_buffer,r,no_llave);
  r='';
  if (st=84) then
  begin
    gotoxy(1,23);
    write('REGISTRO BLOQUEADO POR OTRA COMPUTADORA DE LA RED. Pulse [Enter].'#7);
    readln;rc=' ';
  end else
  if (st=4) then
  begin
    gotoxy(1,23);
    write('NO SE ENCONTRO EL ALUMNO, PULSE [ENTER] P. CONT.' ,#7);
    readln;rc=' ';
  end else
    if st<>0 then error_halt(st)
  end;
else begin
  pone_datos;
  rc='S';
  repeat
  gotoxy(1,23);
  write('DESEA DARLO DE BAJA (S o N)?');
  readln(rc);rc:=upcase(rc);
  until (rc='N') or (rc='S');
end;
if rc='N' then
begin
  lar_buffer:=208;
  st:=BTRV(desbloq,pos,al,lar_buffer,nada,no_llave);
end
else st:=BTRV(borrar,pos,al,cero,nada,cero);
end;
fin bajas;
until rc='x';
end; (bajas)

```

```

(***** C A M B I O S *****)
'3':begin
  repeat (consultas)
  camb:=false;
  repeat (pregunta)
  rc:='x';
  encabezado('CAMBIOS');
  pone_preg;
  gotoxy(1,5);write('NOMBRE, CTA o REG:[',STRI(40,' '),']');
  gotoxy(20,5);
  readln(r);
  lr:=length(r);
  until lr<=40;
  if r='' then goto fin_cambios;
  buf_llave:=r;
  no_llave:=0;
  if lr<=5 then begin
    no_llave:=0;
    val(buf_llave,1,n);
    move(i,buf_llave,2);
  end else
    if lr<=10 then no_llave:=1
    else
      no_llave:=2;

  lar_buffer:=208;
  st:=BTRV(consulti+bloqueo,pos,al,lar_buffer,buf_llave,no_llave);

  if (st=4) then
  begin
    gotoxy(1,23);
    write('NO SE ENCONTRO EL ALUMNO, PULSE [ENTER] P. CONT.',#7);
    readln(rc):=' ';
  end else
    if st<>0 then error_halt(st)
  else begin
    pone_datos;
    repeat
    repeat
    gotoxy(1,23);
    write('ESTA CORRECTA LA INFORMACION (S o N)?');
    readln(rc);rc:=upcase(rc);
    until (rc='N') or (rc='S');
    if rc='N' then
    begin
      repeat
      repeat
      gotoxy(1,23);write(stri(79,' '));
      gotoxy(1,23);
      write('No. DE DATO A MODIFICAR:');readln(r);
      val(r,cc,1);
      until (cc>=0) and (cc<=9);
      if cc>0 then camb:=true;
      case cc of
      1:begin
        repeat
        gotoxy(17, 6);write(stri(10,' '));
        gotoxy(17, 6);readln(al.cta);
        until length(al.cta)>=9;
        end;
      2:begin
        repeat
        gotoxy(17, 7);write(stri(40,' '));
        gotoxy(17, 7);readln(al.nom);
        until length(al.nom)>10;

```

```

end;
3:begin
  gotoxy(17, 8);write(stri(40,' '));
  gotoxy(17, 8);readln(al.dir);
end;
4:begin
  gotoxy(17, 9);write(stri(35,' '));
  gotoxy(17, 9);readln(al.col);
end;
5:begin
  gotoxy(17,10);write(stri(25,' '));
  gotoxy(17,10);readln(al.del);
end;
6:begin
  gotoxy(17,11);write(stri(25,' '));
  gotoxy(17,11);readln(al.est);
end;
7:begin
  gotoxy(17,12);write(stri(16,' '));
  gotoxy(17,12);readln(al.tel);
end;
8:begin
  gotoxy(17,13);write(stri(15,' '));
  gotoxy(17,13);readln(al.gen);
end;
9:begin
  repeat
    gotoxy(17,14);write(stri(2,' '));
    gotoxy(17,14);readln(r);val(r,al.sem,i);
  until (al.sem>=1) and (al.sem<=15);
end;
end; (case)
until cc=0;
end;
until rc='S';
end;

fin cambios;
until (rc='S') or (rc='X');
if camb then
  begin { ACTUALIZA EL REGISTRO }
    lar_buffer:=208;
    st:=BTRV(actualiz,pos,al,lar_buffer,buf_llave,no_llave);
    if st<>0 then error_halt(st);
  end else
    begin
      lar_buffer:=208;
      st:=BTRV(desbloq,pos,al,lar_buffer,nada,no_llave);
    end;
end; (cambios)

{***** CONSULTAS *****}
'4':begin
  repeat (consultas)
  repeat (pregunta)
  rc:='X';
  encabezado('CONSULTAS');
  pone_preg;
  gotoxy(1,5);write('NOMBRE, CTA o REG:[' ,STRI(40,' '),']');
  gotoxy(20,5);
  readln(r);
  lr:=length(r);
  until lr<=40;
  if r='' then goto fin_consulta;
  no_llave:=0;
  if lr<=5 then begin
    no_llave:=0;

```

```

        val(r,i,n);
        move(i,r,2);
    end else
        if lr<=10 then no_llave:=1
            else
                no_llave:=2;
        end

    lar_buffer:=208;
    st:=BTRV(consulti,pos,al,lar_buffer,r,no_llave);
    r:='';
    if (st=4) then
        begin
            gotoxy(1,23);
            write('NO SE ENCONTRO EL ALUMNO, PULSE [ENTER] P. CONT.',#7);
            readln;
            rc:=' ';
        end else
            if st<0 then error_halt(st)
        else begin
            pone_datos;
            rc:='S';
            repeat
                gotoxy(1,23);
                write('DESEA SEGUIR CONSULTANDO (S o N)?');
                readln(rc);rc:=upcase(rc);
            until (rc='N') or (rc='S');
        end;
        fin_consulta;

        until (rc='N') or (rc='x');
    end; {consultas}

    { ***** CONSULTA POR SEM ***** }
'5':begin
    repeat
        rr:='CONSULTA EN PANTALLA POR SEMESTRE/CUENTA o SEMESTRE/NOMBRE';
        encabezado(rr);
        repeat
            gotoxy(1,10);write('INDIQUE EL SEMESTRE A CONSULTAR:');
            readln(r);

            val(r,sem,i);
            until (r='') or ((sem>1) and (sem<15));
            if r<>' ' then
                begin
                    repeat
                        gotoxy(1,11);
                        write('1]CUENTA o 2]NOMBRE (1 o 2)?');
                        readln(r);
                        val(r,orden,i);
                    until (orden=1) or (orden=2);
                    encabezado(rr);
                    gotoxy(1,5);
                    writeln(tit[orden]);
                    writeln(stri(80,'-'));

                    llave34.sem:=sem;llave34.lla:='';
                    lar_buffer:=208;no_llave:=2+orden;
                    st:=BTRV(c_mayor,pos,al,lar_buffer,llave34,no_llave);

                    if (st=9) or (st=4) or (al.sem<>sem) then
                        begin
                            gotoxy(1,23);
                            write('NO EXISTEN ALUMNOS EN EL SEMESTRE INDICADO ...',#7);
                            readln;
                        end
                    else

```

```

begin
if st<>0 then error Halt(st);
if orden=1 then writeln(al.reg:5,',al.cta:10,',al.nom:40,',al.tel:16,',al.sem)
else writeln(al.reg:5,',al.nom:40,',al.cta:10,',al.tel:16,',al.sem);
n:=7;

while (st=0) and (al.sem=sem) do
begin
lar_buffer:=208;
st:=BTRV(c_sigui,pos,al,lar_buffer,llave34,no_llave);
if (st<>0) and (st<>9) and (st<4) then error Halt(st);

if ((st=0) and (al.sem=sem)) then
begin
if orden=1 then writeln(al.reg:5,',al.cta:10,',al.nom:40,',al.tel:16,',al.sem)
else writeln(al.reg:5,',al.nom:40,',al.cta:10,',al.tel:16,',al.sem);
n:=n+1;if n=21 then begin
pausa;
for ib:=7 to 21 do begin
gotoxy(1,ib);write(stri(80,' '));
n:=7;gotoxy(1,7);
end;

end;

end (if)
end; pausa; (while)
r:='x';
end; (else)
end; (if)
until r='';
end; (case)

(***** CATALOGOS IMPRESOS *****)
'6':begin
rr:='CATALOGOS IMPRESOS';
encabezado(rr);
repeat
gotoxy(1,7);
writeln('1) POR REGISTRO');
writeln('2) POR CUENTA');
writeln('3) POR NOMBRE');
writeln('4) POR SEMESTRE/CUENTA');
writeln('5) POR SEMESTRE/NOMBRE');
write(' OPCION ==>');readln(r);
val(r,orden,i);
until (orden>=0) and (orden<=5);
if orden<>0 then
begin
encabezado(rr);
repeat
gotoxy(1,23);
write('ESTA LISTA SU IMPRESORA (S o N)?');
readln(rc);rc:=supcase(rc);
until (rc='N') or (rc='S');
if rc='S' then
begin
lc:=0;pag:=0;
gotoxy(1,23);write('IMPRESION EN PROCESO ... ');

lar_buffer:=208;no_llave:=orden-1;r:=stri(40,' ');
st:=BTRV(c_primer,pos,al,lar_buffer,r,no_llave);

while st=0 do
begin
if (lc=0) or (lc>=60) then

```

```

begin
  if lc>=60 then
    begin
      writeln(1st, #12); lc:=0;
    end;
    r:= 'Programa Muestra, Trabaja Utilizando el Manejador de Bases de Datos de BTRIEVE';
    writeln(1st, stri(40-length(r) div 2, ' '), r);
    r:= 'AGENDA DE ALUMNOS DE M.A.C., EN TURBO PASCAL VS.0';
    writeln(1st, stri(40-length(r) div 2, ' '), r);
    if orden=1 then rr:='No. DE REGISTRO';
    if orden=2 then rr:='No. DE CUENTA';
    if orden=3 then rr:='NOMBRE';
    if orden=4 then rr:='No. DE SEMESTRE Y CUENTA';
    if orden=5 then rr:='No. DE SEMESTRE Y NOMBRE';
    rr:='REPORTE ORDENADO POR '+rr;
    writeln(1st, stri(40-length(rr) div 2, ' '), rr);
    pag:=pag+1; writeln(1st, 'PAG: [' , pag:3, ']');
    writeln(1st, stri(80, '-'));
    writeln(1st, 'REG CUENTA      NOMBRE                SEM GEN TELEFONO');
    writeln(1st, stri(80, '-'));
    LC:=7;
  end; (if)
  writeln(1st, al.reg:5, ', al.cta:10, ', al.nom:37, ', al.sem:2, ', al.gen:5, ', al.tel:16);
  lc:=lc+1;
  lar_buffer:=208;
  st:=BTRV(c_sigui, pos, al, lar_buffer, r, no_llave);
end; (while)
  if lc>0 then writeln(1st, #12);
end; (esta lista su imp.)
  end; (if)
end; (catalogo)
end; (case)
until (op='7');

{***** FIN DEL PROGRAMA *****}

st:=BTRV(cerrar, pos, nada, cero, nada, 0);
clrscr;
end.

```

CAPÍTULO VI

OTRAS MODALIDADES

6.1. POSIBILIDAD DEL USO DE BTRIEVE EN LENGUAJES ORIENTADOS A OBJETOS.

Este trabajo se orienta al manejo de bases de datos por medio de computadoras personales independientes o redes de computadoras con muy pocos requerimientos de Software y Hardware, pero Btrieve también puede trabajar sobre el ambiente que proporciona Microsoft Windows. Para este efecto existen dos modalidades: la modalidad *Basado en Cliente* para Windows y la Modalidad descrita en este trabajo de *Basado en Servidor* con la interface para Windows.

El trabajar con ambiente Windows requiere de computadoras más actuales y con mayor capacidad, aspectos que no siempre es posible cumplir, por esta razón de decidí orientar el trabajo sobre la versión de Btrieve que corre en casi cualquier computadora personal. En este capítulo se describe el proceso de instalación de la interface de Btrieve para Windows y se resaltan las ventajas de este ambiente.

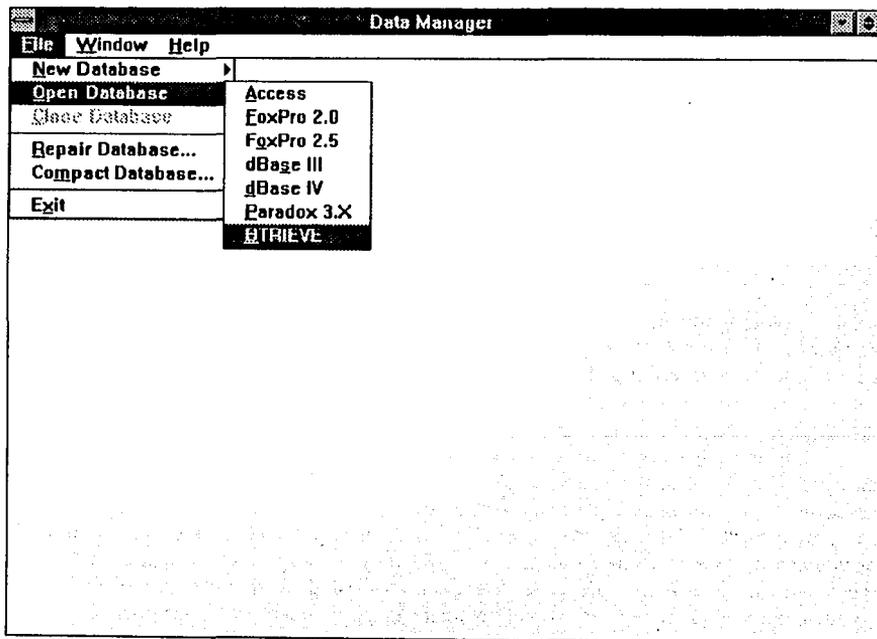
Los lenguajes que se pueden adquirir comercialmente para desarrollar aplicaciones sobre Windows son por ejemplo Visual Basic, Visual C, estos proveen de herramientas excelentes para el manejo de la pantalla, bases de datos y dispositivos instalados sobre Windows que permiten realizar fácilmente aplicaciones de Multimedia, cosas que en los lenguajes tradicionales para DOS requieren de demasiado tiempo y código.

Considerando que Btrieve está definido como una de las herramientas para el manejo de datos (ver cuadro 6.01) y los lenguajes sobre Windows lo presentan como una solución a este problema, se demuestra el buen nivel y la calidad del programa en estudio.

Los lenguajes orientados a objetos proveen herramientas para el manejo de pantalla, bases de datos y dispositivos, al trabajar estos en combinación obtenemos como resultado un ambiente de trabajo óptimo para el nivel de desarrollo sobre el cual se enfoca este trabajo (programación con lenguajes de computadora), siempre y cuando nos encontremos en la posibilidad de cubrir los requerimientos de hardware y software, estos lenguajes son lo más recomendable, puesto que presentan una gran variedad de herramientas para facilitar la codificación y el diseño y casi no tienen limitaciones en cuanto a la manipulación de la computadora moderna.

De la misma forma en que existen interfaces de Btrieve para los lenguajes comunes de computadora que trabajan sobre DOS, existe una versión en la modalidad *basado en cliente* para Windows y la versión *basado en servidor* para Windows, la provee Novell en su sistema operativo Novell NetWare. Esta versión de Btrieve ofrece una interface para Windows por medio de la cual cualquier lenguaje para Windows puede acceder a las herramientas descritas a lo largo de este trabajo, pero además existe una posibilidad interesante que bien vale la pena describir en este capítulo, es la forma en que Visual Basic Versión 3.0 permite acceder archivos de Btrieve.

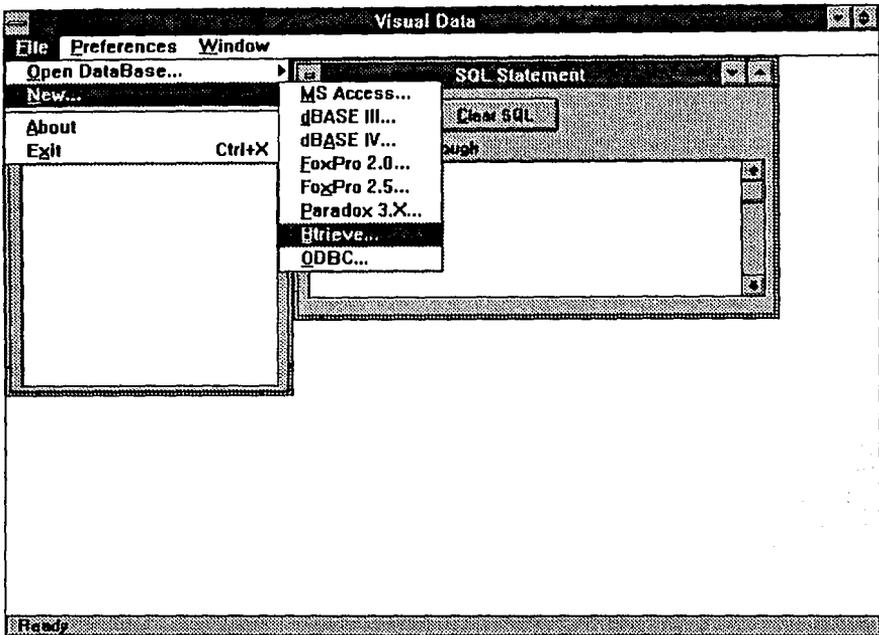
Después de haber instalado la interface de Btrieve para Windows, Visual Basic V3.0 cuenta con la posibilidad de acceder y dar mantenimiento a archivos de Btrieve del formato 5.1x, (ver cuadro 6.02) lo único especial que se requiere son los archivos de definición de cada tabla, que son FILE.DDF y FIELD.DDF (éstos no son necesarios para la versión de Btrieve para DOS), los cuales tienen toda la información de la base de datos como, el número de archivos, los campos de cada uno de los archivos y los campos índice, estos archivos pueden ser creados por medio de un programa que provee Visual Basic (ver cuadro 6.02) o bien por manejadores de bases de datos que utilizan como base a Btrieve para trabajar como Xtrieve®.



Cuadro 6.01
Uso de Btrieve en Visual Basic

En Realidad Visual Basic permite dar mantenimiento a diversos tipos de archivos entre los cuales están:

- Microsoft Access
- Btrieve
- dBASE III
- dBASE IV
- FoxPro Versión 2.0
- FoxPro Versión 2.5
- ODBC (SQL Server, Oracle)
- Paradox



Cuadro 6.02
Mantenimiento de Archivos Btrieve en Visual Basic Ver. 3.0.

Visual Basic puede trabajar con varios formatos de archivo ya que provee de un grupo de instrucciones para el manejo de bases de datos que son las mismas independientemente de el tipo de archivo, la diferencia radica en la información indicada en el momento de abrir la base de datos.

A continuación se presenta la lista de las herramientas generales más importantes que provee Visual Basic para el manejo de bases de datos:

Sobre el manejo del objeto *Database*. Que se refiere al grupo de archivos que conforman una base de datos

Herramientas:

Begin Trans, Close, Commit Trans, CreateDynaset, CreateQueryDef, CreateSnapshot, DeleteCueryDef, Execute, ExecuteSQL, ListFields, ListTables, OpenQueryDef, OpenTable, Rollback.

Propiedades:

CollatingOrder, Connect, Name, QueryTimeout, Transactions, Updatable.

Sobre el manejo de los objetos *Dynaset*, *Table*, *Snapshot*. Que se refieren a una colección de registros, *Table* por ejemplo es el objeto que contiene la totalidad de los registros de un archivo pero *Snapshot* es un conjunto de registros asociados a una consulta o condición.

Herramientas:

AddNew, Clone, Close, CreateDynaset, CreateSnapshot, Delete, Edit, FindFirst, Findlast, findNext, findPrevious, ListFields, ListIndexes, MoveFirst, MoveLast, MoveNext, MovePrevious, Seek, Update.

Propiedades:

BOF, Bookmark, Bookmarkable, DateCreated, EOF, Filter, Index, LastModified, LastUpdated, LockEdits, Name, NoMatch, RecordCount, Sort, Transactions, Updatable.

Como se puede apreciar el grupo de herramientas que provee Visual Basic es en cierta medida semejante a las herramientas que provee Btrieve, la gran diferencia radica en que estas herramientas son generales, independientemente del tipo de archivo y la gama de posibilidades es más completa ya que el trabajar con objetos es más claro y detallado, además de contar con consultas por medio de Microsoft Access, SQL y también no se trabajó con un solo archivo, sino con una base de datos.

A continuación se presenta un ejemplo del código en Visual Basic que permite abrir una base de datos de Btrieve:

```
Dim BaseD As Database
Dim Archivo As Table
Dim conec As String
conec$ = "Btrieve;"

Set BaseD = OpenDatabase("C:\BTRIEVE\FILE.DDF", False, False, conec$)
Set Archivo = BaseD.OpenTable("ALUMNOS")
```

Después de que Visual Basic ejecuta estas instrucciones, se puede acceder y dar mantenimiento al archivo ALUMNOS que se encuentra en el formato de Btrieve.

Este capítulo no pretende explicar completamente el uso de Visual Basic, únicamente dar un panorama general de las posibilidades del manejo de bases de datos y en particular de Btrieve en lenguajes orientados a Windows.

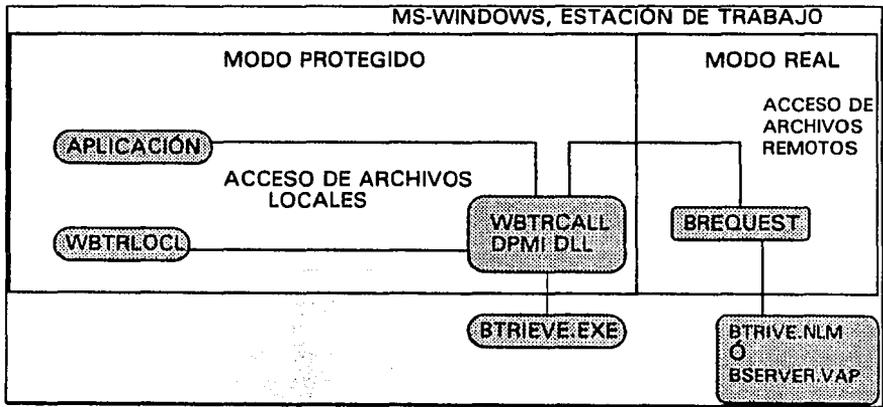
Al igual que en el caso de Btrieve para DOS, las aplicaciones desarrolladas para Windows tanto en la modalidad de basado en cliente como basado en servidor, no requieren de modificaciones para trabajar en cualquiera de ellas, únicamente varían algunos detalles en la instalación.

- INSTALACIÓN DE LA INTERFACE PARA Microsoft Windows.

Para poder acceder a Btrieve desde Windows, primero se debe poner en marcha Btrieve antes de entrar a Windows. Una aplicación desarrollada sobre Windows puede acceder a Btrieve por medio de una librería tipo DLL (Dinamically Linked Library), que utiliza la interface de modo de protección de DOS (DPMI), que provee Windows.

Tanto para la modalidad basado en cliente como la basada en servidor, la función que provee la librería DLL, se reconoce como WBTRCALL para archivos del servidor, y para archivos locales es WBTRLOCL.

A continuación se presenta un diagrama de como se realiza la comunicación entre una aplicación de Windows y Btrieve:



Las opciones de inicialización de la interface para Windows deben ser escritas en el archivo WIN.INI, adicionando las siguientes líneas con el valor que corresponda para cada parámetro:

```
[BREQUESTDPMI]
datalength=4096
tasks=10
local=no
chkparms=no
```

A continuación se describe el significado para cada opción.

Datalength	El valor para esta opción debe ser el mismo que el asignado en la opción /D utilizado en la interface BREQUEST, que se refiere al largo máximo para un registro.
Tasks	Define el número de tareas que puede manejar la librería DLL.
Local	Define si es factible utilizar archivos locales.
Chkparms	Indica que se deben verificar los punteros que pasan a la librería, se utiliza normalmente en la etapa de desarrollo.

En la modalidad de *Basado en cliente* para windows, las líneas para en WIN.INI es otra:

```
{btrieve}
options=/m:64/p:4096/b:16/f:20/l:40/n:12/t:C:\ACCESO\BTRIEVE.TRN
```

- COMO DESACTIVA A LA INTERFACE.

Antes de terminar la aplicación debe utilizar la función WBTRVSTOP(), esto es necesario para remover los trabajos de la lista dentro de la librería DLL. Si la función regresa un código de error diferente de cero indica que existen todavía algunos archivos abiertos o que alguna transacción esta activa, se debe permitir el cierre de archivos antes de ejecutar esta función, o bien, ejecutar la operación RESET antes de llamar a la función WSTRVSTOP().

6.02 OTRAS PLATAFORMAS EN LAS QUE BTRIEVE PUEDE SER UTILIZADO.

Ya que Novell NetWare permite conectar diferentes recursos a una red, es factible conectar estaciones de trabajo que utilizan sistemas operativos diferentes como OS/2. Tomando en cuenta esto, Btrieve provee también la interface y el ruteador para establecer comunicación con una aplicación desarrollada bajo OS/2.

El archivo que contiene la interface es BTRCALLS.DLL, éste debe ser instalado en la estación de trabajo que utiliza OS/2, dentro de uno de los directorios que se especifican en el comando LIBPATH en el CONFIG.SYS. Esta librería sólo permite acceder archivos remotos que se encuentran en el servidor de Novell. Sin embargo existe la versión de Btrieve para OS/2, que permite acceder archivos locales, donde se debe usar la utilidad MAKLOCAL.EXE sobre el prompt de OS/2, con el nombre del directorio donde el archivo BTRCALLS.DLL se encuentra, por ejemplo:

```
C:\>MAKLOCAL C:\DLLS
```

En este caso las opciones de inicialización de Btrieve se deben escribir sobre una variable de ambiente de OS/2 con el nombre de BRQPARMS, por ejemplo:

```
C:\>SET BRQPARMS= /R:4 /D:2048 /S:2
```

CONCLUSIONES

A lo largo de este trabajo se realizó la descripción del programa Btrieve, tanto en su instalación como en su uso, también se presentaron ejemplos claros que demuestran lo conveniente de utilizar las herramientas de manejo de registros multi-indexados que provee Btrieve.

Puesto que las instrucciones que tienen que ver con el manejo de archivos dentro de los códigos, son mínimas (en los ejemplos presentados son aproximadamente 24 instrucciones solamente), dió como resultado la optimización del tiempo en el momento de diseñar y codificar los programas utilizando esta herramienta, que fue específicamente el objetivo que se buscó al elaborar este trabajo.

Por lo tanto: al programar una aplicación utilizando las herramientas que provee Btrieve, para el manejo de bases de datos, se obtendrá mayor claridad en el código, rapidez al diseñar y codificar así como mayor seguridad en sus datos. El resolver el problema del manejo de bases de datos permite enfocar más la atención al problema real a resolver.

- Ventajas del uso de Btrieve.

El programador que utiliza Btrieve como manejador de registros, obtiene las siguientes ventajas:

- Evita el programar algoritmos de ordenamiento muy complicados.
- Adquiere un alto grado de seguridad sobre la información.
- Obtiene controles para el mantenimiento y uso compartido de información.
- Cuenta con la posibilidad de definir hasta 24 índices por archivo, no es necesario abrir un archivo por cada índice. Además, es factible definir índices externos.
- Puede utilizar una gran variedad de lenguajes conservando la compatibilidad en el uso de la información.
- Maneja la información en los archivos Btrieve de forma comprimida o encriptada, con el fin de optimizar espacio y obtener seguridad.

- Desventajas del uso de Btrieve.

- Una aplicación que utiliza esta herramienta pierde la transportabilidad del código.
- La necesidad de conseguir una nueva interface, si se decide utilizar otra versión de lenguaje que no esté incluida en el paquete de Btrieve.
- Los manuales que ofrece son muy técnicos y no documentan formalmente el uso de las interfaces para todos los lenguajes, sólo documentan 5 de las 35 interfaces que incluyen.

RECOMENDACIONES

A continuación se presentan algunas recomendaciones para el mejor aprovechamiento de este trabajo.

Si su intención es comprender y utilizar el manejador de registros indexados Btrieve, se recomienda realizar lo siguiente:

1. Leer completamente el glosario, en él se presentan diferentes términos junto con su definición o interpretación. Estas interpretaciones serán utilizadas a lo largo del trabajo.
2. Leer la introducción, índice, resumen y el capítulo I y VI, ahí se explica que es Btrieve, cuales son requerimientos de hardware y software, sobre que lenguajes puede trabajar, cuales son sus características y beneficios. Esto da los elementos de juicio para determinar si Btrieve es lo que se requiere para satisfacer el problema planteado, de ser así, debe considerar el adquirir Btrieve para contar con las interfaces y programas necesarios.

Recordar que Btrieve para DOS v 5.10a (basado en cliente) proporciona todo lo necesario (programas, interfaces, manual del programador e instalación), la modalidad basado en servidor se encuentra en los discos de Novell NetWare (sin interfaces, ni manual del programador).

3. Leer el capítulo II. En este capítulo se presentan diversos fundamentos y reglas, los cuales permiten conocer el ambiente que se crea al utilizar Btrieve, este capítulo es de consulta, no es indispensable memorizar o recordar el contenido para continuar con el trabajo. Dependiendo la modalidad de Btrieve con la que cuente, localice el subcapítulo que corresponde a su instalación, ya que se presentan tres posibilidades. Antes de llegar al capítulo V, debe completar la instalación del paquete. También es importante que revise en este momento los apéndices A y B.
4. Leer el capítulo III. En este capítulo se describen las operaciones con las que puede contar al hacer uso de Btrieve, es de mucha ayuda en el momento de realizar algún programa, puesto que se deben verificar los formatos de salida y entrada de las operaciones a utilizar cuando no se recuerdan. Este capítulo es de consulta, no es indispensable memorizar o recordar el contenido para continuar con el trabajo.
5. Leer el capítulo IV. En este capítulo se describen las consideraciones que debe contemplar un programa para que pueda manejar archivos en red. Si su intención es trabajar con archivos comunitarios, lea cuidadosamente todo el contenido del capítulo.

6. Leer el capítulo V. En éste se presentan 5 ejemplos de códigos. Estos programas realizan exactamente lo mismo, lea el subcapítulo 5.01 para conocer la apariencia y funcionamiento del programa muestra y posteriormente decida en que lenguaje se le facilitaría más comprender el programa, entre los siguientes:

- BASIC intérprete
- QuickBASIC V2.0
- QuickBASIC V4.5
- Turbo C V2.0
- Turbo PASCAL V5.0

Dependiendo de su elección siga los pasos que se encuentran en este capítulo para poder ejecutar el programa. Es muy importante que ponga en marcha uno o más de los programas que se incluyen, con el fin de que le sirva de práctica y pueda comprender mejor el funcionamiento de Btrieve, estos programas le pueden servir de base para crear mejores aplicaciones.

7. Si desea conocer más sobre como funcionan las interfaces con Btrieve, consulte el apéndice C.

Si usted únicamente desea ver como funciona el programa sin explicaciones tediosas, y prefiere iniciar con la práctica en vez de la teoría, intente comenzar desde el punto No. 6, que propone leer el capítulo V, sólo consulte las referencias importantes que se indican en capítulos anteriores.

BIBLIOGRAFÍA

**Btrieve for DOS V 5.10
PROGRAMMER'S MANUAL
Professional Development Series TM
DOS NetWork, Abril de 1990.**

**Btrieve for DOS
Professional Development Series TM
Installation and operation, Abril de 1990.**

**HandBook of Basic for the IBM PC
David I. Schneider
Brady Communications Company, Inc., 1985.**

**Microsoft Quick Basic Compiler for IBM personal Computers and compatibles V2.0
Microsoft Corporation, 1986.**

**Using Turbo Pascal
Michael Yester
QUE Corporation, 1989.**

**Using Novell Netware
Bill Lawrance
QUE Corporation, 1990.**

**QuickC Programming for the IBM
Carl Townsend
Howard W. Sams & Company, 1988.**

**Microsoft QuickC.
C for yourself V.2.5
Microsoft Corporation
Impreso en USA, 1990**

**Programación en TURBO C
HERBERT SCHILDT
BORLAND OSBORNE/McGRAW HILL
Impreso en España, 1988**

**THE NEW PETER NORTON PROGRAMMER'S GUIDE TO THE IBM PC & PS/2
Peter Norton / Richard Wilton
Microsoft PRESS
Impreso en Washington U.S.A., 1988**

THE WORLD'S FIRST 486
BYTE Volume 14, Number 9
McGRAW-HILL
Impreso en U.S.A, Septiembre de 1989

MicroSoft QuickBASIC (Versión 4.5)
Programming in BASIC
Impreso en USA, 1988.

MicroSoft QuickBASIC (Versión 4.5)
Learning to use
Impreso en USA, 1988.

APÉNDICE A

CÓDIGOS DE ERROR

A continuación se describen los códigos que Btrieve puede regresar si una operación no se realiza satisfactoriamente y al final una tabla con el resumen de los mismos:

01 OPERACIÓN INVÁLIDA.

El código especificado en la operación no es válido.

02 ERROR DE E/S.

Ha ocurrido un error durante la lectura o escritura en el disco. Este código indica que el archivo estuvo dañado y debe ser recreado o bien que el archivo especificado no ha sido creado. Este código puede presentarse si la aplicación pasa una posición de block inválida.

03 ARCHIVO NO ABIERTO.

La operación no puede ser ejecutada porque el archivo no está abierto. Una instrucción de *Abrir* (0) debe ser ejecutada antes de que Btrieve pueda realizar cualquier otra operación en el archivo. Este código también se presenta si la aplicación pasa una posición de block inválido.

04 VALOR DE LLAVE NO ENCONTRADO.

El valor de llave especificado no fue encontrado en el índice.

05 VALOR DE LLAVE DUPLICADO.

El registro tiene un valor de llave que ya existe en el índice y este campo de llave no acepta valores duplicados.

06 NÚMERO DE LLAVE INVÁLIDO.

El valor colocado en el parámetro de número de llave no es válido para que el archivo pueda ser accedido. El número de llave debe corresponder a la numeración dispuesta para llaves definidas cuando el archivo se crea o bien, al número de una llave suplementaria.

07 NÚMERO DE LLAVE DIFERENTE.

El parámetro de número de llave cambió después de una operación de *Consulta siguiente*, *Consulta el anterior*, *Actualizar*, o *Borrar*. La operación a ejecutar requiere del mismo número de llave que la operación anterior ya que Btrieve utiliza la información de la posición del número de llave anterior.

Si necesita cambiar el número de llaves entre operaciones consecutivas; *Consulta siguiente*, *Consulta el anterior*, use una operación de *Obtener posición (22)*, seguida por una operación de *Consulta directamente (23)* para reestablecer el posicionamiento para la nueva tabla indexada.

08 POSICIONAMIENTO INVÁLIDO.

La posición actual debe ser establecida para poder actualizar o borrar el registro, utilizando las operaciones de *Consulta* o de *Escalón* para establecer la posición. Este código también ocurre si la aplicación pasa una posición de block inválido.

09 FIN DEL ARCHIVO.

Alguna operación trata de leer más allá del límite del archivo (fin del archivo o inicio del archivo). Cuando se está leyendo un archivo en orden ascendente de acuerdo a un índice, Btrieve regresa el último registro de ese índice. Cuando se lee el archivo en orden descendente de acuerdo a un índice, Btrieve regresa el primer registro de ese índice.

10 ERROR AL MODIFICAR EL VALOR DE LLAVE.

La operación está tratando de modificar un campo de llave que fué definida como no modificable.

11 NOMBRE DE ARCHIVO INVÁLIDO.

El nombre especificado no cumple con los requisitos para el nombre de archivo (Contiene caracteres inválidos).

12 EL ARCHIVO NO SE ENCUENTRA.

El nombre del archivo especificado no existe. Verificar el parámetro de *Buffer de Llave* para asegurarse que el nombre del archivo termina con blancos o en cero binario.

14 ERROR DE APERTURA DEL ARCHIVO DE PRE-IMAGEN.

El archivo de pre-imagen no se pudo crear o abrir. Hay tres posibles causas para éste error:

- Btrieve no pudo crear el nuevo archivo de pre-imagen porque el directorio del disco está lleno. Btrieve no podrá crear el archivo de pre-imagen.
- Btrieve no pudo abrir el archivo pre-imagen para recobrar la integridad del archivo Btrieve. Si el archivo de pre-imagen está dañado o fue borrado, Btrieve no podrá restaurar la integridad de los archivos. En este caso, deberá usar el comando RECOVER de la utilidad BUTIL, para recobrar la información dañada de los archivos, otra alternativa es reemplazar el archivo con el respaldo más reciente.
- Btrieve no pudo asignar un manejador al archivo pre-imagen porque Btrieve no fue puesto en marcha con el acceso de derechos correctos al archivo de pre-imagen.

15 ERROR DE E/S EN EL ARCHIVO PRE-IMAGEN.

Un error de entrada / salida durante una función de pre-imagen. Se debe a que el disco está lleno o porque el archivo pre-imagen se dañó.

- Si el disco se llenó, borre archivos innecesarios para ganar espacio adicional.
- Si el archivo de pre-imagen está dañado, la integridad del archivo no puede ser asegurada. Utilice el comando RECOVER de la utilidad BUTIL para recobrar la información dañada de los archivos, o bien, reemplace el archivo con el respaldo más reciente.

17 ERROR AL CERRAR UN ARCHIVO.

Un error ha ocurrido mientras se escribía en disco la estructura del directorio previa al cierre del archivo. Puede ser que Btrieve no cierre el archivo, o que una nueva página sea adicionada al archivo y Btrieve no puede cerrar el archivo puesto que lo reabrirá para actualizar la estructura del directorio. Verifique la existencia de una falla del disco. Este código puede ocurrir si la aplicación pasa por una posición de block inválida.

18 DISCO LLENO.

El disco se encuentra lleno y el archivo no podrá ser expandido para acomodar una inserción. Borre cualquier archivo innecesario.

19 ERROR IRRECUPERABLE.

Un error irrecuperable ha ocurrido. La integridad del archivo no puede ser asegurada. Utilice el comando RECOVER de la utilidad BUTIL para recobrar la información dañada de los archivos, o bien, reemplace el archivo con el respado más reciente.

20 MANEJADOR DE REGISTROS BTRIEVE, INACTIVO.

Se ha solicitado una operación antes de que se haya puesto en marcha Btrieve. Ponga en marcha Btrieve (ver capítulo II).

21 EL BUFFER DE LLAVE ES MUY PEQUEÑO.

El parámetro de *Buffer de llave* no es lo suficientemente largo para acomodar el campo de la llave para el índice requerido. Verifique que el largo del *Buffer de llave* sea igual al largo definido para la llave especificada en el parámetro de *número de llave*. Este código puede ser regresado sólo por ciertas interfaces .

22 EL BUFFER DE DATOS ES MUY PEQUEÑO.

El parámetro de *Buffer de datos* no es lo suficientemente largo para acomodar la totalidad del registro de datos definido cuando el archivo fue creado. Verifique que el largo del *Buffer de datos* es al menos del mismo tamaño que el definido para el registro.

- Para las operaciones de *consulta* o *escalón*, si el *buffer de datos* es demasiado pequeño para contener la porción fija del registro, Btrieve no regresará ninguna información en el *buffer de datos* , si el registro es un registro de longitud variable y el *buffer de datos* es muy pequeño para contener la totalidad de la porción variable del registro, Btrieve regresará la información que le fue posible y regresará el código 22, indicando que no pudo regresar la totalidad del registro.
- Para la operación de *insertar*, Btrieve no insertará el registro si el *buffer de datos* es más pequeño que la porción fija del registro.
- Para la operación de *actualizar*, si el *buffer de datos* es muy pequeño para contener la porción fija del registro, Btrieve no actualizará el registro.
- Para las operaciones de *crear*, *status*, y *crea índice suplementario* un código 22 indica que el *buffer de datos* no es lo suficientemente largo para contener todas las especificaciones de archivo , de llave y las de secuencia alternativa de ordenamiento (si alguna fue definida).

23 EL LARGO DEL BLOCK DE POSICIÓN ES MUY PEQUEÑO.

El parámetro de *block de posición* no tiene 128 bytes de largo. Este error sólo puede ser detectado utilizando ciertas interfaces.

24 EL TAMAÑO DE PÁGINA ES MUY PEQUEÑO.

El *tamaño de página* es inválido, este debe ser múltiplo de 512 bytes pero no puede ser mayor a 4096 bytes.

25 ERROR DE E/S AL CREAR UN ARCHIVO.

El archivo especificado no pudo ser creado. Las posibles causas son: el directorio del disco está lleno o el disco mismo está lleno. Si el archivo que se especificó ya existe, Btrieve regresa este código de error si el archivo existente está catalogado como un archivo de transacción.

26 NÚMERO DE LLAVES INVÁLIDO.

El número de llaves especificado es inválido para el tamaño de página indicado. Para un archivo de Btrieve estandar con un tamaño de página de 512 bytes, el número de segmentos de llave debe ser de 1 a 8. Para tamaños de página mayores, el número de segmentos de llave puede ser de 1 a 24. Se debe definir al menos una llave sin el atributo de llave nula.

27 POSICIÓN DE LLAVE INVÁLIDA.

La posición del campo de llave especificada excede el largo del registro definido para el archivo porque la posición de llave es mayor al largo del registro o la posición de la llave más el largo de éste, excede el largo del registro. Para archivos de *sólo llave*, la llave debe empezar en el primer byte del registro (posición 1).

28 TAMAÑO DE REGISTRO INVÁLIDO.

El tamaño del registro es inválido. El largo del registro especificado debe ser menor o igual al tamaño de página menos 6; o mayor o igual a 4 bytes de largo.

29 TAMAÑO DE LLAVE INVÁLIDO.

El largo de la llave es inválido. El largo de la llave especificado debe ser mayor a 0 y no debe exceder los 255 bytes. Btrieve requiere que cada página de llave en el archivo sea lo suficientemente grande para contener al menos ocho llaves. Si el tamaño de página no es lo suficientemente grande para acomodar ocho llaves, cualquiera que éstas sean, del largo de llave especificado (más la cabecera); incremente el tamaño de página del archivo o decremente el largo de llave.

30 NO ES UN ARCHIVO BTRIEVE.

El archivo especificado no es un archivo de Btrieve válido. El archivo no fue creado por Btrieve, o este fue creado por una versión anterior no compatible a esta versión de Btrieve.

35 ERROR DE DIRECTORIO.

Ha ocurrido un error cuando se realizaba un cambio de directorio. El Drive especificado en la operación de *verificar directorio* no existe, o el directorio indicado en la operación de *fijar directorio* es inválido.

36 ERROR DE TRANSACCIÓN.

La operación de *inicio de transacción* no se pudo realizar por que no se definieron transacciones al poner en marcha o configurar Btrieve.

37 UNA TRANSACCIÓN ESTÁ ACTIVA.

Una operación de *inicio de transacción* fue utilizada mientras otra transacción está activa en la estación. La transacción no podrá ser iniciada.

38 ERROR DE E/S EN EL ARCHIVO DE CONTROL DE UNA TRANSACCIÓN.

Un error ha ocurrido cuando Btrieve intentó escribir sobre el archivo de control de transacción. Las posibles causas son que el disco se llenó, el disco tiene la protección contra escritura, o el archivo de control de transacción fue borrado.

39 ERROR EN LAS OPERACIONES *FIN* Ó *ABORTAR* TRANSACCIÓN.

Una operación de fin o abortar fue utilizada sin la operación correspondiente de *inicio de transacción*.

40 SE REBASÓ EL NÚMERO MÁXIMO DE ARCHIVOS EN UNA TRANSACCIÓN.

Una aplicación intentó actualizar más del máximo número de archivos permitidos dentro de una transacción. El número máximo de archivos diferentes que se pueden acceder durante una transacción lógica se indica cuando se pone en marcha o se configura Btrieve.

41 OPERACIÓN NO PERMITIDA.

Una aplicación está tratando de realizar una operación que no es permitida en ese momento. Algunas operaciones no son permitidas bajo algunas condiciones de operación. Por ejemplo, Btrieve regresa este código si se intenta utilizar alguna operación de *escalón*, *actualizar*, o *borrar* sobre un archivo de *sólo llave* o una operación de *consulta* sobre un archivo de *sólo datos*.

42 APERTURA ILEGAL A UN ARCHIVO QUE SE ENCUENTRA EN MODO ACELERADO.

Una aplicación trata de abrir un archivo que fue previamente accedido en modo acelerado y nunca se cerró satisfactoriamente. La integridad de la información no puede ser asegurada. Utilice el comando RECOVER de la utilería BUTIL para recobrar la información dañada de los archivos, o bien, reemplace el archivo con el respaldo más reciente.

43 DIRECCIÓN INVÁLIDA DEL REGISTRO.

La dirección del registro especificada por una operación de *consulta directamente* es inválida. La dirección está fuera de los límites del archivo; éste no es un registro que esté dentro de los límites de la página de datos, o sobre la página de datos. Los cuatro bytes de dirección que se especifican en las operaciones de *consulta directamente* deben de ser obtenidos por la operación de *obtener posición*

44 CONSULTA DIRECTA ILEGAL, POR TENER UN VALOR NULO EN LA LLAVE.

Una aplicación trata de utilizar la operación de *consulta directamente* para establecer un camino de índice para una llave que tiene un valor nulo en el registro correspondiente. Btrieve no puede establecer una posición basándose en una llave con valor nulo.

45 BANDERA DE LLAVE INCONSISTENTE.

La especificación de las banderas de llave en la operación de *crear* son inconsistentes. Si una llave es segmentada entonces los atributos de duplicable, modificable y nula deben ser los mismos para cada segmento que conforma la llave.

46 ACCESO AL ARCHIVO DENEGADO.

Una aplicación realizó una apertura a un archivo en modo de *sólo lectura* y trata de realizar una actualización, borrado, o inserción sobre ese archivo. Otra posible causa es que la clave de acceso requerida para actualizar, no fue especificada correctamente cuando el archivo fue abierto.

47 MÁXIMO DE ARCHIVOS ABIERTOS.

El máximo de archivos abiertos en modo acelerado excedió el número de buffers disponibles en la memoria intermedia de Btrieve. Cuando un archivo es abierto en modo acelerado, Btrieve aparta un buffer de la memoria intermedia para el archivo. Btrieve; siempre reserva cinco buffers vacíos para la manipulación del índice. Reconfigure Btrieve con un menor tamaño de página para disponer de más buffers.

48 DEFINICIÓN DE SECUENCIA ALTERNATIVA INVÁLIDA.

El primer byte definido en la secuencia alternativa (el byte de identificación) no contiene el valor hexadecimal de AC.

49 ERROR DE TIPO DE LLAVE.

La aplicación intenta crear un archivo o un índice suplementario con un tipo de llave extendida inválido, o trata de asignar a una secuencia alternativa una llave binaria o segmentada. Sólo se puede asignar una secuencia alternativa de tipo *Cadena*, *L-Cadena*, o *Z-Cadena*.

Este código también se da si se define un índice suplementario que requiere de una secuencia alternativa de ordenamiento, y ésta no existe en el archivo o en la definición de llave del buffer de datos.

50 SE INTENTA PONER UNA CLAVE A UN ARCHIVO QUE YA LA TIENE.

La aplicación trata de ejecutar una operación de *Fijar clave* a un archivo que ya tiene asignada una clave. Utilice la operación de *Eliminar clave* para remover la clave anterior y después especifique la nueva clave.

51 CLAVE INVÁLIDA.

Existen dos posibles causas para este error:

- Si se recibe el código después de ejecutar la operación de *fijar clave*, la clave especificada en la llave de buffer y en el buffer de datos no son iguales.
- Si se recibe el código después de ejecutar la operación de *Abrir*, el archivo tiene una clave asignada. La aplicación debe especificar la llave correcta en el buffer de datos.

52 ERROR AL ESCRIBIR DE LA MEMORIA INTERMEDIA.

Cuando se intenta crear el espacio para la memoria intermedia, vaciando su contenido, Btrieve trata de escribir datos en el drive lógico del disco de un archivo que anteriormente fue abierto en modo acelerado. Un error de E/S ocurrió durante la escritura.

53 INTERFAZ INVÁLIDA.

Una aplicación trata de acceder a un archivo que contiene registros de longitud variable con una interfaz de lenguaje de Btrieve v 3.15 o anterior. El acceso a un archivo con registros de longitud variable se debe realizar con una versión de interfaz de 4.xx o mayor.

54 ERROR DE PÁGINA VARIABLE.

Durante una operación de *Consulta directamente* Btrieve no puede leer toda o parte de la porción variable de un registro. En este caso, Btrieve regresa tantos datos como le sean posibles a la aplicación. Este error usualmente indica que el archivo contiene una o más páginas dañadas.

55 ERROR DE AUTO INCREMENTO.

La aplicación intenta especificar un atributo de segmentada o duplicada a una llave de tipo autoincrementable. Una llave autoincrementable no puede ser partida en otras llaves y tampoco puede ser duplicable.

56 ÍNDICE INCOMPLETO.

Un índice suplementario fue dañado. Esto puede ocurrir si la operación *Crear índice suplementario* o la operación *Borrar índice suplementario* fueron interrumpidas y no terminaron de realizar su función. Ejecute la operación *Borrar índice suplementario* para completar la eliminación del índice en el archivo.

57 ERROR EN LA MEMORIA EXPANDIDA

Cuando el manejador de memoria expandida manda un error a Btrieve, se regresa este código a la aplicación. Puede ser la incompatibilidad con otro programa que se encuentre en la memoria expandida.

58 BUFFER DE COMPRESIÓN MUY PEQUEÑO.

La aplicación trata de leer o escribir un registro que es más largo que el valor especificado en el tamaño de compresión del buffer. Reconfigure Btrieve especificando un valor más alto para el *Máximo tamaño de compresión de registro*.

59 ARCHIVO YA EXISTENTE.

Este código aparece cuando al utilizar la operación *Crear* se especifica un -1 en el parámetro de número de llave, y el nombre del archivo del parámetro de buffer de llave ya existe.

75 LISTA DEL RUTEADOR DEL SERVIDOR MUY PEQUEÑA.

La inicialización del parámetro /S en BRQPARMS necesita ser incrementado.

76 LISTA DEL SERVIDOR MUY PEQUEÑA.

Se han conectado más servidores de los que se puede tener. Desconecte los servidores que no necesita.

77 ERROR DE ESPERA VAP.

Un VAP que está en un servidor NetWare Btrieve está intentando leer un registro que fue bloqueado por otro cliente. Este código es regresado al cliente que bloqueó el registro.

78 DETECCIÓN DE BLOQUEOS MUERTOS.

Btrieve detectó una condición de Bloqueo muerto. La aplicación debe inicializar todos los recursos antes de proceder. Esto permite a otras aplicaciones acceder los recursos que estaban esperando.

79 ERROR DE PROGRAMACIÓN.

Existe un error interno de programación. Llame a la División de Desarrollo de Productos Novell.

80 CONFLICTO.

La operación de *Actualizar* o *Borrar* no pueden ser ejecutadas porque el registro fue cambiado por otra aplicación desde la aplicación de lectura del registro. Su aplicación debe releer el registro anterior a la ejecución de la operación de *Actualizar* o *Borrar*.

81 ERROR DE BLOQUEO.

Este error resulta por cualquiera de los siguientes tres motivos:

- Las tablas de Bloqueo de Btrieve están llenas. Decremente el número de bloqueos que utilice en su aplicación o bien reconfigure Btrieve y especifique un valor mayor para el *Máximo número de registros bloqueados*.
- Falló el llamado a la función de Bloqueo en red.

- La aplicación trata de desbloquear un registro que fue bloqueado con un múltiple bloqueo de registro, y la posición de registro colocado en el buffer de datos, no corresponde a ningún registro que fue bloqueado en ese archivo.

82 POSICIÓN PERDIDA.

Cuando se ejecuta una operación de *Consulta el siguiente* o *consulta el anterior* en una llave duplicada, la aplicación trató de consultar un registro que fue borrado o cuyo valor de llave fué modificado por otra aplicación. Reestablezca la posición utilizando la operación de *Consulta igual a*, o la operación *Consulta directamente*.

83 LECTURA FUERA DE LA TRANSACCIÓN.

La aplicación intenta borrar o actualizar un registro dentro de una transacción, pero el registro no puede ser leído dentro de la transacción. Si se va a actualizar o a borrar un registro dentro de una transacción, se debe leer el registro dentro de la transacción para asegurar que se tiene el acceso exclusivo a la información.

84 REGISTRO BLOQUEADO.

La aplicación trata de bloquear un registro que ya ha sido bloqueado en otra aplicación, o la aplicación trata de acceder a un archivo que está siendo utilizado dentro de una transacción.

85 ARCHIVO BLOQUEADO.

La aplicación trata de abrir un archivo o acceder un registro en una transacción o bloquear un registro mientras que otra aplicación está accediendo el archivo dentro de una transacción.

Si esta condición ocurre, se debe aplicar cualquiera de los siguientes métodos:

- Volver a intentar la operación hasta que se realice. Bajo una red pequeña, ésta es la más simple y rápida solución.
- Utilice bloqueos con espera en lugar de bloqueos sin espera. La aplicación que encuentre el bloqueo experimentará una espera momentánea hasta que el registro sea liberado.

86 TABLA DE ARCHIVOS LLENA.

La tabla de archivos de Btrieve está llena. Reconfigure Btrieve y especifique un valor mayor para el *Número máximo de archivos Abiertos*.

87 TABLA DE MANEJADORES LLENA.

La tabla de manejadores de Btrieve está llena. Reconfigure Btrieve y especifique un valor mayor para el *Número máximo de manejadores de archivos*.

88 MODO INCOMPATIBLE AL ABRIR UN ARCHIVO.

La aplicación trata de abrir un archivo en un modo incompatible. Si la primera aplicación accedió al archivo de modo acelerado, todas las demás aplicaciones deben acceder de modo acelerado. Si la primera aplicación accedió al archivo de modo no acelerado, todas las demás aplicaciones no lo harán.

90 TABLA DE REDIRECCIÓN DE DISPOSITIVO LLENA.

La tabla de redirección de dispositivo que utiliza Btrieve o el ruteador del servidor está llena. Esto ocurre si se conecta un servidor adicional o se mapea un drive adicional después de llamar el BREQUEST. Llame nuevamente al BREQUEST especificando un número mayor para las opciones de *Número de servidores de Archivo* o bien para el *Número de drives mapeables (/S y /R respectivamente)*.

Este error también ocurre si se desconecta de algún servidor en particular y se conecta a un servidor diferente. Mientras una estación de trabajo esté conectada a un servidor, BREQUEST no removerá su nombre de la tabla de ruta.

91 ERROR DE SERVIDOR.

BREQUEST no puede establecer una sesión de trabajo con el servidor. Esto se debe a que Btrieve no ha sido inicializado o el servidor no está activo. Verifique que Btrieve esté activo en el servidor.

92 TABLA DE TRANSACCIÓN LLENA.

El máximo número de transacciones que pueden estar activas se ha excedido. Reconfigure Btrieve y especifique un valor mayor para el *Número de Transacciones*.

93 TIPO DE BLOQUEO INCOMPATIBLE.

La aplicación trata de mezclar registros bloqueados (+100/+200) y múltiples registros bloqueados (+300/+400) en el mismo archivo al mismo tiempo. Todos los bloqueos deben ser de un mismo tipo y liberados antes de que algún bloqueo de otro tipo pueda ser ejecutado.

94 ERROR DE PRIVILEGIO O PERMISO.

La aplicación trata de abrir o crear un archivo en un directorio al cual no tiene acceso. Btrieve no puede pasar por encima de los privilegios o permisos de acceso que se les asignan a los usuarios.

95 SESIÓN INVALIDADA POR UN ERROR.

La sesión anterior no fue válida porque un error ocurrió en la estación de trabajo, en el servidor o en la red. Verifique que la estación de trabajo esté bien conectada al servidor y vuelva a poner en marcha el REQUESTER.

Este código también puede aparecer si el número máximo de sesiones de Btrieve ha sido excedido. Utilice el comando *CONSOLE B STATUS* para verificar si el número máximo de servidores no ha sido excedido. Si se excedió, reconfigure Btrieve incrementando el número de sesiones disponibles.

96 AMBIENTE DE COMUNICACIONES INVÁLIDO.

La tabla de conexión SPX está llena. Recargue SPX, especificando un valor mayor para la tabla de conexión. Consulte la documentación del sistema NetWare para mayor información.

97 EL LARGO DEL BUFFER DE DATOS ES MÁS GRANDE O CORTO DE LO DEFINIDO.

La aplicación trata de leer o escribir un registro el cual es más largo que el registro de Btrieve o que el REQUESTER pueden soportar. Reconfigure Btrieve especificando un valor mayor para la *Máxima longitud de registro*. Ponga en marcha nuevamente el REQUESTER y especifique un valor mayor para la opción /D.

- Para una operación de *Actualizar, Insertar o Crear*, la aplicación recibe este error si el largo del buffer de datos especificado por el registro, excede el largo definido por Btrieve.
- Para las operaciones de *Consulta, Escalón o Estatus*, la aplicación recibe este error si el valor especificado por el buffer de datos es menor que el largo de datos que Btrieve regresa.

98 ERROR INTERNO EN UNA TRANSACCIÓN.

Un error ocurrió durante una operación dentro de una transacción o en un archivo TTS. Todas las operaciones de *Insertar, Actualizar y Borrar*, desde el último *Inicio de Transacción* fueron reiniciadas. Se puede utilizar una operación de *Abortar Transacción* para cortar la operación y salir de la transacción.

TABLA DE ERRORES EN BTRIEVE

Código	DESCRIPCIÓN DEL ERROR
01	OPERACIÓN INVÁLIDA
02	ERROR DE E/S
03	ARCHIVO NO ABIERTO
04	VALOR DE LLAVE NO ENCONTRADO
05	VALOR DE LLAVE DUPLICADO
06	NÚMERO DE LLAVE INVÁLIDO
07	NÚMERO DE LLAVE DIFERENTE
08	POSICIONAMIENTO INVÁLIDO
09	FIN DE ARCHIVO
10	ERROR AL MODIFICAR EL VALOR DE LLAVE
11	NOMBRE DE ARCHIVO INVÁLIDO
12	ARCHIVO NO SE ENCUENTRA
14	ERROR DE APERTURA DEL ARCHIVO DE PRE-IMAGEN
15	ERROR DE E/S EN EL ARCHIVO DE PRE-IMAGEN
17	ERROR AL CERRAR UN ARCHIVO
18	DISCO LLENO
19	ERROR IRRECUPERABLE
20	MANEJADOR DE REGISTROS BTRIEVE, INACTIVO
21	EL <i>BUFFER DE LLAVE</i> ES MUY PEQUEÑO
22	EL <i>BUFFER DE DATOS</i> ES MUY PEQUEÑO
23	EL LARGO DEL <i>BLOCK DE POSICIÓN</i> ES MUY PEQUEÑO
24	EL TAMAÑO DE PÁGINA ES MUY PEQUEÑO
25	ERROR DE E/S AL CERRAR UN ARCHIVO
26	NÚMERO DE LLAVES INVÁLIDO
27	POSICIÓN DE LLAVE INVÁLIDA
28	TAMAÑO DE REGISTRO INVÁLIDO
29	TAMAÑO DE LLAVE INVÁLIDO
30	NO ES UN ARCHIVO BTRIEVE
35	ERROR DE DIRECTORIO
36	ERROR EN TRANSACCIÓN
37	UNA TRANSACCIÓN ESTÁ ACTIVA
38	ERROR DE E/S EN EL ARCHIVO DE CONTROL DE UNA TRANSACCIÓN
39	ERROR EN LAS OPERACIONES <i>FIN</i> Ó <i>ABORTAR</i> TRANSACCIÓN
40	SE REBASÓ EL NÚMERO MÁXIMO DE ARCHIVOS EN UNA TRANSACCIÓN
41	OPERACIÓN NO PERMITIDA
42	APERTURA ILEGAL DE UN ARCHIVO QUE SE ENCUENTRA EN MODO ACELERADO
43	DIRECCIÓN INVÁLIDA DEL REGISTRO
44	CONSULTA DIRECTA ILEGAL, POR TENER UN VALOR NULO EN LA LLAVE
45	BANDERA DE LLAVE INCONSISTENTE
46	ACCESO AL ARCHIVO DENEGADO
47	MÁXIMO DE ARCHIVOS ABIERTOS
48	DEFINICIÓN DE SECUENCIA ALTERNATIVA INVÁLIDA
49	ERROR DE TIPO DE LLAVE

Continuación...

Código	DESCRIPCIÓN DEL ERROR
50	SE INTENTA PONER UNA CLAVE A UN ARCHIVO QUE YA LA TIENE
51	CLAVE INVÁLIDA
52	ERROR AL ESCRIBIR DE LA MEMORIA INTERMEDIA
53	INTERFAZ INVÁLIDA
54	ERROR DE PÁGINA VARIABLE
55	ERROR DE AUTO INCREMENTO
56	ÍNDICE INCOMPLETO
57	ERROR DE EXPANSIÓN DE MEMORIA
58	BUFFER DE COMPRESIÓN MUY PEQUEÑO
59	ARCHIVO YA EXISTENTE
75	LISTA DEL RUTEADOR DEL SERVIDOR MUY PEQUEÑA
76	LISTA DEL SERVIDOR MUY PEQUEÑA
77	ERROR DE ESPERA VAP
78	DETECCIÓN DE BLOQUEOS MUERTOS
79	ERROR DE PROGRAMACIÓN
80	CONFLICTO
81	ERROR DE BLOQUEO
82	POSICIÓN PERDIDA
83	LEER FUERA DEL LA TRANSACCIÓN
84	REGISTRO BLOQUEADO
85	ARCHIVO BLOQUEADO
86	TABLA DE ARCHIVOS LLENA
87	TABLA DE MENEJADORES LLENA
88	MODO INCOMPATIBLE AL ABRIR UN ARCHIVO
90	TABLA DE DIRECCIÓN DE DISPOSITIVO LLENA
91	ERROR DE SERVIDOR
92	TABLA DE TRANSACCIÓN LLENA
93	TIPO DE BLOQUEO INCOMPATIBLE
94	ERROR DE PRIVILEGIO O PERMISO
95	SESIÓN INVALIDADA POR UN ERROR EN LA ESTACIÓN DE TRABAJO
96	AMBIENTE DE COMUNICACIONES INVÁLIDO
97	EL LARGO DEL BUFFER DE DATOS MÁS GRANDE O CORTO DE LO DEFINIDO
98	ERROR INTERNO EN UNA TRANSACCIÓN
	Los códigos que no aparecen, son considerados como no definidos.

Tabla A.01.

APÉNDICE B

TIPOS DE LLAVES ADICIONALES

Este apéndice describe los códigos de los tipos de llaves adicionales y la forma de almacenamiento interno para los tipos de llave que soporta Btrieve.

Un resumen de los tipos de llaves adicionales y sus códigos se muestran en la tabla B.01.

TIPO	CÓDIGO	Especificación en el archivo de descripción que maneja la utilidad BUTIL
Cadena	0	str
Entero	1	int
Real	2	flo
Fecha	3	dat
Hora	4	tim
Decimal	5	dec
Monetaria	6	mon
Lógica	7	log
Númerica	8	num
B-Real	9	bfl
L-Cadena	10	lst
Z-Cadena	11	zst
Binaria sin signo	14	uns
Autoincrementable	15	aut

Tabla B.01.
Códigos de Tipos de Llaves Adicionales.

AUTOINCREMENTABLE

Una llave autoincrementable es una llave de tipo entero que puede tener dos o cuatro bytes de longitud. Btrieve ordena las llaves autoincrementables por su valor absoluto, comparando los valores almacenados en diferentes registros leyéndolos de derecha a izquierda.

Las llaves autoincrementables son incrementadas vez cada que se inserta un registro dentro del archivo.

Las siguientes restricciones se aplican a las variables autoincrementables:

- Una llave autoincrementable no puede tener valores duplicados.
- Una llave autoincrementable no puede ser segmentada o incluirse como un segmento de otra llave.
- Una llave autoincrementable no puede traslaparse con otra llave.

Los siguientes párrafos describen como Btrieve trata los valores de la llave autoincrementable, cuando se inserta un registro en el archivo.

Si se especifica un valor binario de 0 para la llave autoincrementable, Btrieve asignará un valor a la llave basándose en el siguiente criterio:

- Si el registro que se inserta es el primero que va a contener el archivo, Btrieve le asigna el valor de 1 a la llave autoincrementable e inserta el registro en el archivo.
- Si ya existen registros en el archivo, Btrieve asigna el siguiente valor tomando en cuenta el último número asignado e inserta el nuevo registro dentro del archivo.

Si se especifica un valor diferente de cero para la llave autoincrementable, Btrieve inserta el registro dentro del archivo y utiliza el valor especificado como el valor de la llave. Si el registro contiene un valor que ya exista en el archivo, Btrieve regresará un código de error diferente de cero y no insertará el registro en el archivo.

Cuando se borre un registro que tiene una llave autoincrementable, Btrieve remueve completamente el registro del archivo. Btrieve no reutiliza valores de llaves borradas a menos de que se le especifique ese valor cuando se inserta otro registro en el archivo.

Como se mencionó anteriormente, Btrieve siempre ordena las llaves autoincrementables por su valor absoluto. Si se especifica un valor negativo para alguna llave autoincrementable, cuando se inserta un registro o se actualiza, Btrieve ordena el valor de acuerdo a su valor absoluto. Esto permite que se use la bandera de negación en registros sin alternar la posición del registro en el índice. En conclusión, cuando se ejecute una operación de *Consulta* y se especifique un valor negativo en el buffer de llave, Btrieve tratará ese valor negativo como su valor absoluto.

B-REAL

Un campo con un tipo B-Real es un número real de simple o doble precisión. Un número real de simple precisión es almacenado con un 23-bit mantisa, 8-bit de exponente basado por 128, y un bit de signo. La representación interna para un real de 4 bytes es la siguiente:



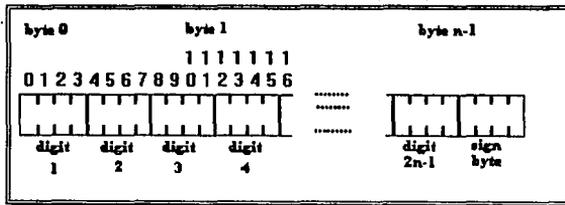
La representación de un número real de doble precisión es el mismo que para el de simple precisión excepto porque la mantisa es de 55 bits en vez de 23 bits. El mínimo significativo 32 bits es almacenado en bytes 0 a 3.

FECHA

Los campos tipo fecha son almacenados internamente como valores 4-byte. El día y el mes son almacenados en 1-byte de forma binaria. El año es un número binario de 2-bytes que representan el valor del año. El día se encuentra en el primer byte, el mes en el segundo byte y el año en los dos últimos bytes después del mes.

DECIMAL

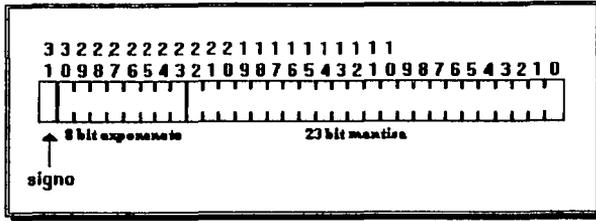
Los campos tipo decimal son almacenados internamente como paquetes de números decimales de dos dígitos por byte. La representación interna para un campo n-byte es la siguiente:



El signo byte es F o C para números positivos y D para números negativos. Atención porque el punto decimal está implícito. Todos los valores para llaves decimales deben tener el mismo número de decimales en orden para que Btrieve ordene la llave correctamente.

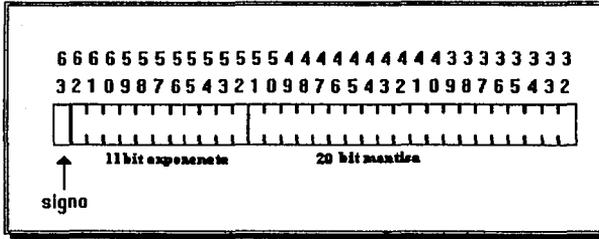
REAL

Un tipo real es consistente con el IEEE estandar para números reales de simple y doble precisión. EL formato interno para un 4-byte real consta de 24 bits de mantisa, 8 bits para la parte exponencial y un bit de signo. La ilustración es la siguiente:



Un campo de tipo real con 8 bytes tiene 52 bits de mantisa, 11 bits para la parte exponencial y un bit de signo. El formato interno es el siguiente:

bytes 7-4



bytes 3-0



ENTERO

Un tipo entero es un número con signo que tiene un número constante de bytes. Internamente, los enteros son almacenados en el formato binario de Intel, con el byte más significativo y el byte menos significativo invertidos dentro de una palabra. Btrieve evalúa la llave de derecha a izquierda, palabra por palabra. El signo debe de estar colocado en la primera parte del byte menos significativo.

LÓGICO

El tipo de llave lógico extendido es colocado en uno o dos bytes. Btrieve ordena este tipo de llave como una cadena. Esto permite que la aplicación determine que valor representa lo verdadero y falso.

L-CADENA

Un L-Cadena en Btrieve tiene las mismas características que un tipo cadena excepto que el primer byte de la cadena contiene la representación binaria del largo de cadena. El largo colocado en el byte 0 de la cadena tipo L-Cadena determina el número de bytes significativos. Btrieve ignora cualquier valor que vaya más allá del largo especificado en la cadena. Este tipo puede ser útil por ejemplo en compiladores como Turbo Pascal.

DINERO

La representación interna para el tipo dinero, es exactamente la misma que para el tipo decimal.

NUMÉRICA

Los valores numéricos son almacenados como una cadena ASCII, justificada a la derecha con ceros. Cada dígito ocupa un byte internamente. El byte más a la derecha del número incluye el signo. La tabla B.02, indica como se representa el dígito más a la derecha, cuando contiene el signo positivo y el signo negativo

DIGITO	POSITIVO	NEGATIVO
1	A	J
2	B	K
3	C	L
4	D	M
5	E	N
6	F	O
7	G	P
8	H	Q
9	I	R
0	{	}

Tabla B.02

Dígito que representa el signo en la llave tipo numérica.

Para números positivos, el dígito derecho puede ser representado en el caso de "1" a "0" por "A" hasta "{". Btrieve procesa números positivos representados de cualquier forma.

CADENA

Un tipo cadena en Btrieve es una secuencia de caracteres ordenados de izquierda a derecha. Cada caracter se respresenta en formato ASCII en un sólo byte.

TIEMPO

Los tipos de campo tiempo son almacenados internamente en valores de 4-bytes. Centésimas de segundo, segundos, minutos, y horas son almacenados en 1-byte de forma binaria. Las centésimas de segundo se encuentran en el primer byte, seguidas respectivamente por los segundos, minutos y horas.

BINARIO SIN SIGNO

Btrieve ordena los tipos de llave binario sin signo, como un entero sin signo. Un tipo binario sin signo mantiene constante el número de bytes. Btrieve compara los tipos de llave binario sin signo palabra por palabra de derecha a izquierda.

Z-CADENA

Un tipo Z-Cadena en Btrieve corresponde a las cadenas utilizadas en los compiladores de lenguaje "C". Tiene las mismas características que la cadena normal, excepto proque la Z-Cadena finaliza con un byte que contiene un 0 binario. Btrieve ignora cualquier valor que rebase el primer 0 binario que encuentre.

APÉNDICE C

¿COMO FUNCIONAN LAS INTERFACES ?

De forma estricta Btrieve es un programa residente de tipo *TSR (Terminate-and-Stay-Resident)* pasivo, es decir, para hacer uso de este manejador primero se debe cargar en la memoria el programa y después ejecutar la aplicación. Al ejecutar el programa BTRIEVE.EXE o BREQUEST.EXE éste permanece residente en memoria, modifica la tabla de vectores de interrupciones, para qué de esta forma, quede como un ISR (Interrupt-Service-Routine), rutina de servicios de interrupciones. Esta tabla de vectores la conforman las rutinas del Rom-Bios, interrupciones y funciones del DOS y algunos lugares quedan sin uso, precisamente uno de estos lugares, el 07BH, es el que ocupa Btrieve como número de interrupción. Este número de interrupción es necesario, si se pretende elaborar una Interfaz no incluida con el paquete.

De manera más clara, ahora, es posible definir cómo funcionan las interfaces anteriormente mencionadas. La interfaz de un lenguaje específico es una función o procedimiento que tiene como parámetros la información requerida para ejecutar una operación del manejador Btrieve. De forma general una interfaz sigue los siguientes pasos para comunicarse con Btrieve:

- Al ser ejecutada la función o procedimiento lo primero que debe realizar es verificar si Btrieve está instalado, para este efecto utiliza la interrupción 21H del DOS con la función 35H (AH=53H), esta función busca el vector de una interrupción específica (AL=07BH) dentro de la tabla de vectores de interrupciones, la función da como resultado los registros ES:BX en el registro ES regresa el segmento de memoria y en BX regresa el offset que es el importante, si BX es igual con 0033H quiere decir que Btrieve si está instalado, de lo contrario la función debe retornar como status el código de error No. 20, que es el código para indicar que el manejador no está instalado.
- Si Btrieve está instalado, entonces la función debe crear un registro de memoria llamado "block de parámetros", éste debe contener en determinada secuencia, los datos y direcciones de los parámetros de la función o procedimiento.
- Finalmente ejecuta la interrupción 07BH que le da el control a Btrieve, lo único que necesita esta interrupción es que en el registro DS esté el segmento de memoria en el que se encuentra el block de parámetros y en DX el offset.

Esto es lo que realizan las llamadas interfaces. Algunos lenguajes contienen instrucciones para ejecutar directamente las interrupciones como por ejemplo Pascal, C, Basic compilado, pero algunos otros no (Basic intérprete) y tienen que recurrir a funciones en lenguaje máquina para que éste realice las interrupciones y procesos necesarios.

No olvidar que el paquete de Btrieve contiene codificadas las interfaces para diferentes versiones de los lenguajes más populares y si registramos la compra del paquete debidamente tendremos acceso a actualizaciones del producto así como a interfaces de nuevas versiones de lenguajes.

Nota: para mayor información sobre el tema puede consultar el apéndice D donde se encuentra el código fuente de algunas interfaces.

APENDICE D

ALGUNAS INTERFACES DE LENGUAJES COMUNES.

En este apéndice se enlistan algunos códigos fuente de interfaces y en otros casos se presenta un programa para generar el archivo objeto que la contiene, esto es porque algunas de las interfaces se encuentran únicamente como archivos .OBJ

Si se cuenta con una red NOVELL, hacen falta dos cosas importantes para poder utilizar Btrieve en la modalidad *basado en servidor*: el manual del programador y las interfaces, esto quiere decir que parte del software está incluido en el sistema operativo de NOVELL pero no lo documentan y sin la parte que falta no se puede utilizar. El camino a seguir es adquirir lo necesario aparte.

Si no se tiene la intención de invertir más para hacer uso de Btrieve en esta modalidad, se puede utilizar esta tesina como manual y recuperar las interfaces que se presentan de manera escrita en este apéndice.

Por otra parte, para utilizar la modalidad *basado en cliente* sólo es necesario el archivo BTRIEVE.EXE, este se puede localizar, en redes de compuservicio o programas comerciales como *Norton PcAnywhere* (programa comercial de comunicaciones, que ocupa Btrieve), Xtrieve u otra aplicación que lo utilice.

Las interfaces que se presentan en este apéndice permiten utilizar Btrieve con los siguientes lenguajes:

- Borland Turbo Pascal
- Borland Turbo Pascal V5.0
- Borland Turbo C V2.0
- Borland Turbo Basic
- Microsoft Quick Basic V2.0 y otros compatibles con Gwbasic.
- Microsoft BASIC V 7.0
- Microsoft COBOL
- IBM COBOL
- Micro Focus COBOL /2
- Microsoft COBOL V3.0
- Relia COBOL
- Microsoft Fortran
- IBM Pascal
- Microsoft BASIC v6.0
- Microsoft QuickBasic v4.0, v4.5, v6.0
- IBM BASIC v1.0, v2.0, v3.0

A continuación se presenta el código fuente de algunas interfaces:

Nombre de la interfaz:	TUXBTRV.PAS
Lenguaje(s) que la utiliza(n):	Borland Turbo Pascal

```

{
  Module Name: TUXBTRV.PAS
}
{
  Description: This is the Btrieve interface for Turbo Pascal (MS-DOS).
  This routine sets up the parameter block expected by
  Btrieve, and issues interrupt 7B. It should be compiled
  with the $V- switch so that runtime checks will not be
  performed on the variable parameters.
}
{
  Synopsis: STAT := BTRV (OP, POS.START, DATA.START, DATALEN,
                      KBUF.START, KEY);
  where
  OP is an integer,
  POS is a 128 byte array,
  DATA is an untyped parameter for the data buffer,
  DATALEN is the integer length of the data buffer,
  KBUF is the untyped parameter for the key buffer,
  and KEY is an integer.
}
{
  Returns: Btrieve status code.
}
{
}
{
  var DATA = record case boolean of
  FALSE: ( START: integer );
  TRUE: ( EMPLOYEE_ID: 0..99999;
        EMPLOYEE_NAME: packed array[1..50] of char;
        SALARY: Real;
        DATA_OF_HIRE: DATE_TYPE );
  end;
}
{
  There should NEVER be any string variables declared in the
  data or Key records, because strings store an extra byte for
  the length, which affects the total size of the record.
}
function BTRV (OP:integer; var POS,DATA; var DATALEN: integer;
              var KBUF; KEY: integer): integer;
const
  PASCAL_ID      = $AAAA;                (Pascal language id)
  VAR_ID         = $6176;                (id for variable length records - 'va')
  BTR_INT        = $7B;
  BTR2_INT       = $2F;
  BTR_OFFSET     = $0033;
  MULTI_FUNCTION = $AB;
{
  ProcId is used for communicating with the Multi Tasking Version of
  Btrieve. It contains the process id returned from BMulti and should
  not be changed once it has been set.
}
{
  ProcId: integer = 0;                    ( initialize to no process id )
  MULTI: boolean = false;                ( set to true if BMulti is loaded )
  VSet: boolean = false;                 ( set to true if we have checked for BMulti )
}
type
  ADDR32 = record                          (32 bit address)
    OFFSET: integer;
    SEGMENT: integer;
  end;
  BTR_PARAMS = record
    USER_BUF_ADDR: ADDR32;                (data buffer address)
    USER_BUF_LEN: integer;                (data buffer length)
    USER_CUR_ADDR: ADDR32;                (currency block address)

```

```

USER_FCB_ADDR: ADDR32;           (file control block address)
USER_FUNCTION: integer;         (Btrieve operation)
USER_KEY_ADDR: ADDR32;          (key buffer address)
USER_KEY_LENGTH: BYTE;          (key buffer length)
USER_KEY_NUMBER: BYTE;          (key number)
USER_STAT_ADDR: ADDR32;         (return status address)
XFACE_ID: integer;              (language interface id)
end;

```

```

Result = record
  AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS: integer;
end;

```

```

var
  STAT: integer;                 (Btrieve status code)
  XDATA: BTR_PARAMS;            (Btrieve parameter block)
  REGS: Result;                 (register structure used on interrupt call)
  DONE: boolean;

```

```

begin
  REGS.AX := $3500 + BTR_INT;
  INTR ($21, REGS);
  if (REGS.BX <> BTR_OFFSET) then (make sure Btrieve is installed)
    STAT := 20
  else
    begin
      if (not VSet) then (if we haven't checked for Multi-User version)
        begin
          REGS.AX := $3000;
          INTR ($21, REGS);
          if ((REGS.AX AND $00FF) >= 3) then
            begin
              VSet := true;
              REGS.AX := MULTI_FUNCTION * 256;
              INTR (BTR2_INT, REGS);
              MULTI := ((REGS.AX AND $00FF) = $004D);
            end
          else
            MULTI := false;
          end;
        end;
      with XDATA do
        begin
          USER_BUF_ADDR.SEGMENT := SEG (DATA);
          USER_BUF_ADDR.OFFSET := OFS (DATA); (set data buffer address)
          USER_BUF_LEN := DATALEN;
          USER_FCB_ADDR.SEGMENT := SEG (POS);
          USER_FCB_ADDR.OFFSET := OFS (POS); (set FCB address)
          USER_CUR_ADDR.SEGMENT := USER_FCB_ADDR.SEGMENT; (set cur seg)
          USER_CUR_ADDR.OFFSET := USER_FCB_ADDR.OFFSET+38; (set cur ofs)
          USER_FUNCTION := OP; (set Btrieve operation code)
          USER_KEY_ADDR.SEGMENT := SEG (KBUF);
          USER_KEY_ADDR.OFFSET := OFS (KBUF); (set key buffer address)
          USER_KEY_LENGTH := 255; (assume its large enough)
          USER_KEY_NUMBER := KRY; (set key number)
          USER_STAT_ADDR.SEGMENT := SEG (STAT);
          USER_STAT_ADDR.OFFSET := OFS (STAT); (set status address)
          XFACE_ID := VAR_ID; (set language id)
        end;
      REGS.DX := OFS (XDATA);
      REGS.DS := SEG (XDATA);

      if (NOT MULTI) then (MultiUser version not installed)
        INTR (BTR_INT, REGS)
      else
        begin
          DONE := FALSE;
          repeat
            REGS.BX := ProcId;
            REGS.AX := 1;
            if (REGS.BX <> 0) then

```

```
REGS.AX := 2;  
REGS.AX := REGS.AX + (MULTI_FUNCTION * 256);  
INTR (BTR2 INT, REGS);  
if ((REGS.AX AND $00FF) = 0) then  
  DONE := TRUE  
else begin  
  REGS.AX := $0200;  
  INTR ($7F, REGS);  
  DONE := FALSE;  
end;  
until (DONE);  
if (ProcId = 0) then  
  ProcId := REGS.BX;  
end;  
DATALEN := XDATA.USER_BUF_LEN;  
end;  
BTRV := STAT;  
end;
```

Nombre de la interfaz:	TUR5BTRV.PAS
Lenguaje(s) que la utiliza(n):	Borland Turbo Pascal V5.0

```

UNIT TUR5BTRV;

INTERFACE
USES DOS;

FUNCTION BTRV (OP:INTEGER;
              VAR POS,
              DATA;
              VAR DATALEN: integer;
              VAR KBUF;
              KEY: INTEGER): INTEGER;

IMPLEMENTATION

($R-) (Range checking off)
($B+) (Boolean complete evaluation on)
($S+) (Stack checking on)
($I+) (I/O checking on)

{
{ Module Name: TUR5BTRV.PAS }
{
{ Description: This is the Btrieve interface for Turbo Pascal (MS-DOS). }
{ This routine sets up the parameter block expected by }
{ Btrieve, and issues interrupt 7B. It should be compiled }
{ with the $V- switch so that runtime checks will not be }
{ performed on the variable parameters. }
{
{ Synopsis: STAT := BTRV (OP, POS.START, DATA.START, DATALEN, }
{ KBUF.START, KEY); }
{ where }
{ OP is an integer, }
{ POS is a 128 byte array, }
{ DATA is an untyped parameter for the data buffer, }
{ DATALEN is the integer length of the data buffer, }
{ KBUF is the untyped parameter for the key buffer, }
{ and KEY is an integer. }
{
{ Returns: Btrieve status code. }
{
{ Note: The Btrieve manual states that the 2nd, 3rd, and 5th }
{ parameters be declared as variant records, with an integer }
{ type as one of the variants (used only for Btrieve calls), }
{ as is shown in the example below. This is supported, but }
{ the restriction is no longer necessary. In other words, any }
{ variable can be sent in those spots as long as the variable }
{ uses the correct amount of memory so Btrieve does not }
{ overwrite other variables. }
{
{ var DATA = record case boolean of }
{ FALSE: ( START: integer ); }
{ TRUE: ( EMPLOYEE_ID: 0..99999; }
{ EMPLOYEE_NAME: packed array[1..50] of char; }
{ SALARY: Real; }
{ DATA_OF_HIRE: DATE_TYPE ); }
{ end; }
{
{ There should NEVER be any string variables declared in the }
{ data or key records, because strings store an extra byte for }
{ the length, which affects the total size of the record. }
{
function BTRV (OP:integer; var POS,DATA; var DATALEN: integer;
              var KBUF; KEY: integer): integer;
const

```

```

VAR ID      = $6176;      (id for variable length records - 'va')
BTR_INT    = $7B;
BTR2_INT   = $2F;
BTR_OFFSET = $0033;
MULTI_FUNCTION = $AB;

{ ProcId is used for communicating with the Multi Tasking Version of }
{ Btrieve. It contains the process id returned from BMulti and should }
{ not be changed once it has been set. }
{
  ProcId: integer = 0;      ( initialize to no process id )
  MULTI: boolean = false;  ( set to true if BMulti is loaded )
  VSet: boolean = false;   ( set to true if we have checked for BMulti )
}

type
ADDR32 = record
  OFFSET: word;           (32 bit address)
  SEGMENT: word;         (offset->integer)
end;
BTR_PARMS = record
  USER_BUF_ADDR: ADDR32; (data buffer address)
  USER_BUF_LEN: integer; (data buffer length)
  USER_CUR_ADDR: ADDR32; (currency block address)
  USER_FCB_ADDR: ADDR32; (file control block address)
  USER_FUNCTION: integer; (Btrieve operation)
  USER_KEY_ADDR: ADDR32; (key buffer address)
  USER_KEY_LENGTH: BYTE; (key buffer length)
  USER_KEY_NUMBER: shortint; (key number&old->BYTE)
  USER_STAT_ADDR: ADDR32; (return status address)
  XFACE_ID: integer;     (language interface id)
end;

var
STAT: integer;           (Btrieve status code)
XDATA: BTR_PARMS;      (Btrieve parameter block)
REGS: Dos.Registers;   (register structure used on interrupt call)
DONE: boolean;

begin
  REGS.AX := $3500 + BTR_INT;
  INTR ($21, REGS);
  if (REGS.BX <> BTR_OFFSET) then (make sure Btrieve is installed)
    STAT := 20
  else
    begin
      if (not VSet) then (if we haven't checked for Multi-User version)
        begin
          REGS.AX := $3000;
          INTR ($21, REGS);
          if ((REGS.AX AND $00FF) >= 3) then
            begin
              VSet := true;
              REGS.AX := MULTI_FUNCTION * 256;
              INTR (BTR2_INT, REGS);
              MULTI := ((REGS.AX AND $00FF) = $004D);
            end
          else
            MULTI := false;
        end
      end;
    end;
  with XDATA do
    begin
      USER_BUF_ADDR.SEGMENT := SEG (DATA);
      USER_BUF_ADDR.OFFSET := OFS (DATA); (set data buffer address)
      USER_BUF_LEN := DATALEN;
      USER_FCB_ADDR.SEGMENT := SEG (POS);
      USER_FCB_ADDR.OFFSET := OFS (POS); (set FCB address)
      USER_CUR_ADDR.SEGMENT := USER_FCB_ADDR.SEGMENT; (set cur seg)
      USER_CUR_ADDR.OFFSET := USER_FCB_ADDR.OFFSET+30; (set cur ofs)
      USER_FUNCTION := OP; (set Btrieve operation code)
      USER_KEY_ADDR.SEGMENT := SEG (KBUF);
    end
  end;
end;

```

```

USER_KEY_ADDR.OFFSET := OFS (KBUF); {set key buffer address}
USER_KEY_LENGTH := 255; {assume its large enough}
USER_KEY_NUMBER := KEY; {set key number}
USER_STAT_ADDR.SEGMENT := SEG (STAT);
USER_STAT_ADDR.OFFSET := OFS (STAT); {set status address}
XFACE_ID := VAR_ID; {set language id}
end;

```

```

REGS.DX := OFS (XDATA);
REGS.DS := SEG (XDATA);

```

```

if (NOT MULTI) then {MultiUser version not installed}
  INTR (BTR_INT, REGS)
else

```

```

  begin
    DONE := FALSE;
    repeat
      REGS.EX := ProcId;
      REGS.AX := 1;
      if (REGS.EX < 0) then
        REGS.AX := 2;
      REGS.AX := REGS.AX + (MULTI_FUNCTION * 256);
      INTR (BTR2_INT, REGS);
      if ((REGS.AX AND $00FF) = 0) then
        DONE := TRUE
      else begin
        REGS.AX := $0200;
        INTR ($7F, REGS);
        DONE := FALSE;
      end;
    until (DONE);
    if (ProcId = 0) then
      ProcId := REGS.EX;
    end;
  end;

```

```

  DATALEN := XDATA.USER_BUF_LEN;
end;

```

```

  BTRV := STAT;

```

```

end;
{$B-}
END.

```

Nombre de la interfaz:

TURCBTRV.PAS

Lenguaje(s) que la utiliza(n):

Borland Turbo C V2.0

```

/*
    Borland's Turbo C Version 2 interface to the
    Btrieve Record Manager, version 4, 5, 5.1
*/

#include <dos.h>
#define BTR_ERR      20          /* record manager not started */
#define BTR_INT      0x7B       /* Btrieve interrupt vector */
#define BTR2_INT     0x2F       /* multi-user interrupt vector */
#define BTR_VECTOR   BTR_INT * 4 /* offset for interrupt */
#define BTR_OFFSET   0x33       /* Btrieve offset within segment */
#define VARIABLE_ID  0x6176     /* id for variable length records - 'va' */
#define _2FCODE      0xAB00     /* function code for int 2F to btrieve */

/* For Compact, Large, and Huge memory models
   /* LMODEL means 32-bit pointers in use
*/

#if defined( _COMPACT_ ) || defined( _LARGE_ ) || defined( _HUGE_ )
#define LMODEL 1
#endif

/* ProcId is used for communicating with the Multi Tasking Version of
   /* Btrieve. It contains the process id returned from BMulti and should
   /* not be changed once it has been set.
*/
/*
static unsigned ProcId = 0;          /* initialize to no process id */
static char MULTI = 0;              /* flag set to true if MultiUser */
static char VSet = 0;               /* flag set to true if checked version */

int BTRV( OP, POS_BLK, DATA_BUF, DATA_LEN, KEY_BUF, KEY_NUM)
int OP;
char POS_BLK[];
char DATA_BUF[];
int *DATA_LEN;
char KEY_BUF[];
int KEY_NUM;

{
union REGS b_regs;          /* define registers */

struct SREGS b_sregs;      /* define segment registers */

struct BTRIEVE_PARMS /* structure passed to Btrieve Record Manager */
{
char *BUF_OFFSET;          /* callers data buffer offset */
#ifndef LMODEL
int BUF_SEG;              /* callers data buffer segment */
#endif
int BUF_LEN;              /* length of data buffer */
char *CUR_OFFSET;         /* user position block offset */
#ifndef LMODEL
int CUR_SEG;              /* user position block segment */
#endif
char *FCB_OFFSET;         /* offset of disk FCB */
#ifndef LMODEL
int FCB_SEG;              /* segment of disk FCB */
#endif
int FUNCTION;             /* requested function */
char *KEY_OFFSET;        /* offset of user's key buffer */
#ifndef LMODEL
int KEY_SEG;              /* segment of user's key buffer */
#endif
char KEY_LENGTH;         /* length of user's key buffer */
char KEY_NUMBER;         /* key of reference for request */
}

```

```

int *STAT_OFFSET; /* offset of status word */
#ifdef LMODEL
int STAT_SEG; /* segment of status word */
#endif
int XFACE_ID; /* language identifier */
} XDATA;

int STAT = 0; /* status of Btrieve call */

/*
/* Check to see that the Btrieve Record Manager has been started.
/*
if (!VSet) /* if we don't know version of Btrieve yet */
{
VSet = 1;
b_regs.x.ax = 0x3000; /* check dos version */
int86x (0x21, &b_regs, &b_sregs);
if ((b_regs.x.ax & 0x00FF) >= 3) /* if DOS version 3 or later */
{
b_regs.x.ax = 2FCODE;
int86x (BTR2_INT, &b_regs, &b_sregs);
MULTI = ((b_regs.x.ax & 0xFF) == 'M'); /* if al is M, bmulti is loaded */
}
}

if (!MULTI) /* if bmulti not loaded */
{
b_regs.x.ax = 0x3500 + BTR_INT;
int86x (0x21, &b_regs, &b_sregs);
if (b_regs.x.bx != BTR_OFFSET)
return (BTR_ERR);
}

/* Read segment registers and initialize segment part of addresses to
/* user's data segment.
/*

segread (&b_sregs);
#ifdef LMODEL
XDATA.BUF_SEG = XDATA.CUR_SEG = XDATA.FCB_SEG =
XDATA.KEY_SEG = XDATA.STAT_SEG = b_sregs.ss;
#endif

/*
/* Move user parameters to XDATA, the block where Btrieve expects them.
/*

XDATA.FUNCTION = OP;
XDATA.STAT_OFFSET = &STAT;
XDATA.FCB_OFFSET = POS_BLK;
XDATA.CUR_OFFSET = POS_BLK + 38;
XDATA.BUF_OFFSET = DATA_BUF;
XDATA.BUF_LEN = *DATA_LEN;
XDATA.KEY_OFFSET = KEY_BUF;
XDATA.KEY_LENGTH = 255; /* use max since we don't know */
XDATA.KEY_NUMBER = KEY_NUM;
XDATA.XFACE_ID = VARIABLE_ID;

/*
/* Make call to the Btrieve Record Manager.
/*

b_regs.x.dx = (int) &XDATA; /* parameter block is expected to be in DX */
b_sregs.ds = b_sregs.ss;
if (!MULTI)
int86x (BTR_INT, &b_regs, &b_sregs);
else
{
while (1) /* call bmulti */
{
b_regs.x.ax = 1; /* assume no proc id obtained yet */
}
}

```

```

if ((b_regs.x.bx = ProcId) != 0)      /* if we have a proc id */
    b_regs.x.ax = 2;                  /* tell bmulti that */
b_regs.x.ax += 2FCODE;
int86x (BTR2_INT, &b_regs, &b_regs, &b_sregs);
if ((b_regs.x.ax & 0X00FF) == 0) break; /* if call was processed */
                                        /* by bmulti, leave loop */
b_regs.x.ax = 0x0200; /* if multilink advanced is loaded, it will */
int86x (0x7F, &b_regs, &b_regs, &b_sregs); /* it will switch processes */
}
if (ProcId == 0) ProcId = b_regs.x.bx;

*DATA_LEN = XDATA.BUF_LEN;
return (STAT);                          /* return status */
}

```

Nombre de la interfaz:	TURBBTRV.PAS
Lenguaje(s) que la utiliza(n):	Borland Turbo Basic

```

REM-----
REM
REM This routine interfaces between Borland's Turbo Basic and
REM Strieve on PC-DOS machines. Turbo Basic is compiled.
REM
REM To use, use the metastatement $include:
REM      $INCLUDE "TURBBTRV.BAS"
REM
REM Calling Procedure from Turbo Basic
REM CALL BTRV (OP%, STATUS%, POSITION$, FILEBUF$, DATALEN%,
REM           KEYBUF$, KEYNUM%) where
REM   OP% - function number for request
REM   STATUS% - return status from request
REM   POSITION$ - 128 byte data area for FCB and positioning,
REM             not to be changed by user
REM   FILEBUF$ - address of data buffer from FIELDING statement
REM   DATALEN% - length of user data buffer
REM   KEYBUF$ - string pointer to key buffer
REM   KEYNUM% - key number to be processed
REM
REM Important notes:
REM This interface changes the data segment which is used by BLOAD, BSAVE,
REM CALL ABSOLUTE, PEEK, and POKE, because it does a DEF SEG. Thus, if
REM these commands are used after a call to BTRV, it is important to do
REM a DEF SEG after the BTRV call and before using the above statements.
REM
REM Unlike some BASIC interpreters, Turbo Basic checks that the number and
REM type of the arguments to a procedure (such as BTRV) agree with the
REM formal parameters.
REM
REM Static variables are not visible outside the subroutine, so it is
REM not possible for another part of the application to change them.
REM They must be static because their values must be retained between
REM calls to BTRV.
REM
REM Note that all string parameters must be within the string segment.
REM Vaxptr cannot be used to get the address of the string parameters.
REM It returns a pointer to the string descriptor, of which the first two
REM bytes are the length and the next two bytes are the offset of the
REM string WITHIN the default string segment. The value of the default
REM string segment is stored in the default segment:0. The default segment
REM can be obtained by doing a DEF SEG with no arguments.
REM
REM History:
REM   5/22/87- New for Turbo Basic.
REM
REM-----
SUB BTRV (OP%, STATUS%, POSITION$, FILEB$, DATALEN%, KEYBUF$, KEYNUM%)
STATIC VSET%      : REM these variables must be static because
STATIC MULTI%    : REM values must be retained between calls
STATIC PROCID%   : REM to BTRV
LOCAL PARMBLOCK% : REM local parameter block
DIM A% (14)      : REM dimension param block as integer array
                 : REM save error vector variables
LOCAL ERRVECO1A%, ERRVECO1B%, ERRVECO2A%, ERRVECO2B%
LOCAL STRINGSSEG% : REM string segment
LOCAL TEMP%      : REM temporary for byte to word conversions
                 : REM define indices within parm block
%databufo = 0   : %databufs = 1
%dblen      = 2
%posoff    = 3   : %posseg  = 4
%fcboff   = 5   : %fcboseg = 6
%tbop     = 7
%keybufo  = 8   : %keybufs = 9           : %keynum = 10

```

```

%statoff = 11 : %statseg = 12
%varid = 13
%poslen = 128 : %poslenerr = 23 : %btrloderr = 20
%axreg = 1 : %bxreg = 2 : %dxreg = 4
%dsreg = 8

DEF SEG = 0 : REM must save basic's disk error
ERRVCO1A% = PEEK (6H90)
ERRVCO1B% = PEEK (6H91)
ERRVCO2A% = PEEK (6H92)
ERRVCO2B% = PEEK (6H93)
POKE 6H90, PEEK (6H51A)
POKE 6H91, PEEK (6H51B) : REM and set DOS to handle fatal
POKE 6H92, PEEK (6H51C) : REM errors
POKE 6H93, PEEK (6H51D) : REM ensure Btrieve has been loaded

REG %axreg, 6H357B : REM by using fct 35, get vector
CALL INTERRUPT 6H21 : REM with Btrieve's interrupt
: REM if Btrieve's code offset, ok

IF REG (%bxreg) = 6H33, THEN
IF VSET% = 0, THEN
INCR VSET% : REM if version not set, set it
REG %axreg, 6H3000 : REM get DOS version number

CALL INTERRUPT 6H21
IF (REG (%axreg) AND 6H00FF) < 3, THEN
ELSE : REM if major version >= 3
REG %axreg, 6HAB00
CALL INTERRUPT 6H2F : REM check for bmulti being loaded
IF ((REG (%axreg) AND 6H00FF) = 77), THEN
LET MULTI% = 1 : REM yes, bmulti
ELSE
LET MULTI% = 0
END IF : REM set bmulti flag
END IF : REM bmulti possible only if DOS above 3.x
END IF : REM set ver of Btrieve loaded (multi or not)

DEF SEG
CALL SETSSEG (STRINGSEG%)

LET A%(%bop) = OP% : REM set function number
LET A%(%statoff) = VARPTR (STATUS%) : REM status addr
LET A%(%statseg) = VARSEG (STATUS%)
IF (LEN (POSITION%) < %poslen), THEN : REM check pos block len
STATUS% = %poslenerr
EXIT SUB
END IF

DEF SEG = VARSEG (POSITION%) : REM fcb addr
CALL SETWORD (A%(%fcbloff), POSITION%)
LET A%(%fcbseg) = STRINGSEG% : REM position block

LET A%(%posoff) = A%(%fcbloff) + 38
LET A%(%posseg) = STRINGSEG%

DEF SEG = VARSEG (FILEB%) : REM data buffer addr
CALL SETWORD (A%(%databufo), FILEB%)
LET A%(%databufa) = STRINGSEG%

LET A%(%dblenn) = DATALEN% : REM data buf len

DEF SEG = VARSEG (KEYBUF%) : REM key buffer address
CALL SETWORD (A%(%keybufo), KEYBUF%)
LET A%(%keybufa) = STRINGSEG%

CALL SETKEY (A%(%keynum), KEYNUM%, KEYBUF%) : REM key len, number

LET A%(%varid) = 6H6176 : REM variable id

REG %dsreg, VARSEG (A%(0)) : REM set up ds:dx = data segment
REG %dxreg, VARPTR (A%(0)) : REM and offset => param block

```

```

IF (MULTI% = 0), THEN
CALL INTERRUPT 6H7B : REM call Btrieve
ELSE : REM bmulti loaded
DO : REM make request
IF PROCID% = 0, THEN
REG %axreg, 6HAB01 : REM set function to get proc id
ELSE
REG %bxreg, PROCID%
REG %axreg, 6HAB02 : REM set function, have proc id
END IF
CALL INTERRUPT 6H2F : REM call bmulti
IF ( ((REG (%axreg) AND 6H00FF) = 0) ), THEN
EXIT LOOP : REM request processed by bmulti
END IF
REG %axreg, 6H200 : REM not processed,
CALL INTERRUPT 6H7F : REM allow task switch
LOOP : REM and make request again

IF (PROCID% = 0), THEN
LET PROCID% = REG (%bxreg) : REM set procid
END IF : REM didn't have before
END IF
DATALEN% = A*(2) : REM set up new db len
ELSE
STATUS% = %btrloderr : REM Btrieve not loaded
END IF

DEF SEG = 0 : REM restore error vector
POKE 6H90,ERRVECO1A%
POKE 6H91,ERRVECO1B%
POKE 6H92,ERRVECO2A%
POKE 6H93,ERRVECO2B%
END SUB

SUB SETSSEG INLINE 'set string segment from default ds:0 to parm
' current ds must be tbasic default ds
' setsseg (integer to hold stringseg)
$INLINE 6H55 'push bp
$INLINE 6H99, 6HE5 'mov bp,sp
$INLINE 6HC4, 6H7E, 6H06 'les di,[bp+06]
$INLINE 6HBB, 6H00, 6H00 'mov bx,0
$INLINE 6HB8, 6H07 'mov ax,[bx]
$INLINE 6H26, 6H99, 6H05 'es: mov [di],ax
$INLINE 6H5D 'pop bp
END SUB

SUB SETWORD INLINE 'set array entry with offset in parm
' set word (array entry, variable addr)
$INLINE 6H55 'push bp
$INLINE 6H99, 6HE5 'mov bp,sp
$INLINE 6HBB, 6H5E, 6H06 'mov bx,[bp+06] offset of source
$INLINE 6HB8, 6H46, 6H08 'mov ax,[bp+08] segment of source
$INLINE 6H8E, 6HC0 'mov es,ax
$INLINE 6H26 'es:
$INLINE 6HB8, 6H47, 6H02 'mov ax,[bx+2]
$INLINE 6HC4, 6H7E, 6H0A 'les di,[bp+0A] array element
$INLINE 6H26, 6H99, 6H05 'es: mov [di],ax
$INLINE 6H5D 'pop bp
END SUB

SUB SETKEY INLINE 'set key length/number array element
' setkey (array entry, keynum, keybuf)
$INLINE 6H55 'push bp
$INLINE 6H99, 6HE5 'mov bp,sp
$INLINE 6HC4, 6H7E, 6H0A 'les di,[bp+0a] key number pointer
$INLINE 6H26, 6HBB, 6H0D 'mov cx,es:[di] key number value
$INLINE 6HC4, 6H7E, 6H06 'les di,[bp+06] off,seg of key buf
$INLINE 6H26, 6H9B, 6H05 'mov ax,es:[di] key length
$INLINE 6H88, 6HCC 'mov ah,cl
$INLINE 6HC4, 6H7E, 6H0E 'les di,[bp+0e] off,seg of array entry
$INLINE 6H26, 6H99, 6H05 'mov es:[di],ax
$INLINE 6H5D 'pop bp
END SUB

```

A continuación se presentan los programas para generar los archivos objeto de las interfaces restantes:

Nota: los programas se presentan en BASIC y pueden ser ejecutados en cualquier intérprete de BASIC o compilador de Microsoft.

Nombre de la interfaz:	BASXBTRV.OBJ
Lenguaje(s) que la utiliza(n):	Microsoft Quick Basic V2.0 y otros compiladores compatibles con Gwbasic. Compatibilidad específica con la instrucción FCB%=VARPTR(#1) Donde "#1" representa a un archivo.

Ver ejemplo del subcapítulo 5.03 para revisar el procedimiento de llamado.

```

3 REM ESTE PROGRAMA SE PUEDE EJECUTAR EN CUALQUIER BASIC DE MICROSOFT.
4 REM DESDE INTERPRETES O COMPILADORES, INCLUSIVE QBASIC DE DOS
5 SUMA = 0
10 READ ARCH$
20 READ TBYTES%
25 PRINT "ESTE PROGRAMA CREA LA INTERFAZ GENERICA DE BASIC. ESTA PERMITE"
27 PRINT "TRABAJAR CON BTRIEVE."
28 PRINT "ARCHIVO A CREAR: "; ARCH$
29 PRINT
30 OPEN "R", 1, ARCH$, 1
40 FIELD 1, 1 AS BYTE$
50 FOR B% = 1 TO TBYTES%
60 READ CH%
65 SUMA = SUMA + CH%
70 LSET BYTE$ = CHR$(CH%)
80 PUT 1, B%
90 NEXT B%
92 READ SUMAR
95 IF SUMAR <> SUMA THEN PRINT "ARCHIVO MAL CREADO. LOS NUMEROS SE CAPTURARON
MAL" ELSE PRINT "PROCESO CONCLUIDO."
97 CLOSE
98 END
100 DATA BASXBTRV.OBJ, 784
110 DATA 128,9,0,7,66,84,82,73,69,86,69,95,150,13,0,0,4
120 DATA 67,79,68,69,5,88,70,65,67,69,210,152,7,0,104,197,1
130 DATA 3,2,1,45,160,8,0,1,0,0,235,38,0,0,70,160,8
140 DATA 0,1,6,0,0,0,0,0,81,160,6,0,1,12,0,0,0
150 DATA 77,160,8,0,1,16,0,0,0,0,0,71,160,8,0,1,22
160 DATA 0,0,0,0,0,65,160,173,1,1,28,0,0,0,0,0
170 DATA 0,0,0,0,0,0,0,46,199,6,0,0,1,0,6,30,85
180 DATA 30,7,14,31,139,236,30,43,219,142,219,139,191,144,0,139,183
190 DATA 146,0,139,143,26,5,137,143,144,0,139,143,28,5,137,143,146
200 DATA 0,31,6,83,184,123,53,205,33,131,251,51,91,7,116,12,139
210 DATA 94,18,184,20,0,38,137,7,233,0,0,46,137,62,0,0,86
220 DATA 46,131,62,0,0,0,117,30,46,255,6,0,0,80,184,0,48
230 DATA 205,33,60,3,88,114,14,184,0,171,205,47,60,77,117,5,46
240 DATA 255,6,0,0,139,118,20,38,139,12,46,137,14,0,0,139,78
250 DATA 18,46,137,14,0,0,46,140,6,0,0,139,118,16,38,139,28
260 DATA 141,71,51,46,163,0,0,46,140,6,0,0,141,71,89,46,163
270 DATA 0,0,46,140,6,0,0,141,135,188,0,46,163,0,0,46,140
280 DATA 6,0,0,139,94,14,38,139,7,46,163,0,0,139,94,12,38
290 DATA 139,23,46,131,62,0,0,0,117,3,182,0,75,46,136,22,0
300 DATA 0,38,139,71,2,46,163,0,0,46,140,6,0,0,139,118,10

```

310 DATA	38	,139,12	,46	,136,14	,0	,0	,46	,199,6	,0	,0	,118,97	,141,22				
320 DATA	0	,0	,46	,131,62	,0	,0	,0	,116,44	,51	,192,64	,46	,139,30	,0			
330 DATA	0	,11	,219,116	,1	,64	,180,171	,205,47	,60	,0	,116,7	,184,0	,2				
340 DATA	205	,127,235	,228,46	,131,62	,0	,0	,0	,117,10	,46	,137,30	,0	,0				
350 DATA	235	,3	,144,205	,123,46	,161,0	,0	,0	,139,94	,14	,38	,137,7	,6	,46			
360 DATA	196	,54	,0	,0	,38	,139,4	,7	,94	,46	,139,62	,0	,0	,30	,43	,219	
370 DATA	142	,219,137	,191,144	,0	,137,183	,146,0	,31	,46	,161,0	,0	,61	,5				
380 DATA	0	,114,5	,61,14	,0	,117,49	,43	,192,180	,48	,205,33	,61	,3	,0				
390 DATA	117	,38	,46	,131,62	,0	,0	,0	,116,30	,46	,139,54	,0	,0	,139,254			
400 DATA	131	,239,50	,131,199	,12	,131,198	,12	,46	,142,6	,0	,0	,30	,6	,31			
410 DATA	185	,25	,0	,243,164	,31	,93	,31	,7	,202,12	,0	,176	,156	,239,0	,196		
420 DATA	15	,0	,1	,1	,30	,0	,132,82	,0	,1	,1	,114,1	,196,87	,0	,1		
430 DATA	1	,32	,0	,196,93	,0	,1	,1	,38	,0	,196,101	,0	,1	,1	,38	,0	
440 DATA	196	,126,0	,1	,1	,36	,0	,196,137	,0	,1	,1	,16	,0	,196,145	,0		
450 DATA	1	,1	,24	,0	,196,150	,0	,1	,1	,26	,0	,196,163	,0	,1	,1	,12	
460 DATA	0	,196,168	,0	,1	,1	,14	,0	,196,175	,0	,1	,1	,8	,0	,196,180		
470 DATA	0	,1	,1	,10	,0	,196,188	,0	,1	,1	,2	,0	,196,193	,0	,1	,1	
480 DATA	4	,0	,196,203	,0	,1	,1	,6	,0	,196,214	,0	,1	,1	,30	,0	,196	
490 DATA	225	,0	,1	,1	,22	,0	,196,233	,0	,1	,1	,18	,0	,196,238	,0	,1	
500 DATA	1	,20	,0	,196,249	,0	,1	,1	,23	,0	,196,254	,0	,1	,1	,28	,0	
510 DATA	197	,4	,0	,1	,1	,2	,0	,197,9	,0	,1	,1	,36	,0	,197,20	,0	
520 DATA	1	,1	,34	,0	,197,45	,0	,1	,1	,34	,0	,197,53	,0	,1	,1	,34	
530 DATA	0	,197,62	,0	,1	,1	,6	,0	,197,74	,0	,1	,1	,24	,0	,197,84		
540 DATA	0	,1	,1	,32	,0	,197,102	,0	,1	,1	,16	,0	,197,128	,0	,1	,1	
550 DATA	30	,0	,197,136	,0	,1	,1	,12	,0	,197,152	,0	,1	,1	,14	,0	,155	
560 DATA	144	,11	,0	,0	,1	,4	,66	,84	,82	,86	,40	,0	,0	,250,138	,2	,0
570 DATA	0	,116														
580 DATA	47360															

Nota: La(s) línea(s) que no cuenta(n) con número de línea forman parte del renglón superior.

Nombre de la interfaz:	BC7RBTRV.OBJ
Lenguaje(s) que la utiliza(n):	Microsoft BASIC V 7.0

Procedimiento de llamado:

BTRVFAR (Op%, Status%, PosBlk\$, VARPTR (DataBuf), VARSEG (DataBuf),
DLen%, KeyBuf\$, KeyNum%)

6

BTRVFAR (Op%, Status%, PosBlk\$, SADD (DataBuf\$), SSEG (DataBuf\$),
DLen%, KeyBuf\$, KeyNum%)

Nota:

1. Se utiliza VARPTR/VARSEG cuando el buffer de datos esta definido por un registro de variables utilizando TYPE --- END TYPE.
2. Se utiliza SADD/SSEG cuando el buffer de datos esta definido por la instruccion FIELD.
3. Las cadenas que se utilicen en la funcion deben ser de longitud fija ejem: DIM A AS STRING * 10

```

3 REM ESTE PROGRAMA SE PUEDE EJECUTAR EN CUALQUIER BASIC DE MICROSOFT.
4 REM DESDE INTERPRETES O COMPILADORES, INCLUSIVE QBASIC DE DOS
5 SUMA = 0
10 READ ARCH$
20 READ TBYTES%
22 CLS
25 PRINT "ESTE PROGRAMA CREA LA INTERFAZ CON Microsoft BASIC V7.0. ESTA PERMITE"
27 PRINT "TRABAJAR CON BTRIEVE."
28 PRINT "ARCHIVO A CREAR: "; ARCH$
29 PRINT
30 OPEN "R", 1, ARCH$, 1
40 FIELD 1, 1 AS BYTE$
50 FOR B% = 1 TO TBYTES%
60 READ CH%
65 SUMA = SUMA + CH%
70 LSET BYTE$ = CHR$(CH%)
80 PUT 1, B%
90 NEXT B%
92 READ SUMAR
95 IF SUMAR <> SUMA THEN PRINT "ARCHIVO MAL CREADO. LOS NUMEROS SE CAPTURARON
MAL" ELSE PRINT "PROCESO CONCLUIDO."
97 CLOSE
98 END
100 DATA BC7RBTRV.OBJ, 869
110 DATA 128,9 ,0 ,7 ,66 ,84 ,82 ,73 ,69 ,86 ,69 ,95 ,150,25 ,0 ,0 ,4
120 DATA 67 ,79 ,68 ,69 ,4 ,68 ,65 ,84 ,65 ,5 ,88 ,70 ,65 ,67 ,69 ,6 ,95
130 DATA 68 ,65 ,84 ,65 ,88 ,209,152,7 ,0 ,104,38 ,0 ,5 ,3 ,1 ,202,152
140 DATA 7 ,0 ,104,142,1 ,4 ,2 ,1 ,99 ,140,30 ,0 ,13 ,83 ,84 ,82 ,73
150 DATA 78 ,71 ,65 ,68 ,68 ,82 ,69 ,83 ,83 ,0 ,12 ,83 ,84 ,82 ,73 ,78 ,71
160 DATA 76 ,69 ,78 ,71 ,84 ,72 ,0 ,199,160,6 ,0 ,1 ,0 ,0 ,0 ,0 ,89
170 DATA 160,8 ,0 ,1 ,4 ,0 ,0 ,0 ,0 ,0 ,83 ,160,6 ,0 ,1 ,10 ,0
180 DATA 0 ,0 ,79 ,160,8 ,0 ,1 ,14 ,0 ,0 ,0 ,0 ,0 ,73 ,160,8 ,0
190 DATA 1 ,20 ,0 ,0 ,0 ,0 ,0 ,67 ,160,16 ,0 ,1 ,26 ,0 ,0 ,0 ,0
200 DATA 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,53 ,160,146,1 ,2 ,0 ,0 ,80
210 DATA 30 ,184,0 ,0 ,142,216,88 ,163,0 ,0 ,88 ,80 ,83 ,81 ,82 ,87 ,86
220 DATA 6 ,85 ,161,0 ,0 ,142,192,139,236,51 ,219,38 ,139,191,144,0 ,38
230 DATA 139,183,146,0 ,38 ,139,143,26 ,5 ,38 ,137,143,144,0 ,38 ,139,143
240 DATA 28 ,5 ,38 ,137,143,146,0 ,6 ,184,123,53 ,205,33 ,131,251,51 ,7
250 DATA 116,17 ,139,94 ,32 ,6 ,142,6 ,0 ,0 ,38 ,199,7 ,20 ,0 ,7 ,233

```

260 DATA	0	0	137,62	0	0	86	131,62	0	0	0	117,26	255,6	0	
270 DATA	0	184,0	48	205,33	60	3	114,13	184,0	171,205,47	60	77			
280 DATA	117,4	255,6	0	0	139,118,34	38	139,12	137,14	0	0	139			
290 DATA	78	32	137,14	0	0	140,6	0	0	30	139,94	30	6	31	83
300 DATA	154,0	0	0	0	60	128	114,12	83	154,0	0	0	0	142	194
310 DATA	139,216	235,17	31	142,6	0	0	196,30	0	0	38	199,7	23		
320 DATA	0	233,0	0	31	141,7	163,0	0	140,6	0	0	141,71	38		
330 DATA	163,0	0	140,6	0	0	142,6	0	0	139,94	28	38	139,7		
340 DATA	163,0	0	139,94	26	38	139,7	163,0	0	139,94	24	38	139		
350 DATA	7	163,0	0	139,94	22	30,6	31	83	154,0	0	0	0	31	
360 DATA	142,6	0	0	163,0	0	137,22	0	0	186,255,0	136,22	0			
370 DATA	0	139,118,20	38	139,12	136,14	0	0	199,6	0	0	118,97			
380 DATA	6	141,22	0	0	131,62	0	0	116,41	51	192,64	139,30			
390 DATA	0	0	11	219,116,1	64	180,171,205,47	60	0	116,7	184,0				
400 DATA	2	205,127	235,229,131,62	0	0	0	117,9	137,30	0	0	235			
410 DATA	3	144,205,123,7	161,0	0	139,94	24	38	137,7	6	196,54				
420 DATA	0	0	38	139,4	7	94	139,62	0	0	51	219,38	137,191,144		
430 DATA	0	38	137,183,146,0	93	161,0	0	142,216,7	94	95	90	89			
440 DATA	91	88	252,202,16	0	251,156,20	1	200,3	4	1	1	196,9			
450 DATA	0	1	1	36,0	196,21	0	1	1	36,0	196,77	0	1	1	1
460 DATA	36	0	132,86	0	2	2	113,1	196,90	0	1	1	28	0	196
470 DATA	95	0	1	1	34,0	196,102,0	1	1	34,0	196,124,0	1			
480 DATA	1	32	0	196,134,0	1	1	14,0	196,141,0	1	1	22,0			
490 DATA	196,145,0	1	1	24,0	204,155,38	2	2	204,165,38	1	1	1			
500 DATA	196,178,0	1	1	36,0	196,182,0	1	1	22,0	132,190,0					
510 DATA	2	2	108,1	196,196,0	1	1	10,0	196,200,0	1	1	12			
520 DATA	0	196,206,0	1	1	6,0	196,210,0	1	1	8,0	196,214				
530 DATA	0	1	1	36,0	196,223,4	1	1	196,232,0	1	1	2,0			
540 DATA	196,241,0	1	1	4,0	204,251,38	1	1	197,2,0	1	1	1			
550 DATA	36	0	197,5	0	1	1	16,0	197,9,0	1	1	18,0	197		
560 DATA	16	0	1	1	20,0	197,26,0	1	1	21,0	197,30,0	1			
570 DATA	1	26,0	197,37,4	1	1	197,41,0	1	1	32,0	197,51				
580 DATA	0	1	1	30,0	197,75,0	1	1	30,0	197,82,0	1	1			
590 DATA	30	0	197,91,0	1	1	4,0	197,102,0	1	1	22,0	197			
600 DATA	111,0	1	1	28,0	197,127,0	1	1	36,0	59,144,14,0					
610 DATA	0	2	7	66,84	82	86,70	65	82,0	0	0	66	138,2	0	
620 DATA	0	116												
630 DATA	51712													

Nombre de la interfaz:	QB6RBTRV.OBJ
Lenguaje(s) que la utiliza(n):	Microsoft BASIC v6.0 Microsoft QuickBasic v4.0, v4.5, v6.0 IBM BASIC v1.0, v2.0, v3.0

METODO POR LA INSTRUCCION "FIELD":

```

-----
OPEN "NUL" AS #1 LEN=40
FIELD #1,20 AS FNAME$, 10 AS FTEL$, 2 AS FFLD$, 8 AS FFBAL$
OP% = 0
POSBLK$ = SPACE$ (128)
KEYBUF$ = "MYFILE.BTR "
KEYNUM% = 0
FCB% = SADD (FNAME$)
BUFLN% = 40
CALL BTRV (OP%, STAT%, POSBLK$, FCB%, BUFLN%, KEYBUF$, KEYNUM%)

```

METODO POR LA INSTRUCCION "TYPE":

```

-----
TYPE DATABUFFER
NAME AS STRING * 20
TEL AS STRING * 10
ID AS INTEGER
BAL AS DOUBLE
END TYPE
DIM FIL1 AS DATABUFFER
OP% = 0
POSBLK$ = SPACE$ (128)
KEYBUF$ = "MYFILE.BTR "
KEYNUM% = 0
FCB% = VARPTR (FIL1)
'ó = VARPTR (FIL1.NAME)
BUFLN% = 40
CALL BTRV (OP%, STAT%, POSBLK$, FCB%, BUFLN%, KEYBUF$, KEYNUM%)

```

```

3 REM ESTE PROGRAMA SE PUEDE EJECUTAR EN CUALQUIER BASIC DE MICROSOFT.
4 REM DESDE INTERPRETES O COMPILADORES, INCLUSIVE QBASIC DE DOS
5 SUMA = 0
10 READ ARCH$
20 READ TBYTES%
22 CLS
25 PRINT "ESTE PROGRAMA CREA LA INTERFAZ PARA QBASIC V6. ESTA PERMITE"
27 PRINT "TRABAJAR CON BTRIEVE."
28 PRINT "ARCHIVO A CREAR: "; ARCH$
29 PRINT
30 OPEN "R", 1, ARCH$, 1
40 FIELD 1, 1 AS BYTE$
50 FOR B% = 1 TO TBYTES%
60 READ CH%
65 SUMA = SUMA + CH%
70 LSET BYTE$ = CHR$(CH%)
80 PUT 1, B%
90 NEXT B%
92 READ SUMAR
95 IF SUMAR <> SUMA THEN PRINT "ARCHIVO MAL CREADO. LOS NUMEROS SE CAPTURARON
MAL" ELSE PRINT "PROCESO CONCLUIDO."
97 CLOSE
98 END
100 DATA QB6RBTRV.OBJ, 951
110 DATA 128,9 ,0 ,7 ,66 ,84 ,82 ,73 ,69 ,86 ,69 ,95 ,150,25 ,0 ,0 ,4

```

120 DATA	67	,79	,68	,69	,4	,68	,65	,84	,65	,5	,88	,70	,65	,67	,69	,6	,95
130 DATA	68	,65	,84	,65	,88	,209	,152	,7	,0	,104	,41	,0	,5	,3	,1	,199	,152
140 DATA	7	,0	,104	,194	,1	,4	,2	,1	,47	,160	,6	,0	,1	,0	,0	,0	,0
150 DATA	89	,160	,8	,0	,1	,4	,0	,0	,0	,0	,0	,83	,160	,6	,0	,1	,10
160 DATA	0	,0	,0	,79	,160	,8	,0	,1	,14	,0	,0	,0	,0	,0	,73	,160	,8
170 DATA	0	,1	,20	,0	,0	,0	,0	,0	,67	,160	,19	,0	,1	,26	,0	,0	,0
180 DATA	0	,0	,0	,0	,0	,0	,0	,0	,0	,0	,0	,0	,0	,50	,160	,198	,1
190 DATA	2	,0	,0	,80	,30	,184	,0	,0	,142	,216	,88	,163	,0	,0	,88	,198	,6
200 DATA	0	,0	,0	,80	,83	,81	,82	,87	,86	,6	,85	,161	,0	,0	,142	,192	,139
210 DATA	236	,128	,62	,0	,0	,0	,116	,2	,69	,69	,199	,6	,0	,0	,1	,0	,43
220 DATA	219	,38	,139	,191	,144	,0	,38	,139	,183	,146	,0	,38	,139	,143	,26	,5	,38
230 DATA	137	,143	,144	,0	,38	,139	,143	,28	,5	,38	,137	,143	,146	,0	,6	,184	,123
240 DATA	53	,205	,33	,131	,251	,51	,7	,116	,17	,139	,94	,30	,6	,142	,6	,0	,0
250 DATA	38	,199	,7	,20	,0	,7	,233	,0	,0	,137	,62	,0	,0	,86	,131	,62	,0
260 DATA	0	,0	,117	,26	,255	,6	,0	,0	,184	,0	,48	,205	,33	,60	,3	,114	,13
270 DATA	184	,0	,171	,205	,47	,60	,77	,117	,4	,255	,6	,0	,0	,139	,118	,32	,38
280 DATA	139	,12	,137	,14	,0	,0	,139	,78	,30	,137	,14	,0	,0	,140	,6	,0	,0
290 DATA	139	,118	,28	,38	,139	,4	,131	,62	,0	,0	,0	,116	,1	,70	,70	,60	,128
300 DATA	115	,12	,196	,30	,0	,0	,38	,199	,7	,23	,0	,233	,0	,0	,38	,139	,28
310 DATA	141	,7	,163	,0	,0	,140	,6	,0	,0	,141	,71	,38	,163	,0	,0	,140	,6
320 DATA	0	,0	,139	,94	,26	,38	,139	,7	,163	,0	,0	,128	,62	,0	,0	,0	,116
330 DATA	14	,77	,77	,139	,94	,26	,38	,139	,7	,163	,0	,0	,235	,7	,144	,140	,195
340 DATA	137	,30	,0	,0	,139	,94	,24	,38	,139	,7	,163	,0	,0	,139	,94	,22	,186
350 DATA	255	,0	,131	,62	,0	,0	,0	,117	,3	,182	,0	,75	,136	,22	,0	,0	,38
360 DATA	139	,71	,2	,163	,0	,0	,140	,6	,0	,0	,139	,118	,20	,38	,139	,12	,136
370 DATA	14	,0	,0	,199	,6	,0	,0	,118	,97	,6	,141	,22	,0	,0	,131	,62	,0
380 DATA	0	,0	,116	,41	,51	,192	,64	,139	,30	,0	,0	,11	,219	,116	,1	,64	,180
390 DATA	171	,205	,47	,60	,0	,116	,7	,184	,0	,2	,205	,127	,235	,229	,131	,62	,0
400 DATA	0	,0	,117	,9	,137	,30	,0	,0	,235	,3	,144	,205	,123	,7	,161	,0	,0
410 DATA	139	,94	,24	,38	,137	,7	,6	,196	,54	,0	,0	,38	,139	,4	,7	,94	,139
420 DATA	62	,0	,0	,43	,219	,38	,137	,191	,144	,0	,38	,137	,183	,146	,0	,93	,128
430 DATA	62	,0	,0	,0	,161	,0	,0	,142	,216	,7	,94	,95	,90	,89	,91	,88	,252
440 DATA	116	,3	,202	,16	,0	,202	,14	,0	,80	,30	,184	,0	,0	,142	,216	,88	,163
450 DATA	0	,0	,88	,198	,6	,0	,0	,1	,233	,0	,0	,38	,156	,66	,1	,200	,3
460 DATA	4	,1	,1	,196	,9	,0	,1	,1	,39	,0	,196	,14	,0	,1	,1	,38	,0
470 DATA	196	,26	,0	,1	,1	,39	,0	,196	,34	,0	,1	,1	,38	,0	,196	,43	,0
480 DATA	1	,1	,28	,0	,196	,97	,0	,1	,1	,39	,0	,132	,106	,0	,2	,2	,135
490 DATA	1	,196	,110	,0	,1	,1	,30	,0	,196	,115	,0	,1	,1	,36	,0	,196	,122
500 DATA	0	,1	,1	,36	,0	,196	,144	,0	,1	,1	,34	,0	,196	,154	,0	,1	,1
510 DATA	14	,0	,196	,161	,0	,1	,1	,22	,0	,196	,165	,0	,1	,1	,24	,0	,196
520 DATA	175	,0	,1	,1	,28	,0	,196	,188	,0	,1	,1	,22	,0	,132	,196	,0	,2
530 DATA	2	,130	,1	,196	,204	,0	,1	,1	,10	,0	,196	,208	,0	,1	,1	,12	,0
540 DATA	196	,214	,0	,1	,1	,6	,0	,196	,218	,0	,1	,1	,8	,0	,196	,227	,4
550 DATA	1	,1	,196	,231	,0	,1	,1	,38	,0	,196	,245	,0	,1	,1	,2	,0	,196
560 DATA	254	,0	,1	,1	,2	,0	,197	,7	,0	,1	,1	,4	,0	,197	,17	,0	,1
570 DATA	1	,28	,0	,197	,27	,0	,1	,1	,20	,0	,197	,34	,0	,1	,1	,16	,0
580 DATA	197	,38	,0	,1	,1	,18	,0	,197	,48	,0	,1	,1	,21	,0	,197	,52	,0
590 DATA	1	,1	,26	,0	,197	,59	,4	,1	,1	,197	,63	,0	,1	,1	,34	,0	,197
600 DATA	73	,0	,1	,1	,32	,0	,197	,97	,0	,1	,1	,32	,0	,197	,104	,0	,1
610 DATA	1	,32	,0	,197	,113	,0	,1	,1	,4	,0	,197	,124	,0	,1	,1	,22	,0
620 DATA	197	,133	,0	,1	,1	,30	,0	,197	,150	,0	,1	,1	,38	,0	,197	,154	,0
630 DATA	1	,1	,39	,0	,201	,177	,4	,1	,1	,197	,183	,0	,1	,1	,39	,0	,197
640 DATA	188	,0	,1	,1	,38	,0	,133	,192	,0	,2	,2	,17	,0	,254	,144	,11	,0
650 DATA	0	,2	,4	,66	,84	,82	,86	,0	,0	,0	,33	,144	,14	,0	,0	,2	,7
660 DATA	66	,84	,82	,86	,70	,65	,82	,174	,1	,0	,147	,138	,2	,0	,0	,116	,0
670 DATA	56064																

Nombre de la interfaz:	COBXBTRV.OBJ
Lenguaje(s) que la utiliza(n):	Microsoft COBOL IBM COBOL

Procedimiento de llamado:

CALL 'BTRV' USING B-OPEN,B-STATUS, POSITION-BLOCK, DATA-BUFFER, BUF-LEN,
KEY-BUFFER, NAME-KEY.

```

3 REM ESTE PROGRAMA SE PUEDE EJECUTAR EN CUALQUIER BASIC DE MICROSOFT.
4 REM DESDE INTERPRETES O COMPILADORES, INCLUSIVE QBASIC DE DOS
5 SUMA = 0
10 READ ARCH$
20 READ TBYTES%
22 CLS
25 PRINT "ESTE PROGRAMA CREA LA INTERFAZ GENERICA DE COBOL. ESTA PERMITE"
27 PRINT "TRABAJAR CON BTRIEVE."
28 PRINT "ARCHIVO A CREAR: "; ARCH$
29 PRINT
30 OPEN "R", 1, ARCH$, 1
40 FIELD 1, 1 AS BYTE$
50 FOR B% = 1 TO TBYTES%
60   READ CH%
65   SUMA = SUMA + CH%
70   LSET BYTE$ = CHR$(CH%)
80   PUT 1, B%
90 NEXT B%
92 READ SUMAR
95 IF SUMAR <> SUMA THEN PRINT "ARCHIVO MAL CREADO. LOS NUMEROS SE CAPTURARON
MAL" ELSE PRINT "PROCESO CONCLUIDO."
97 CLOSE
98 END
100 DATA COBXBTRV.OBJ, 761
110 DATA 128,9,0,7,66,84,82,73,69,86,69,95,150,8,0,0,5
120 DATA 95,85,83,69,82,191,152,7,0,32,193,1,2,1,1,123,160
130 DATA 8,0,1,0,0,235,34,0,0,74,160,8,0,1,6,0,0
140 DATA 0,0,0,81,160,6,0,1,12,0,0,0,77,160,8,0,1
150 DATA 16,0,0,0,0,0,71,160,8,0,1,22,0,0,0,0,0
160 DATA 65,160,169,1,1,28,0,0,0,0,0,0,0,0,0,6,30
170 DATA 85,30,7,14,31,139,236,6,83,184,123,53,205,33,131,251,51
180 DATA 91,7,116,14,139,94,20,184,20,0,134,224,38,137,7,233,0
190 DATA 0,46,131,62,0,0,0,117,30,46,255,6,0,0,80,184,0
200 DATA 48,205,33,60,3,88,114,14,184,0,171,205,47,60,77,117,5
210 DATA 46,255,6,0,0,139,118,22,38,139,12,134,205,46,137,14,0
220 DATA 0,139,78,20,46,137,14,0,0,46,140,6,0,0,139,94,18
230 DATA 141,7,46,163,0,0,46,140,6,0,0,141,71,38,46,163,0
240 DATA 0,46,140,6,0,0,139,94,16,46,137,30,0,0,46,140,6
250 DATA 0,0,139,118,14,38,139,12,134,205,46,137,14,0,0,139,94
260 DATA 12,46,137,30,0,0,46,140,6,0,0,46,198,6,0,0,255
270 DATA 139,118,10,38,139,12,46,136,46,0,0,46,199,6,0,0,118
280 DATA 97,141,22,0,0,232,0,0,46,131,62,0,0,0,116,44,51
290 DATA 192,64,46,139,30,0,0,11,219,116,1,64,180,171,205,47,60
300 DATA 0,116,7,184,0,2,205,127,235,228,46,131,62,0,0,0,117
310 DATA 10,46,137,30,0,0,235,3,144,205,123,232,0,0,46,139,14
320 DATA 0,0,134,205,139,94,14,38,137,15,139,94,20,38,139,7,134
330 DATA 224,38,137,7,93,31,7,202,14,0,46,131,62,0,0,15,116
340 DATA 16,46,131,62,0,0,14,116,8,46,131,62,0,0,31,117,104
350 DATA 80,83,81,82,86,30,46,139,22,0,0,46,197,30,0,0,83
360 DATA 185,8,0,139,7,134,196,137,7,67,67,226,246,94,139,76,4
370 DATA 128,253,0,116,2,134,205,131,249,24,119,52,83,131,235,16,43

```

380 DATA	192,64	,131,195,16	,131,250,14	117,9	,247,71	,4	,0	,16	117,240
390 DATA	235,7	,247,71	,4	,16,0	,117,231,226,229,91	,139,200,176,8	,246		
400 DATA	225,139,200,139,7	,134,196,137,7	,67,67	,226,246,31	,94,90	,89			
410 DATA	91,88	,195,27	,156,225,0	,132,43,0	,1,1	,47,1	,196,48	,0	
420 DATA	1,1	,34,0	,196,56,0	,1,1	,34,0	,196,81,0	,1,1	,32	
430 DATA	0	,196,94,0	,1,1	,16,0	,196,102,0	,1,1	,24,0	,196,107	
440 DATA	0	,1,1	,26,0	,196,116,0	,1,1	,12,0	,196,121,0	,1,1	
450 DATA	14,0	,196,128,0	,1,1	,8,0	,196,133,0	,1,1	,10,0	,196	
460 DATA	141,0	,1,1	,2,0	,196,146,0	,1,1	,4,0	,196,159,0	,1	
470 DATA	1,6	,0	,196,167,0	,1,1	,18,0	,196,172,0	,1,1	,20,0	
480 DATA	196,177,0	,1,1	,22,0	,196,189,0	,1,1	,23,0	,196,194,0		
490 DATA	1,1	,28,0	,196,200,0	,1,1	,2,0	,132,203,0	,1,1	,72	
500 DATA	1,196,208,0	,1,1	,32,0	,196,219,0	,1,1	,30,0	,196,244		
510 DATA	0,1,1	,30,0	,196,252,0	,1,1	,30,0	,133,4	,0,1,1		
520 DATA	64,1	,197,9	,0,1,1	,6,0	,197,39,0	,1,1	,16,0	,197	
530 DATA	47,0	,1,1	,16,0	,197,55,0	,1,1	,16,0	,197,69,0	,1	
540 DATA	1,16,0	,197,74,0	,1,1	,2,0	,20,144,11,0	,0,1,4			
550 DATA	66,84,82,86,0	,0,0	,0,34	,138,2,0	,0,116				
560 DATA	48640								

Nombre de la interfaz:	COBRBTRV.OBJ
Lenguaje(s) que la utiliza(n):	Micro Focus COBOL /2 Microsoft COBOL V3.0 Relia COBOL

```

3 REM ESTE PROGRAMA SE PUEDE EJECUTAR EN CUALQUIER BASIC DE MICROSOFT.
4 REM DESDE INTERPRETES O COMPILADORES, INCLUSIVE QBASIC DE DOS
5 SUMA = 0
10 READ ARCH$
20 READ TBYTES%
25 CLS:PRINT "ESTE PROGRAMA CREA LA INTERFAZ DE MS COBOL V3.0. ESTA PERMITE"
27 PRINT "TRABAJAR CON BTRIEVE."
28 PRINT "ARCHIVO A CREAR: "; ARCH$
29 PRINT
30 OPEN "R", 1, ARCH$, 1
40 FIELD 1, 1 AS BYTE$
50 FOR B% = 1 TO TBYTES%
60 READ CH%
65 SUMA = SUMA + CH%
70 LSET BYTE$ = CHR$(CH%)
80 PUT 1, B%
90 NEXT B%
92 READ SUMAR
95 IF SUMAR <> SUMA THEN PRINT "ARCHIVO MAL CREADO. LOS NUMEROS SE CAPTURARON
MAL" ELSE PRINT "PROCESO CONCLUIDO."
97 CLOSE
98 END
100 DATA COBRBTRV.OBJ, 511
110 DATA 128,9,0,7,66,84,82,73,69,86,69,95,150,25,0,0,4
120 DATA 67,79,68,69,4,68,65,84,65,5,95,84,69,88,84,6,95
130 DATA 88,68,65,84,65,148,152,7,0,72,38,0,5,3,1,234,152
140 DATA 7,0,72,25,1,4,2,1,248,160,16,0,1,26,0,0,0
150 DATA 0,0,0,0,0,0,0,0,0,0,53,160,29,1,2,0,0
160 DATA 30,186,0,0,142,218,141,30,0,0,90,137,23,141,30,0,0
170 DATA 140,7,203,141,30,0,0,142,7,141,30,0,0,139,23,142,218
180 DATA 203,141,54,0,0,184,123,53,205,33,131,251,51,116,5,184,0
190 DATA 0,235,38,131,62,0,0,0,117,28,255,6,0,0,80,184,0
200 DATA 48,205,33,60,3,88,114,13,184,0,171,205,47,60,77,117,4
210 DATA 255,6,0,0,184,1,0,203,85,139,236,154,0,0,0,0,154
220 DATA 0,0,0,0,11,192,117,12,184,20,0,196,94,10,38,137,7
230 DATA 233,0,0,196,94,6,38,139,15,137,76,14,196,94,10,137,92
240 DATA 22,140,68,24,196,94,14,141,7,137,68,10,140,68,12,141,71
250 DATA 38,137,68,6,140,68,8,196,94,18,137,28,140,68,2,196,94
260 DATA 22,38,139,15,137,76,4,196,94,26,137,92,16,140,68,18,198
270 DATA 68,20,255,196,94,30,38,139,15,136,76,21,199,6,0,0,118
280 DATA 97,139,214,131,62,0,0,0,116,41,51,192,64,139,30,0,0
290 DATA 11,219,116,1,64,180,171,205,47,60,0,116,7,184,0,2,205
300 DATA 127,235,229,131,62,0,0,0,117,9,137,30,0,0,235,3,144
310 DATA 205,123,139,76,4,196,94,22,38,137,15,196,92,22,38,139,7
320 DATA 154,0,0,0,0,139,229,93,203,60,156,121,0,200,2,4,1
330 DATA 1,196,8,0,1,1,34,0,196,15,0,1,1,36,0,196,22
340 DATA 0,1,1,36,0,196,28,0,1,1,34,0,196,37,4,1,1
350 DATA 196,56,0,1,1,32,0,196,63,0,1,1,32,0,196,87,0
360 DATA 1,1,30,0,204,97,4,2,2,204,102,0,2,2,35,0,132
370 DATA 120,0,2,2,16,1,196,201,0,1,1,26,0,196,209,0,1
380 DATA 1,30,0,196,219,0,1,1,28,0,196,243,0,1,1,28,0
390 DATA 196,250,0,1,1,28,0,205,17,0,2,2,20,0,96,144,12
400 DATA 0,0,2,5,95,66,84,82,86,93,0,0,99,138,2,0,0
410 DATA 116
420 DATA 34304

```

Nombre de la interfaz:	FORXBTRV.OBJ
Lenguaje(s) que la utiliza(n):	Microsoft Fortran

Procedimiento de llamado:

```

PROGRAM TESTIT
CHARACTER POSITN(128)*1
CHARACTER DATA(300)*1,KEYBUF*25
ISTAT = ITRV(0,POSITN,DATA,IDLLEN,'DATA',0)
WRITE(*,*)'ISTAT = ',ISTAT
100 CONTINUE
WRITE(*,*)'ENTER DATA'
READ(*,*)IOP,KEY,KEYBUF
IDLLEN = 300
ISTAT = ITRV(IOP,POSITN,DATA,IDLLEN,KEYBUF,KEY)
WRITE(*,*)'ISTAT=',ISTAT
WRITE(*,*)'IDLLEN=',IDLLEN
WRITE(*,*)(DATA(I),I=1,IDLLEN)
GO TO 100
ISTAT = ITRV(1,POSITN,DATA,IDLLEN,KEYBUF,0)
END

```

```

3 REM ESTE PROGRAMA SE PUEDE EJECUTAR EN CUALQUIER BASIC DE MICROSOFT.
4 REM DESDE INTERPRETES O COMPILADORES INCLUSIVE QBASIC DE DOS
5 SUMA = 0
10 READ ARCH$
20 READ TBYTES%
25 PRINT "ESTE PROGRAMA CREA LA INTERFAZ GENERICA DE FORTRAN. ESTA PERMITE"
27 PRINT "TRABAJAR CON BTBRIEVE."
28 PRINT "ARCHIVO A CREAR: "; ARCH$
29 PRINT
30 OPEN "R", 1, ARCH$, 1
40 FIELD 1, 1 AS BYTES$
50 FOR B% = 1 TO TBYTES%
60 READ CH%
65 SUMA = SUMA + CH%
70 LSET BYTES$ = CHR$(CH%)
80 PUT 1, B%
90 NEXT B%
100 READ SUMAR
110 IF SUMAR <> SUMA THEN PRINT "ARCHIVO MAL CREADO. LOS NUMEROS SE CAPTURARON MAL" ELSE PRINT "PROCESO CONCLUIDO."
120 CLOSE
130 END
1000 DATA FORXBTRV.OBJ, 790
1010 DATA 129,9 ,0 ,7 ,66 ,84 ,82 ,73 ,69 ,86 ,69 ,95 ,150,13 ,0 ,0 ,4
1020 DATA 67 ,79 ,68 ,69 ,5 ,88 ,70 ,65 ,67 ,69 ,210,152,7 ,0 ,104,20 ,1
1030 DATA 3 ,2 ,1 ,222,160,6 ,0 ,1 ,0 ,0 ,0 ,89 ,160,8 ,0 ,1
1040 DATA 4 ,0 ,0 ,0 ,0 ,0 ,83 ,160,6 ,0 ,1 ,10 ,0 ,0 ,0 ,79 ,160
1050 DATA 8 ,0 ,1 ,14 ,0 ,0 ,0 ,0 ,0 ,73 ,160,8 ,0 ,1 ,20 ,0 ,0
1060 DATA 0 ,0 ,0 ,67 ,160,254,0 ,1 ,26 ,0 ,0 ,0 ,0 ,0 ,0 ,0
1070 DATA 0 ,0 ,0 ,85 ,30 ,6 ,139,236,14 ,31 ,6 ,83 ,184 ,123,53 ,205,33
1080 DATA 131,251,51 ,91 ,7 ,116,9 ,199,6 ,0 ,0 ,20 ,0 ,233,0 ,0 ,131
1090 DATA 62 ,0 ,0 ,0 ,0 ,117,28 ,255,6 ,0 ,0 ,80 ,184,0 ,48 ,205,33 ,60
1100 DATA 3 ,88 ,114,13 ,184,0 ,171,205,47 ,60 ,77 ,117,4 ,255,6 ,0 ,0
1110 DATA 196,118,30 ,38 ,139,52 ,137,54 ,0 ,0 ,141,14 ,0 ,0 ,137,14 ,0
1120 DATA 0 ,140,30 ,0 ,0 ,196,94 ,26 ,141,7 ,163,0 ,0 ,141,71 ,38 ,163
1130 DATA 0 ,0 ,140,6 ,0 ,0 ,140,6 ,0 ,0 ,196,94 ,22 ,137,30 ,0 ,0
1140 DATA 140,6 ,0 ,0 ,6 ,196,94 ,18 ,38 ,139,7 ,163,0 ,0 ,7 ,196,94

```

1150 DATA	14	,137,30	0	0	,140,6	0	0	,198,6	0	0	,255,196,118,10						
1160 DATA	38	,139,28	,136,30	0	0	,199,6	0	0	,118,97	,141,22	0	0					
1170 DATA	131,62	0	0	0	,116,41	,51	,192,64	,139,30	0	0	,11	,219,116					
1180 DATA	1	,64	,180	,171,205,47	,60	0	,116,7	,184,0	,2	,205,127,235,229							
1190 DATA	131,62	0	0	0	,117,9	,137,30	0	0	,235,3	,144,205,123,6							
1200 DATA	196,94	,18	,161,0	0	,38	,137,7	,7	,161,0	0	,186,0	0	,7					
1210 DATA	31,93	,202,24	0	28	,156,193,0	,196,33	0	,1	,1	,28	0	,132					
1220 DATA	38	0	,1	,1	,8	,1	,196,42	0	,1	,1	,34	0	,196,49	0	,1		
1230 DATA	1	,34	0	,196,73	0	,1	,1	,32	0	,196,83	0	,1	,1	,14	0		
1240 DATA	196,87	0	,1	,1	,28	0	,196,91	0	,1	,1	,22	0	,196,95	0			
1250 DATA	1	,1	,24	0	,196,103	0	,1	,1	,10	0	,196,109	0	,1	,1	,6		
1260 DATA	0	,196,113	0	,1	,1	,12	0	,196,117	0	,1	,1	,8	0	,196,124			
1270 DATA	4	,1	,1	,196,128	0	,1	,1	,2	0	,196,138	0	,1	,1	,4	0		
1280 DATA	196,146	0	,1	,1	,16	0	,196,150	0	,1	,1	,18	0	,196,154	0			
1290 DATA	1	,1	,20	0	,196,165	0	,1	,1	,21	0	,196,169	0	,1	,1	,26		
1300 DATA	0	,196,175	,4	,1	,1	,196,179	0	,1	,1	,32	0	,196,189	0	,1			
1310 DATA	1	,30	0	,196,213	0	,1	,1	,30	0	,196,220	0	,1	,1	,30	0		
1320 DATA	196,232	0	,1	,1	,4	0	,196,239	0	,1	,1	,28	0	,227,144	,11			
1330 DATA	0	0	,1	,4	,73	,84	,82	,86	,36	0	,247,144	,20	0	0	,1		
1340 DATA	13	,85	,83	,69	,82	,95	,66	,85	,70	,95	,65	,68	,68	,82	0	0	0
1350 DATA	89	,144,19	0	0	,1	,12	,85	,83	,69	,82	,95	,66	,85	,70	,95	,76	
1360 DATA	69	,78	,4	0	0	,147,144	,20	0	0	,1	,13	,85	,83	,69	,82	,95	
1370 DATA	67	,85	,82	,95	,65	,68	,68	,82	,6	0	0	,70	,144	,20	0	0	,1
1380 DATA	13	,85	,83	,69	,82	,95	,70	,67	,66	,95	,65	,68	,68	,82	,10	0	0
1390 DATA	97	,144,20	0	0	,1	,13	,85	,83	,69	,82	,95	,70	,85	,78	,67	,84	
1400 DATA	73	,79	,78	,14	0	0	,60	,144,20	0	0	,1	,13	,85	,83	,69	,82	
1410 DATA	95	,75	,69	,89	,95	,65	,68	,68	,82	,16	0	0	,61	,144	,22	0	0
1420 DATA	1	,15	,85	,83	,69	,82	,95	,75	,69	,89	,95	,76	,69	,78	,71	,84	,72
1430 DATA	20	0	0	,142,144	,22	0	0	,1	,15	,85	,83	,69	,82	,95	,75	,69	
1440 DATA	89	,95	,78	,85	,77	,66	,69	,82	,21	0	0	,134,144	,21	0	0	,1	
1450 DATA	14	,85	,83	,69	,82	,95	,83	,84	,65	,84	,95	,65	,68	,68	,82	,22	0
1460 DATA	0	,226,144	,15	0	0	,1	,8	,88	,70	,65	,67	,69	,95	,73	,68	,26	
1470 DATA	0	0	,235,138	,2	0	0	,116										
1480 DATA	45312																

Nombre de la interfaz:	PASXBTRV.OBJ
Lenguaje(s) que la utiliza(n):	IBM Pascal

Procedimiento de llamado:

STAT := BTRV (OP, POS.START, DATA.START, DATALEN, KBUF.START, KEY);

```

3 REM ESTE PROGRAMA SE PUEDE EJECUTAR EN CUALQUIER BASIC DE MICROSOFT.
4 REM ESTE INTERPRETES O COMPILADORES, INCLUSIVE QBASIC DE DOS
5 SUMA = 0
10 READ ARCH$
20 READ TBYTES%
22 CLS
25 PRINT "ESTE PROGRAMA CREA LA INTERFAZ GENERICA DE PASCAL. ESTA PERMITE"
27 PRINT "TRABAJAR CON BTRIEVE."
28 PRINT "ARCHIVO A CREAR: "; ARCH$
29 PRINT
30 OPEN "R", 1, ARCH$, 1
40 FIELD 1, 1 AS BYTE$
50 FOR B% = 1 TO TBYTES%
60 READ CH%
65 SUMA = SUMA + CH%
70 LSET BYTE$ = CHR$(CH%)
80 PUT 1, B%
90 NEXT B%
92 READ SUMAR
95 IF SUMAR <> SUMA THEN PRINT "ARCHIVO MAL CREADO. LOS NUMEROS SE CAPTURARON
MAL" ELSE PRINT "PROCESO CONCLUIDO."
97 CLOSE
98 END
100 DATA PASXBTRV.OBJ, 614
110 DATA 128,9 ,0 ,7 ,66 ,84 ,82 ,73 ,69 ,86 ,69 ,95 ,150,13 ,0 ,0 ,4
120 DATA 67 ,79 ,68 ,69 ,5 ,88 ,70 ,65 ,67 ,69 ,210,152,7 ,0 ,104,59 ,1
130 DATA 3 ,2 ,1 ,183,160,6 ,0 ,1 ,0 ,0 ,0 ,0 ,89 ,160,8 ,0 ,1
140 DATA 4 ,0 ,0 ,0 ,0 ,0 ,83 ,160,6 ,0 ,1 ,10 ,0 ,0 ,0 ,79 ,160
150 DATA 8 ,0 ,1 ,14 ,0 ,0 ,0 ,0 ,0 ,73 ,160,8 ,0 ,1 ,20 ,0 ,0
160 DATA 0 ,0 ,0 ,67 ,160,37 ,1 ,1 ,26 ,0 ,0 ,0 ,0 ,0 ,0 ,0
170 DATA 0 ,0 ,0 ,85 ,30 ,6 ,139,236,14 ,31 ,6 ,83 ,184,123,53 ,205,33
180 DATA 131,251,51 ,91 ,7 ,116,10 ,46 ,199,6 ,0 ,0 ,20 ,0 ,233,0 ,0
190 DATA 46 ,131,62 ,0 ,0 ,0 ,117,34 ,46 ,199,6 ,0 ,0 ,1 ,0 ,80 ,184
200 DATA 0 ,48 ,205,33 ,60 ,3 ,88 ,114,16 ,184,0 ,171,205,47 ,60 ,77 ,117
210 DATA 7 ,46 ,199,6 ,0 ,0 ,1 ,0 ,139,118,34 ,46 ,137,54 ,0 ,0 ,141
220 DATA 14 ,0 ,0 ,46 ,137,14 ,0 ,0 ,46 ,140,30 ,0 ,0 ,139,70 ,32 ,61
230 DATA 100,0 ,115,10 ,46 ,199,6 ,0 ,0 ,23 ,0 ,233,0 ,0 ,196,94 ,28
240 DATA 141,7 ,46 ,163,0 ,0 ,0 ,141,71 ,38 ,46 ,163,0 ,0 ,46 ,140,6 ,0
250 DATA 0 ,46 ,140,6 ,0 ,0 ,196,94 ,28 ,46 ,137,30 ,0 ,0 ,46 ,140,6
260 DATA 0 ,0 ,6 ,196,94 ,18 ,38 ,139,7 ,46 ,163,0 ,0 ,7 ,196,94 ,12
270 DATA 46 ,137,30 ,0 ,0 ,46 ,140,6 ,0 ,0 ,139,94 ,16 ,46 ,136,30 ,0
280 DATA 0 ,139,94 ,10 ,46 ,136,30 ,0 ,0 ,46 ,199,6 ,0 ,0 ,118,97 ,141
290 DATA 22 ,0 ,0 ,46 ,131,62 ,0 ,0 ,0 ,116,43 ,51 ,192,64 ,46 ,139,30
300 DATA 0 ,0 ,11 ,219,116,1 ,64 ,180,171,205,47 ,60 ,0 ,116,7 ,184,0
310 DATA 2 ,205,127,235,228,46 ,131,62 ,0 ,0 ,0 ,117,9 ,46 ,137,30 ,0
320 DATA 0 ,235,2 ,205,123,6 ,46 ,161,0 ,0 ,196,94 ,18 ,38 ,137,7 ,7
330 DATA 46 ,161,0 ,0 ,7 ,31 ,93 ,202,26 ,0 ,115,156,207,0 ,196,34 ,0
340 DATA 1 ,1 ,28 ,0 ,132,39 ,0 ,1 ,1 ,49 ,1 ,196,44 ,0 ,1 ,1 ,34
350 DATA 0 ,196,52 ,0 ,1 ,1 ,34 ,0 ,196,79 ,0 ,1 ,1 ,32 ,0 ,196,89
360 DATA 0 ,1 ,1 ,14 ,0 ,196,93 ,0 ,1 ,1 ,28 ,0 ,196,98 ,0 ,1 ,1
370 DATA 22 ,0 ,196,103,0 ,1 ,1 ,24 ,0 ,196,116,0 ,1 ,1 ,28 ,0 ,132
380 DATA 121,0 ,1 ,1 ,49 ,1 ,196,130,0 ,1 ,1 ,10 ,0 ,196,137,0 ,1
390 DATA 1 ,6 ,0 ,196,142,0 ,1 ,1 ,12 ,0 ,196,147,0 ,1 ,1 ,8 ,0
400 DATA 196,155,4 ,1 ,1 ,196,160,0 ,1 ,1 ,2 ,0 ,196,171,0 ,1 ,1

```

410 DATA	4	,0	,196,180,0	,1	,1	,16,0	,196,185,0	,1	,1	,18,0	,196
420 DATA	193,0	,1	,1,20,0	,196,201,0	,1	,1,21,0	,196,206,0	,1			
430 DATA	1,26	,0	,196,212,4	,1	,1	,196,217,0	,1	,1	,32,0	,196,228	
440 DATA	0,1	,1	,30,0	,196,253,0	,1	,1,30,0	,197,5,0	,1			
450 DATA	30,0	,197,14,0	,1	,1,4,0	,197,25,0	,1	,1,28,0	,96			
460 DATA	144,11	,0	,0,1,4,66,84,82,86,36,0	,0	,254,138,2	,0					
470 DATA	0,116										
480 DATA	33536										

APÉNDICE E

PROGRAMA PARA FACILITAR LA CREACIÓN DE ARCHIVOS BTRIEVE

Este apéndice es un complemento del subcapítulo 2.06, donde se discute la forma en la que un archivo Btrieve se genera por primera vez para ser utilizado. Las formas de crear un archivo Btrieve que se presentaron en el subcapítulo mencionado son un tanto complicadas por lo que la intención de este apéndice es facilitar este punto tan importante. En el subcapítulo 2.06 se presentó el uso de la utilidad BUTIL -CREATE, que permite crear un archivo Btrieve, lo que requiere esta utilidad es un *archivo de descripción* que contiene la definición del archivo a crear, pues bien, en esta parte se presenta un programa que facilita la creación del archivo de descripción de una forma más amigable y sin recordar tantas palabras clave.

A continuación se presenta todo lo necesario, para poner en marcha y utilizar este programa auxiliar llamado CREADES.BAS.

Para utilizar este programa se requiere de la utilidad QBASIC.EXE que se encuentra a partir de la versión 5 del sistema operativo (Se eligió este lenguaje por ser de los más comunes y accesibles) . Pasos a seguir:

- 1.- Entre a QBASIC.
- 2.- Capture el código que aparece al final del anexo.
- 3.- Guarde el programa con el nombre de CREADES.BAS
- 4.- Para correr el programa desde QBASIC oprima [Shift]+[F5].
- 5.- Para correr el programa directamente desde DOS digite la siguiente instrucción:

```
C:\DOS\>QBASIC /RUN CREADES.BAS
```

A continuación se presenta una pequeña descripción del programa auxiliar:

Al ejecutar el programa aparecerá la siguiente pantalla:

Encabezado	(F1) o (^A) = AYUDA		Fecha: 88-28-1995																																																																																																									
	GENERACION DEL ARCHIVO DE DESCRIPCION		BTRIEVE VER 5.X. UTILERIA APENDICE E																																																																																																									
Definición de Datos	Largo del Registro :200		Tamaño de Página :1024																																																																																																									
	Longitud de Registro Variable:n		Pre-asignación de Página :0																																																																																																									
Titulos	Truncación en Blanco:n		Reemplazar archivo Existente:y																																																																																																									
	Compresión de Datos:n		Incluir Datos :y																																																																																																									
	Cantidad de Líneas :?		Espacio Libre :10																																																																																																									
	<table border="1"> <thead> <tr> <th>Pos</th> <th>Long</th> <th>Dup</th> <th>Modif</th> <th>Tipo</th> <th>Dese</th> <th>Alte</th> <th>Manua</th> <th>Nula</th> <th>UaMu</th> <th>Segme</th> <th>Arch.</th> <th>Alt</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2</td> <td>n</td> <td>y</td> <td>aut</td> <td>n</td> <td>n</td> <td>n</td> <td>n</td> <td>0</td> <td>n</td> <td></td> <td></td> </tr> <tr> <td>3</td> <td>10</td> <td>n</td> <td>y</td> <td>str</td> <td>n</td> <td>n</td> <td>n</td> <td>n</td> <td>0</td> <td>n</td> <td></td> <td></td> </tr> <tr> <td>13</td> <td>40</td> <td>n</td> <td>y</td> <td>str</td> <td>n</td> <td>n</td> <td>n</td> <td>n</td> <td>0</td> <td>n</td> <td></td> <td></td> </tr> <tr> <td>199</td> <td>2</td> <td>y</td> <td>y</td> <td>int</td> <td>n</td> <td>n</td> <td>n</td> <td>n</td> <td>0</td> <td>y</td> <td></td> <td></td> </tr> <tr> <td>3</td> <td>10</td> <td>y</td> <td>y</td> <td>str</td> <td>n</td> <td>n</td> <td>n</td> <td>n</td> <td>0</td> <td>n</td> <td></td> <td></td> </tr> <tr> <td>199</td> <td>2</td> <td>y</td> <td>y</td> <td>int</td> <td>n</td> <td>n</td> <td>n</td> <td>n</td> <td>0</td> <td>y</td> <td></td> <td></td> </tr> <tr> <td>13</td> <td>40</td> <td>y</td> <td>y</td> <td>str</td> <td>n</td> <td>n</td> <td>n</td> <td>n</td> <td>0</td> <td>n</td> <td></td> <td></td> </tr> </tbody> </table>					Pos	Long	Dup	Modif	Tipo	Dese	Alte	Manua	Nula	UaMu	Segme	Arch.	Alt	1	2	n	y	aut	n	n	n	n	0	n			3	10	n	y	str	n	n	n	n	0	n			13	40	n	y	str	n	n	n	n	0	n			199	2	y	y	int	n	n	n	n	0	y			3	10	y	y	str	n	n	n	n	0	n			199	2	y	y	int	n	n	n	n	0	y			13	40	y	y	str	n	n	n	n	0	n	
Pos	Long	Dup	Modif	Tipo	Dese	Alte	Manua	Nula	UaMu	Segme	Arch.	Alt																																																																																																
1	2	n	y	aut	n	n	n	n	0	n																																																																																																		
3	10	n	y	str	n	n	n	n	0	n																																																																																																		
13	40	n	y	str	n	n	n	n	0	n																																																																																																		
199	2	y	y	int	n	n	n	n	0	y																																																																																																		
3	10	y	y	str	n	n	n	n	0	n																																																																																																		
199	2	y	y	int	n	n	n	n	0	y																																																																																																		
13	40	y	y	str	n	n	n	n	0	n																																																																																																		
Definición de Llaves																																																																																																												
	Llave Segmentada (y/n)																																																																																																											
Descripción de Llaves																																																																																																												

Definición de Datos

En esta sección se indican los datos generales del archivo Btrieve a crear.

Para navegar entre la sección de Definición de Datos utilice las flechas de dirección del cursor (← ↑ → ↓), si desea pasar a la sección de Definición de Llaves, oprima [Tab], para regresar a la sección anterior oprima nuevamente [Tab].

Al teclear un dato completo, el cursor pasará automáticamente al siguiente dato, si el valor de éste no está dentro del rango que se necesita, el programa toma el valor por omisión para este dato y pasa al siguiente.

El dato "Cantidad de Líneas" se refiere al número de líneas para definir las llaves y no al número de llaves, ya que las llaves segmentadas necesitan más de una línea de descripción.

Se debe dar un valor para el dato "Cantidad de líneas" para poder pasar a la sección de descripción de llaves. Este valor puede ser modificado aún cuando ya se hayan descrito las llaves, pero se debe tener cuidado ya que aún cuando en la pantalla se visualice un número de líneas, en el archivo sólo se grabará el número que esté especificado en "Cantidad de líneas".

Definición de Llaves

En esta sección se presenta una tabla en renglones y columnas donde se debe indicar las características para cada llave. Cada llave o segmento de llave se representa en un renglón.

La descripción de los títulos de cada columna es la siguiente:

Pos	=	Posición de la Llave.
Long	=	Longitud de la Llave.
Dup	=	Aceptará Llaves Duplicadas.
Modif	=	Permitirá modificar el valor de Llave.
Tipo	=	Tipo de la Llave.
Dese	=	Ordenación Descendente.
Alte	=	Se alterará el orden de la tabla ASCII para la ordenación
Manua	=	Llave manual, cualidad para indexar o no la Llave.
Nula	=	Llave nula, cualidad para que NO indexe Llaves nulas.
VaNu	=	Valor nulo de Llave.
Segme	=	Llave Segmentada.
Arch. Alt	=	Nombre del Archivo que contiene la modificación a la tabla ASCII.

Cuando se entra en esta sección de deben ingresar las características para cada llave en los renglones de la tabla, al finalizar un renglón y presionar [Enter] el programa da el espacio para un nuevo renglón siempre y cuando no se haya llegado al límite de líneas.

Si desea insertar un renglón entre dos líneas, seleccione el renglón y posicione el cursor en la primer columna, presione [INS] y después [Enter], automáticamente se insertará la nueva línea debajo de la seleccionada por el cursor.

Si desea borrar una línea, posicione el cursor en el renglón que quiere borrar y presione [Ctrl]+[Y], automáticamente la línea desaparecerá y subirá las líneas restantes.

Ayuda

Si necesita ayuda sobre algún dato oprima [F1] o [Ctrl]+[A] y aparecerá una ventana con información general acerca del dato donde esté posicionado el cursor, independientemente si se encuentra en la sección de definición de datos o definición de llaves. La ayuda consiste en una descripción general del dato y en ocasiones el rango válido para el mismo.

La ayuda aparece de la siguiente forma en pantalla:

GENERACION DEL ARCHIVO DE DESCRIPCION

BTRIEVE VER 5.X. UTILERIA APENDICE E

Largo del Registro :200	Tamaño de Página :1024
Longitud de Registro Variable:n	Pre-asignación de Página :0
Truncación en Blanco:n	Reemplazar archivo Existente:y
Compresión de Datos:n	Incluir Datos :y
Cantidad de Líneas :7	Espacio Libre :10

Pos	Long	Du	Arch. Alt
1	2	n	
3	10	n	
13	40	n	
199	2	y	
3	10	y	
199	2	y	y
13	40	y	y

El largo del registro define la longitud de los registros lógicos en bytes, este valor corresponde al largo en bytes del registro incluyendo las llaves.
 - Presione [Return] o [Esc] -

y	str	n	n	n	n	n	n	0	y
y	int	n	n	n	n	n	n	0	y
y	str	n	n	n	n	n	n	0	n

Llave Segmentada <y/n>

Para terminar

Al terminar de definir la descripción de las llaves y cuando todos los datos estén correctos, oprima [Ctrl]+[End], el programa solicitará el nombre del archivo de descripción en el cual se grabará la información, digite el nombre y oprima [Enter], saldrá al sistema operativo o al interprete guardando el archivo de descripción con el nombre que se escribió. Después de generar el archivo solo resta poner en marcha a Btrieve y utilizar la utilería BUTIL para que genere el archivo Btrieve.

Si desea abortar la operación y salir del programa sin grabar el archivo oprima entonces [Ctrl]+[Q], regresará al sistema operativo sin grabar el archivo.

Para cualquier duda adicional sobre los datos que pide el programa o el uso de BUTIL consulte el capítulo II.

NOTA: Los datos que aparecen en la pantalla de ejemplo corresponden a los datos para generar el archivo de descripción de ejemplo que se define en el subcapítulo 2.06 y que se requiere para ejecutar los programas del capítulo IV.

A continuación se presenta el código completo del programa CREADES.BAS.

```

DECLARE SUB QBrest (r%, C%, LR%, LC%, f%)
DECLARE SUB QBSAVE (r%, C%, LR%, LC%, f%)

10 KEY 1, CHR$(1)
20 ulx% = 0: ULY% = 4: DRX% = 79: DRY% = 24: LNS% = 17
30 RESTORE 4000
40 B$ = CHR$(29)

50 ' Definición de colores
60 fore1 = 7: back1 = 1: FORE5 = 7: BACK5 = 1: FORE3 = 7: BACK3 = 1: fore2 = 15: back2 = 4:
fore4 = 1
70 DIM CA$(30, 1), TIPO$(30, 6, 1), F$(30, 0), A$(30), RG$(0, 0), OP$(10), TIPO$(30, 1, 1),
CA$(1), B$(30, 15), RGB$(30), RE$(19)
80 DEF FNC$(A$) = CHR$(ASC(A$) + (32 * (A$ >= "a" AND A$ <= "z")))
90 DEF FNM$(A$) = CHR$(ASC(A$) - (32 * (A$ >= "A" AND A$ <= "Z"))): DEF FNM$(A$) =
MID$(STR$(A%), 2)

300 RESTORE 4000
310 FOR X = 0 TO 1
320 READ CA$(X)
330 FOR I% = 1 TO CA$(X): READ CA$(I%, X), TIPO$(I%, 0, X), TIPO$(I%, 1, X), TIPO$(I%, 2,
X), TIPO$(I%, 3, X), TIPO$(I%, 4, X), TIPO$(I%, 0, X), TIPO$(I%, 5, X), TIPO$(I%, 6, X),
TIPO$(I%, 1, X): NEXT I%
340 NEXT X

1000 GOSUB 3800 ' Inicialización de Pantalla
1010 GOSUB 10000
1020 COLOR fore1, back1
1030 LOCATE 11, 1: PRINT CHR$(195) + STRING$(78, 196) + CHR$(180):
1040 LOCATE 12, 1: PRINT CHR$(179) + "Pos Long Dup Modif Tipo Dese Alte Manua Nula
VaNu Segme Arch. Alt " + CHR$(179):
1050 LOCATE 13, 1: PRINT CHR$(195) + STRING$(78, 196) + CHR$(180):
1060 LOCATE 23, 1: PRINT CHR$(195) + STRING$(78, 196) + CHR$(180):
1070 GOSUB 6000 ' Solicitud de datos

1080 ' Generación de Archivo de Descripción
1500 LOCATE 24, 2: PRINT SPACE$(78): : IF R1% = 17 THEN 2000: salida ^Q
1502 C% = 2: r% = 24: Lnt% = 8: RP% = "": RZ% = "INDIQUE EL NOMBRE DEL ARCHIVO: "
1503 LOCATE 24, 42: PRINT ".DES": : GOSUB 12000: ARCHIVO$ = z$
1504 IF ARCHIVO$ = "" THEN ER% = 24: GOSUB 30000: ER% = 0: GOTO 1500
1505 IF INSTR(ARCHIVO$, ".") + INSTR(ARCHIVO$, " ") <> 0 THEN ER% = 24: GOSUB 30000: ER% =
0: GOTO 1500
1510 CLOSE 30: OPEN "O", 30, DIR$ + ARCHIVO$ + ".DES"
1520 FOR I = 1 TO 10
1521 IF I = 5 THEN GOSUB 17000
1522 IF I = 3 THEN IF A$(I) = "n" THEN 1540
1524 IF I = 4 THEN IF A$(I) = "n" THEN 1540
1526 IF I = 7 THEN IF VAL(A$(I)) = 0 THEN 1540
1528 IF I = 10 AND A$(2) = "n" THEN 1540
1530 PRINT #30, TIPO$(I, 0, 0): "": A$(I)
1540 NEXT I
1550 NLLA% = VAL(A$(5)): IF NLLA% = 0 THEN 2000
1560 FOR I = 1 TO AAL0%: FOR J = 1 TO 12
1570 IF J = 10 THEN IF B$(I, 8) = "n" AND B$(I, 9) = "n" THEN 1590
1572 IF J = 12 THEN IF B$(I, 7) = "n" THEN 1590
1580 PRINT #30, TIPO$(J, 0, 1): "": B$(I, J)
1590 NEXT J
1595 NEXT I
2000 GOTO 3500

3500 COLOR 7, 0: CLS : CLOSE : END ' Fin de Programa

3712 COLOR f1%, f2%: RETURN

3800 ' Encabezado
3805 COLOR fore1, back1: CLS
3806 LOCATE 3, 3: PRINT "GENERACION DEL ARCHIVO DE DESCRIPCION BTRIEVE VER 5.X, UTILERIA
APENDICE E":
3810 LOCATE 2, 2: PRINT "+": : PRINT STRING$(76, "-"): : LOCATE 2, 40: PRINT "-": : LOCATE
2, 79: PRINT "+":
3812 LOCATE 3, 2: PRINT "|": : LOCATE 3, 40: PRINT "|": : LOCATE 3, 79: PRINT "|":

```

```

3813 LOCATE 4, 2: PRINT "+"; : PRINT STRING$(76, "-"); : LOCATE 4, 40: PRINT "-": LOCATE 4,
79: PRINT "+";
3814 D$ = DATE$: T$ = TIME$
3815 LOCATE 1, 60: PRINT "Fecha: "; DATE$;
3816 LOCATE 1, 10: PRINT "[F1] o [^A] = AYUDA"
3890 RETURN

4000 'Información Semiconstante
4010 DATA 10
4020 DATA "Largo del Registro :", 6, 3,4,2,2,"record", 0,0,"256"
4030 DATA "Longitud de Registro Variable:", 7, 3,1,1,1,"variable", 0,0,"n"
4040 DATA "Truncación en Blanco:", 8, 3,1,1,1,"truncate", 1,1,"n"
4050 DATA "Compresión de Datos:", 9, 3,1,1,1,"compress", 1,0,"n"
4060 DATA "Cantidad de Líneas :", 10, 3,2,3,2,"key", 0,0,"0"
4070 DATA "Tamaño de Página :", 6,43,4,4,2,"page", 0,0,"512"
4080 DATA "Re-asignación de Página :", 7,43,5,5,2,"allocation",1,0,"0"
4090 DATA "Reemplazar archivo Existente:", 8,43,1,1,1,"replace", 1,0,"y"
4100 DATA "Incluir Datos :", 9,43,1,1,1,"data", 1,0,"y"
4110 DATA "Espacio Libre :", 10,43,2,6,2,"threshold",0,0,"10"
4115 *****
4117 DATA 12
4120 DATA "Posición de la llave (desde 1 hasta el largo del
registro)",16,2,4,10,2,"position",0,0,"1"
4130 DATA "Longitud de la llave (desde 1 hasta el límite del tipo de
llave)",17,9,3,11,2,"length",0,0,"1"
4140 DATA "Se aceptarán valores duplicados (y/n)",18,15,1,12,1,"duplicates",0,0,"n"
4150 DATA "Se permitirá modificar el valor de llave (y/n)",19,21,1,13,1,"modifiable",0,0,"y"
4160 DATA "Tipo de llave",20,27,3, 1,1,"type",0,0,"str"
4170 DATA "Se ordena descendente (y/n)",21,33,1,15,1,"descending",1,0,"n"
4180 DATA "Se alterará el orden de la tabla ASCII para la ordenación
(y/n)",22,39,1,16,1,"alternate",0,0,"n"
4190 DATA "Llave Manual, cualidad para indexar o no la llave
(y/n)", 6,45,1,17,1,"manual",1,19,"n"
4200 DATA "Llave Nula, cualidad para que no indexe llaves nulas
(y/n)", 7,51,1,18,1,"null",0,19,"n"
4210 DATA "Valor Nulo de Llave (cualquier valor hexadecimal)",8,57,2,19,2,"value",0,0,"0"
4220 DATA "Llave Segmentada (y/n)",9,62,1,20,1,"segment",0,0,"n"
4230 DATA "Nombre de Archivo que contiene la modificación a la tabla
Ascii",23,68,12,21,1,"name",0,16,""

4890 rem instrucciones en lenguaje máquina para el Scroll
4890 DATA 85,139,236,139,118,16,138,12,139,118,14,138,44,139,118
4891 DATA 12,138,20,139,118,10,138,52,139,118,8,139,4,128,252
4892 DATA 0,116,6,247,216,180,7,235,2,180,6,139,118,6,138
4893 DATA 60,208,231,208,231,208,231,208,231,205,16,93,202,12,0

4900 'Llamado a las funciones de Scroll del DOS
4902 WHILE SCROLL.FLAG = 0
4903 DIM scroll$(29): scrolls = VARSEG(scroll$(0)): DEF SEG = scrolls
4904 ulx% = 1: DRX% = 78: ATR% = fore4: RESTORE 4890
4905 FOR ISROLL = 0 TO 59
4906 READ JSCROLL
4907 POKE ISROLL + VARPTR(scroll$(0)), JSCROLL
4908 NEXT ISROLL
4909 SCROLL.FLAG = 1
4910 WEND
4911 DEF SEG = scrolls
4912 CALL absolute(ulx%, ULY%, DRX%, DRY%, LNS%, ATR%, VARPTR(scroll$(0)))
4913 DEF SEG
4914 RETURN

6000 'Rutina para pedir los Datos Generales del Archivo
6040 COLOR fore1, back1: f1% = fore1: f2% = back1
6045 F1% = 0
6047 GOSUB 8000
6060 CC% = 1
6070 z% = TIPO%(CC%, 0, 0): C% = TIPO%(CC%, 1, 0): RP% = A$(CC%): LN% = TIPO%(CC%, 2, F1%)
6090 RZ% = CA$(CC%, F1%)
6095 EDITAR% = -10: PAT% = 9: GOSUB 12000
6100 A$(CC%) = z$: TT% = TIPO%(CC%, 3, F1%): ON TT% GOSUB 7000, 7050, 7100, 7150, 7200, 7250
6110 IF R1% <> 17 THEN IF (FLECHA% AND R1% = 13) OR FLECHA% = 0 THEN CC% = CC% + 1: GOTO
6180

```

```

6120 IF FLECHAS% AND RI% = 80 THEN CC% = CC% + 1
6130 IF FLECHAS% AND RI% = 72 THEN CC% = CC% - 1
6140 IF FLECHAS% AND RI% = 9 THEN 6500
6145 IF (FLECHAS% AND RI% = 117) OR RI% = 17 THEN RETURN
6150 IF FLECHAS% AND RI% = 77 THEN CC% = CC% + 5
6160 IF FLECHAS% AND RI% = 75 THEN CC% = CC% - 5
6180 IF CC% > CA%(0) THEN CC% = 1 ELSE IF CC% < 1 THEN CC% = CA%(0)
6190 GOTO 6070
6500 'Rutina que solicita los datos de las Llaves
6520 IF VAL(A$(5)) = 0 THEN ER% = 23: GOSUB 30000: ER% = 0: GOTO 6000 ELSE FI% = 1
6540 FILEV% = DIR% + "PRODUCTO.VEN": RENN% = 0: NUMCOL% = CA%(FI%)
6550 NCOL% = 2: NREN% = 14: LIN% = 8: limite% = VAL(A$(5)): RRC% = 4: GOSUB 14000: RENN% =
0: FILEV% = FV% pide partidas ****
6580 IF RI% = 9 AND FLECHAS% THEN GOTO 6000
6590 IF (RI% = 117 AND FLECHAS%) OR RI% = 17 THEN RETURN
7000 'Validación de Datos
7010 A$(CC%) = FNW$(A$(CC%))
7015 IF A$(CC%) <> "y" AND A$(CC%) <> "n" THEN A$(CC%) = "n"
7020 LOCATE r%, c% + LEN(CA$(CC%, 0)): PRINT A$(CC%): RETURN
7050 'Validación del Largo del Registro
7060 IF VAL(A$(CC%)) < 4 OR VAL(A$(CC%)) > 4090 THEN A$(CC%) = "4"
7070 GOTO 7020
7100 'Validación del Número de Llaves
7110 IF VAL(A$(CC%)) < 0 OR VAL(A$(CC%)) > 25 THEN A$(CC%) = "0"
7120 GOTO 7020
7150 'Validación del Largo de la Página
7160 IF VAL(A$(CC%)) < 512 OR VAL(A$(CC%)) > 4096 THEN A$(CC%) = "512"
7170 GOTO 7020
7200 'Validación de la Pre-colocación
7210 IF VAL(A$(CC%)) < 0 OR VAL(A$(CC%)) > 65535 THEN A$(CC%) = "0"
7220 GOTO 7020
7250 'Validación del Espacio Libre
7260 EL% = VAL(A$(CC%))
7270 IF EL% < 0 OR EL% > 30 THEN A$(CC%) = "10"
7280 GOTO 7020
8000 'Rutina para imprimir las A$( )
8010 FOR CC% = 1 TO CA%(FI%)
8015 IF A$(CC%) = "" THEN A$(CC%) = TIPO$(CC%, 1, FI%)
8020 r% = TIPO$(CC%, 0, FI%): c% = TIPO$(CC%, 1, FI%)
8025 LOCATE r%, c%: PRINT CA$(CC%, FI%); A$(CC%)
8030 NEXT CC%: RETURN
10000 'Rutina para pintar el cuadro de la pantalla
10005 COLOR fore!, back!: LOCATE 5, 1: PRINT "+": PRINT STRING$(78, "-"): LOCATE 5, 80:
PRINT "+";
10010 FOR TYU% = 1 TO 19
10015 LOCATE 5 + TYU%, 1: PRINT "|": LOCATE 5 + TYU%, 80: PRINT "|";
10020 NEXT TYU%
10025 LOCATE 25, 1: PRINT "+": PRINT STRING$(78, "-"): LOCATE 25, 80: PRINT "+";
10900 RETURN
11999 ' Rutina general para editar una respuesta
12000 L2% = 1: IF L2% THEN L2% = L2%: IF L2% > LEN(RP%) THEN L2% = LEN(RP%): IF L2% < 1
THEN L2% = 1
12001 r% = RP%: IF LEN(r%) > LN% THEN r% = LEFT$(r%, LN%)
12002 L1% = LEN(r%): LP% = CHR$(176): LOCATE r%, c%: COLOR FORE!, BACK!: PRINT RZ%: COLOR
FORE!, BACK!: PRINT r% + STRING$(LN% - L1%, LP%): IF c% + LEN(r%) + LN% - L1% < 80 THEN
PRINT " ";
12003 C% = c% + LEN(RZ%) - 1: IF C% < 1 THEN C% = 1
12004 l% = LEN(r%): LOCATE r%, c% + l%, 1: IF EDITAR% THEN FLECHAS% = -1: EDI% = 0
12005 X98% = "": GOSUB 12051
12006 X98% = INKEY$: IF INB% THEN IF MCK% < 6 THEN GOSUB 12048: MCK% = MCK% + 1 ELSE GOSUB
12051: MCK% = 0
12007 IF X98% = "" THEN 12006 ELSE W% = X98%: RI% = ASC(W%): IF RI% > 64 AND RI% < 91 THEN
W% = CHR$(RI% + 32)
12008 IF LEN(W%) <> 1 OR RI% = 10 THEN 12025 ELSE IF PAT2% AND RI% <> 13 AND RI% <> 9 THEN
12006 ELSE IF EDITAR% AND RI% = 25 THEN 12022'FLECHAS
12009 EDI% = -1: IF ASC(W%) = PAT% THEN FLECHAS% = PAT%: RI% = PAT%: GOTO 12022 ELSE IF
ASC(W%) = 1 THEN GOTO 12060
12010 RI% = ASC(W%): IF (RI% < 32 OR RI% > 165) AND RI% <> 17 AND RI% <> 8 AND RI% <> 27 AND
RI% <> 13 THEN 12006

```

```

12011 IF R1% = 8 THEN 12018 ELSE IF R1% = 27 THEN 12020 ELSE IF R1% = 13 THEN 12022 ELSE IF
R1% = 17 THEN 12022
12012 IF L% = L2% - 1 THEN R$ = R$ + W$: GOSUB 12048: L% = L% + 1: L2% = L2% + 1: IF L% <
LN% THEN GOSUB 12051: GOTO 12021 ELSE L2% = LN% + 1: GOTO 12021'12400 revisa si lla se acabo
el campo
12013 R$ = LEFT$(R$, L2% - 1) + W$ + MID$(R$, L2% + 1 - INS%) : GOSUB 12048
12014 L2% = L2% + 1: L% = L% + INS%
12015 IF INS% = 0 AND L2% - 1 < LN% THEN GOSUB 12051 ELSE IF INS% THEN GOSUB 12051
12016 IF INS% THEN: F1% = FORE3: F2% = BACK3: GOSUB 3712: IF C% + L2% + 1 <= 80 THEN LOCATE
R$, C% + L2% + 1: IF LEN(R$) > LN% THEN L% = L% - 1: R$ = LEFT$(R$, L%): IF L2% > LN% THEN
GOTO 12021
12017 IF INS% THEN PRINT MID$(R$, L2% + 1, LEN(R$) - L2%): : GOSUB 12051: GOTO 12021 ELSE
12021
12018 IF L2% = 1 THEN SOUND 200, 5: GOTO 12006
12019 GOSUB 12048: L2% = L2% - 1: LOCATE R$, C% + L2%: F1% = FORE3: F2% = BACK3: GOSUB 3712:
PRINT MID$(R$, L2% + 1): LP$: L% = L% - 1: R$ = LEFT$(R$, L2% - 1) + MID$(R$, L2% + 1):
GOSUB 12051: GOTO 12006
12020 R$ = STRING$(LEN(R$), LP$): GOSUB 12048: LOCATE R$, C% + 1: F1% = FORE3: F2% = BACK3:
GOSUB 3712: PRINT R$: L% = 0: L2% = 1: R$ = "": GOSUB 12051: GOTO 12006
12021 IF L2% > LN% THEN 12022 ELSE 12006
12022 GOSUB 12048: F1% = fore1: F2% = back1: GOSUB 3712: IF LN% < LEN(R$) THEN IF C% +
LEN(R$) + 1 <= 80 THEN LOCATE R$, C% + LEN(R$) + 1: PRINT STRING$(LN% - LEN(R$), 32): : IF
C% + LN% < 80 THEN PRINT "": ***** Regreso.
12023 L22% = 0: C% = C% - LEN(R2$) + 1: R2$ = "": RP% = "": MINUS% = 0: EDITAR% = 0: PAT% =
0: PAT2% = 0: IF FLECHAS% = -1 THEN FLECHAS% = 0
12024 RETURN
12025 IF LEN(W$) < 2 AND W$ < CHR$(10) THEN 12006 ELSE R1% = ASC(RIGHT$(W$, 1)): IF
FLECHAS% AND (R1% = 72 OR R1% = 80 OR R1% = 117 OR R1% = 10) THEN FLECHAS% = 1: GOTO 12022
12026 IF R1% < 75 AND R1% < 77 AND R1% < 71 AND R1% < 79 AND R1% < 82 AND R1% < 83 AND
R1% < 116 AND R1% < 115 THEN 12006 ELSE L3% = L2%
12027 IF R1% = 82 THEN IF PAT2% THEN 12006 ELSE IF INS% = 1 THEN INS% = 0: GOSUB 12051: GOTO
12006 ELSE INS% = 1: GOTO 12006
12028 IF R1% = 75 THEN IF L2% < 1 THEN L3% = L2% - 1 ELSE IF EDITAR% THEN FLECHAS% = R1%:
GOTO 12022 ELSE SOUND 200, 5: GOTO 12006
12029 IF R1% = 77 THEN IF EDITAR% = -10 AND EDI% = 0 THEN FLECHAS% = R1%: GOTO 12022 ELSE IF
L2% > LEN(R$) OR L2% = LN% THEN IF EDITAR% THEN FLECHAS% = R1%: GOTO 12022 ELSE SOUND 200,
5: GOTO 12006
12030 IF R1% = 77 THEN L3% = L2% + 1
12031 IF R1% = 83 THEN GOTO 12045
12032 IF R1% = 71 THEN EDI% = -1: L3% = 1
12033 IF R1% = 79 THEN EDI% = -1: IF L% = LN% THEN L3% = L% ELSE L3% = L% + 1
12034 IF R1% = 9 THEN IF L2% + 5 > L% THEN IF L% = LN% THEN L3% = LN% ELSE L3% = L% + 1 ELSE
L3% = L2% + 5
12035 IF R1% < 116 THEN 12039 ELSE EDI% = -1
12036 IF R1% = 116 THEN L3% = INSTR(L2%, R$, " "): IF L3% = 0 THEN L3% = L% + 1: IF L3% >
LN% THEN L3% = LN% ELSE L3% = L3% ELSE L3% = L3% + 1
12037 IF R1% = 116 AND L3% < 1 AND MID$(R$, L3%, 1) = " " THEN L3% = L3% + 1: WHILE
MID$(R$, L3%, 1) = " " AND L3% <= L%: L3% = L3% + 1: WEND
12038 IF R1% = 116 AND L3% > LN% THEN L3% = LN%
12039 IF R1% < 115 OR L2% = 1 THEN 12043 ELSE L3% = L2% - 1: EDI% = -1
12040 IF LEFT$(R$, L2% - 1) = SPACE$(L2% - 1) THEN L3% = 1: GOTO 12043
12041 WHILE MID$(R$, L3%, 1) = " " AND L3% > 1: L3% = L3% - 1: WEND: IF L3% < 2 THEN L3% =
1: GOTO 12043
12042 WHILE MID$(R$, L3%, 1) < " " AND L3% > 1: L3% = L3% - 1: WEND: L3% = L3% + 1: IF L3%
<= 2 THEN L3% = 1
12043 'continúa
12044 GOSUB 12048: L2% = L3%: GOSUB 12051: GOTO 12006
12045 IF PAT2% THEN 12006 ELSE IF L2% = L% + 1 THEN SOUND 200, 5 ELSE L% = L% - 1
12046 F1% = FORE3: F2% = BACK3: GOSUB 3712: LOCATE R$, C% + L2%: PRINT MID$(R$, L2% + 1):
LP$:
12047 R$ = LEFT$(R$, L2% - 1) + MID$(R$, L2% + 1): GOSUB 12051: GOTO 12006
12048 Q$ = MID$(R$, L2%, 1): IF Q$ = "" THEN Q$ = LP$
12049 IF L2% <= LN% THEN LOCATE R$, C% + L2%: F1% = FORE3: F2% = BACK3: GOSUB 3712: PRINT
Q$: : Q$ = ""
12050 RETURN
12051 Q$ = MID$(R$, L2%, 1): IF Q$ = "" THEN Q$ = LP$
12052 IF C% + L2% <= 80 THEN LOCATE R$, C% + L2%: F1% = fore2: F2% = back2: GOSUB 3712:
PRINT Q$, BS$: : Q$ = "": RETURN ELSE RETURN
12053 GOTO 12028
12054 IF L1% THEN L1% = L1% - 1: R$ = LEFT$(R$, L1%): PRINT BS$: "": : LOCATE R$, C% + L1%,
1
12055 IF EC% AND L1% = 0 THEN 12057

```

```

12056 GOTO 12028
12057 C% = C% - LEN(RZ$); RZ$ = ""; RROC% = r$: r$ = ""; fl% = forel; f2% = backl; EC% = 0:
GOSUB 3712: r$ = RROC$(LN% - LI% + 1);
12058 RETURN
12060 IF Fl% = 0 THEN CW% = CC% ELSE CW% = 10 + COL%
12061 ER% = CW%: GOSUB 30000: ER% = 0: GOTO 12022
12300 'Rutina para preguntar si o no
12305 RW% = r$
12310 r% = 22: C% = 5: RZ$ = RW$ + " (S/N)? ": LN% = 1: GOSUB 12000
12315 IF r$ = "" THEN 12310
12320 IF FNC$(r$) <> "N" AND FNC$(r$) <> "S" THEN 12310
12340 r$ = FNC$(r$): RETURN
14000 'Rutina para procesar una lista b$(xx,yy)
14005 COLOR forel, backl: l% = AA10%: IF l% > LN% THEN l% = LN%
14010 FOR CC% = 1 TO l%: r% = NREN% + CC% - 1: GOSUB 14800: NEXT CC%
14030 CC% = 1: r% = NREN%: COL% = 1: IF AA10% = 0 THEN AA10% = 1
14040 GOSUB 14900: CAMBIO% = 0
14100 ' Punto de inicio *****
14110 GOSUB 14600 'Pide la información del renglón
14112 IF (Rl% = 117 OR Rl% = 119) AND FLECHAS% THEN IF B$(CC%, 2) = "" THEN Rl% = 25:
AA10% = AA10% + 1: GOTO 14116
14113 IF (Rl% = 117 OR Rl% = 119 OR Rl% = 9) AND FLECHAS% THEN GOSUB 14950: RETURN'^ End ^
Home
14114 IF (Rl% = 9 AND FLECHAS%) OR Rl% = 17 THEN RETURN' tab
14115 IF INS% AND FLECHAS% = 0 AND COL% = 1 AND AA10% < limite% THEN GOSUB 14960: INS% =
0: GOTO 14100' Inserta ren ***
14116 IF Rl% = 25 THEN FLECHAS% = -1
14117 IF FLECHAS% THEN Fl% = Rl%: GOTO 14200
14125 IF COL% = NUMCOL% AND CC% = AA10% AND CC% < limite% THEN AA10% = CC% + 1: FOR IC = 1
TO NUMCOL%: B$(AA10%, IC) = "": NEXT IC
14130 Fl% = 77
14200 IF Fl% = 80 THEN IF CC% = AA10% THEN BEEP ELSE CC% = CC% + 1: IF r% = NREN% + LN% -
1 THEN GOSUB 4900: GOSUB 14970 ELSE r% = r% + 1
14210 IF Fl% = 72 THEN IF CC% = 1 THEN BEEP ELSE CC% = CC% - 1: IF r% = NREN% THEN LNS% = -
1: GOSUB 4900: LNS% = 1: GOSUB 14970 ELSE r% = r% - 1
14220 IF Fl% = 77 THEN COL% = COL% + 1: IF COL% = NUMCOL% THEN COL% = 1: Fl% = 80: GOTO
14200
14230 IF Fl% = 75 THEN COL% = COL% - 1: IF COL% < 1 THEN COL% = NUMCOL%: Fl% = 72: GOTO
14200
14240 IF Fl% = 25 AND AA10% <> 1 THEN UL% = r% - 1: GOSUB 4900: GOSUB 14900: AA10% = AA10%
- 1: FOR II = CC% TO AA10%: RGH$(II) = RGH$(II + 1): FOR IC = 1 TO NUMCOL%: B$(II, IC) =
B$(II + 1, IC): NEXT IC, II: CAMBIO% = -1: IF CC% > AA10% THEN Fl% = 72: GOTO 14200
14595 GOTO 14100
14600 ' Pide la información: (renglón, columna)
14603 IF B$(CC%, COL%) = "" THEN B$(CC%, COL%) = TIPO$(COL%, 1, 1)
14605 C% = TIPO$(COL%, 1, FI%): LN% = TIPO$(COL%, 2, FI%): RZ$ = "": RP% = B$(CC%, COL%)
14610 LOCATE 24, 2: PRINT SPACE$(78): : LOCATE 24, 2: PRINT CA$(COL%, FI%):
14618 EDITAR% = -10: PAT% = 9: GOSUB 12000: IF RPP% THEN RPP% = 0: IF RPP% <> r$ THEN
CAMBIO% = -1
14620 IF Rl% = 9 OR Rl% = 17 THEN RETURN ELSE B$(CC%, COL%) = r$: Tl% = 2: Rl% = Rl%:
GOSUB 14700: Rl% = Rl%: IF ER% AND (Rl% = 25 OR (FLECHAS% AND (Rl% = 117 OR Rl% = 119 OR
Rl% = 9))) THEN 14630 ELSE IF ER% THEN 14610
14630 CCANT% = CC%: COLANT% = COL%: IF COL% = 12 THEN LOCATE r%, 80: PRINT CHR$(179):
14640 RETURN
14700 ON COL% GOTO 14710, 14720, 14730, 14730, 14740, 14730, 14730, 14730, 14730, 14750,
14730, 14760
14710 P% = VAL(B$(CC%, COL%)): IF P% < 1 OR P% > VAL(A$(1)) THEN B$(CC%, COL%) = "1"
14712 GOTO 14760
14720 P% = VAL(B$(CC%, COL%)): IF P% < 1 OR P% > 999 THEN B$(CC%, COL%) = "1"
14722 GOTO 14760
14730 IF B$(CC%, COL%) <> "" THEN B$(CC%, COL%) = FNV$(B$(CC%, COL%))
14732 IF B$(CC%, COL%) <> "y" AND B$(CC%, COL%) <> "n" THEN B$(CC%, COL%) = "n"
14734 GOTO 14760
14740 IF B$(CC%, COL%) = "" THEN B$(CC%, COL%) = "str"
14742 GOTO 14760
14750 P% = VAL("6H" + B$(CC%, COL%)): IF P% < 1 OR P% > 255 THEN B$(CC%, COL%) = "0"
14760 LOCATE r%, C%: PRINT B$(CC%, COL%): RETURN
14800 FOR COL% = 1 TO NUMCOL%
14810 C% = TIPO$(COL%, 1, FI%): LN% = TIPO$(COL%, 2, FI%): GOSUB 14700
14820 NEXT COL%
14830 RETURN

```

```

14900 ULY% = NREN% - 1: DRX% = NREN% + LIN% - 2: ULX% = NCOL% - 1: DRX% = NCOL% + 75: LNS% =
1: RETURN
14950 ULY% = 4: DRX% = 24: ULX% = 1: DRX% = 78: LNS% = 20: RETURN
14960 ULY% = r% - 1: LNS% = -1: GOSUB 14900: AAI0% = AAI0% + 1: FOR II = AAI0%
TO CC% STEP -1: RBG%(II) = RBG%(II - 1): FOR IC = 1 TO NUMCOL%: B$(II, IC) = B$(II - 1, IC):
NEXT IC, II: RBG%(CC%) = 0: FOR IC = 1 TO NUMCOL%: B$(CC%, IC) = "": NEXT IC: RETURN
14970 IF B$(CC%, 1) = " " THEN MAS% = 2 ELSE MAS% = 1
14973 COL% = COL%: FOR COL% = HAS% TO NUMCOL%: C% = TIPO$(COL%, 1, FI%): LN% = TIPO$(COL%,
2, FI%): GOSUB 14700: NEXT COL%: COL% = COL%
14975 RETURN
17000 'Rutina para contar el número de llaves
17010 NLLA% = 0: FOR ILLA = 1 TO AAI0%
17020 IF B$(ILLA, 1) = "n" THEN NLLA% = NLLA% + 1
17030 NEXT ILLA: A$(I) = FNMS(NLLA%)
17040 RETURN

29999 ' Rutina de Manejo de Ayuda en Línea y Mensajes de Error.
30000 SOUND 1500, 1: RWQK% = r%: FIR% = f1%: F2R% = f2%: lnet = 0: f1% = fore2: f2% = back2:
SISTEMA% = 0: rene% = 11: cole% = 0: net = 1: CCCC% = POS(0): RRRR% = CSRLIN: r% = " "
30001 ON ER% GOTO 30010, 30020, 30030, 30040, 30050, 30060, 30070, 30080, 30090, 30100,
30110, 30120, 30130, 30140, 30150, 30160, 30170, 30180, 30190, 30200, 30210, 30220, 30230,
30240
30004 SISTEMA% = 0: GOTO 31000
30005 SISTEMA% = -1: GOTO 31000
30006 ** REN% = El renglon inicial
30007 ** cole% = columna del texto si es = 0 lo centra
30008 ** net = es el número de líneas de texto del error
30009 ** SISTEMA% indica si despues del error se sale
30010 net = 4: RE$(0) = "El largo del registro define la longitud de"
30011 RE$(1) = "los registros lógicos en bytes, este valor "
30012 RE$(2) = "corresponde al largo en bytes del registro "
30013 RE$(3) = "incluyendo las llaves. ": GOTO 30004
30020 net = 4: RE$(0) = "Longitud variable del registro. Especifica "
30021 RE$(1) = "si el archivo va a contar con registro de "
30022 RE$(2) = "longitud variable. Oprima 'y' si desea"
30023 RE$(3) = "que el archivo contenga registros variables": GOTO 30004
30030 net = 4: RE$(0) = "Esta campo es opcional y le especifica a "
30031 RE$(1) = "Btrieve que debe eliminar los espacios en blanco "
30032 RE$(2) = "del registro, como método de compresión. "
30033 RE$(3) = "Solo es válida si especificó 'y' en 'long. var.': GOTO 30004
30040 net = 3: RE$(0) = "La compresión de datos. Especifica si Btrieve"
30041 RE$(1) = "ejecutará la compresión de los registros "
30042 RE$(2) = "automáticamente. ": GOTO 30004
30050 net = 4: RE$(0) = "Cantidad de líneas. Especifica el Número de "
30051 RE$(1) = "renglones para definir las llaves en el archivo."
30052 RE$(2) = "Si se especifica un valor de cero, Btrieve creara"
30053 RE$(3) = "un archivo de solo datos. ": GOTO 30004
30060 net = 4: RE$(0) = "El tamaño de la página. Especifica el tama- "
30061 RE$(1) = "ño físico de la pagina en bytes para contener"
30062 RE$(2) = "registros. Se pueden especificar multiples de"
30063 RE$(3) = "512, menores o iguales a 4096 ": GOTO 30004
30070 net = 8: RE$(0) = "La pre-colocación de página especifica el "
30071 RE$(1) = "número de páginas que se le pre-colocarán "
30072 RE$(2) = "al archivo, si no se necesita pre-colocar "
30073 RE$(3) = "ninguna página, entre un valor de cero. "
30074 RE$(4) = "Esta acción aparta sectores contiguos "
30075 RE$(5) = "en el disco para mayor rapidez de acceso y"
30076 RE$(6) = "requiere que haya espacio suficiente en "
30077 RE$(7) = "el disco.": GOTO 30004
30080 net = 4: RE$(0) = "Si al crear un archivo existe otro con el "
30081 RE$(1) = "mismo nombre, con esta opción se reemplaza "
30082 RE$(2) = "el archivo anterior con el nuevo. Este dato"
30083 RE$(3) = "es opcional. ": GOTO 30004
30090 net = 5: RE$(0) = "Es opcional y especifica el tipo de archivo"
30091 RE$(1) = "Si desea un 'solo llave' responda 'n' "
30092 RE$(2) = "Si desea un 'estandar' especifique 'y' "
30093 RE$(3) = "Si desea un 'solo datos' especifique 'y' "
30094 RE$(4) = "y coloque cero en la 'Cantidad de líneas': GOTO 30004
30100 net = 4: RE$(0) = "El elemento de espacio libre especifica la "
30101 RE$(1) = "cantidad de espacio que Btrieve reservara "
30102 RE$(2) = "en las páginas de datos para preveer la "
30103 RE$(3) = "expansión de registros. ": GOTO 30004

```

```

30110 net = 4: RE$(0) = "Indica la posición inicial de la "
30111 RE$(1) = "llave dentro del registro, esta debe ser "
30112 RE$(2) = "mayor a uno y no debe rebasar el largo de "
30113 RE$(3) = "el registro. ": GOTO 30004
30120 net = 4: RE$(0) = "El largo de la llave define la longitud de "
30121 RE$(1) = "la llave en el registro. El total del "
30122 RE$(2) = "largo de las llaves no debe exceder la "
30123 RE$(3) = "longitud del registro. ": GOTO 30004
30130 net = 3: RE$(0) = "El valor duplicado de la llave indica si en"
30131 RE$(1) = "el archivo existirán registros con el mismo"
30132 RE$(2) = "valor de llave.": GOTO 30004
30140 net = 4: RE$(0) = "El elemento de llaves modificables, especi-"
30141 RE$(1) = "fica si el valor de las llaves pueden ser "
30142 RE$(2) = "modificadas durante una operación de "
30143 RE$(3) = "actualización. (Update) ": GOTO 30004
30150 net = 8: RE$(0) = "Tipo de llave, especifica el tipo de dato "
30151 RE$(1) = "para la llave, se digitan las tres pri-"
30152 RE$(2) = "meras letras del tipo que se desee dar"
30153 RE$(3) = "string,integer,float,date,time,decimal,money"
30154 RE$(4) = "logical,numeric,bfloat,lstring,zstring "
30155 RE$(5) = "unsigned binary, autoincrement"
30156 RE$(6) = "lstring=primer bite contiene largo de la cadena"
30157 RE$(7) = "zstring=cadena delimitada por char(0) (leng.C)": GOTO 30004
30160 net = 4: RE$(0) = "La ordenación desendente especifica si se "
30161 RE$(1) = "desea que los datos se ordenen descendente-"
30162 RE$(2) = "mente, si se omite o si se responde N "
30163 RE$(3) = "los datos se ordenaran ascendente": GOTO 30004
30170 net = 4: RE$(0) = "Esta opción indica si se quiere cambiar el "
30171 RE$(1) = "orden de los elementos en ASCII, se emplea "
30172 RE$(2) = "cuando no se utiliza el alfabeto en ingles "
30173 RE$(3) = "o para alterar la prioridad de las letras": GOTO 30004
30180 net = 6: RE$(0) = "Llave manual especifica si las llaves se "
30181 RE$(1) = "definanr manualmente , si se definen llaves"
30182 RE$(2) = "manuales, se puede indicar con una bandera si"
30183 RE$(3) = "la llave se escribe o no en el indice. "
30184 RE$(4) = "(la bandera debe ser un segmento de llave,"
30185 RE$(5) = "si la bandera tiene un nulo no se escribe)": GOTO 30004
30190 net = 6: RE$(0) = "Llave nula especifica si la llave que se "
30191 RE$(1) = "define se escribirá o no en el registro,"
30192 RE$(2) = "dependiendo de su valor. Si el valor de cada "
30193 RE$(3) = "bit de la llave es un caracter nulo, la llave no"
30194 RE$(4) = "se escribira. Si se definen llaves nulas, se "
30195 RE$(5) = "debe especificar un valor para la misma": GOTO 30004
30200 net = 3: RE$(0) = "El valor nulo de llave especifica el carac-"
30201 RE$(1) = "ter que se toma como valor nulo de llave "
30202 RE$(2) = "Ejemplo: 20 hex = blanco": GOTO 30004
30210 net = 3: RE$(0) = "Llave segmentada especifica si la llave se "
30211 RE$(1) = "dividirá en segmentos o será una llave "
30212 RE$(2) = "no-segmentada ": GOTO 30004
30220 net = 4: RE$(0) = "En este elemento se especifica el nombre "
30221 RE$(1) = "del archivo el cual contendrá la secuencia "
30222 RE$(2) = "de caracteres que utilizará para la ordenación."
30223 RE$(3) = "Se pueden incluir directorios ": GOTO 30004
30230 r$ = "LA 'Cantidad de líneas' ES IGUAL CON CERO": GOTO 30004
30240 r$ = "NOMBRE DE ARCHIVO INVALIDO, REPITALO P.F.": GOTO 30004
31000 -----
31001 IF r$ <> "" THEN RE$(net - 1) = r$
31002 IF SISTEMA% = 0 THEN net = net + 1: RE$(net - 1) = "-" Presione [Return] o [Esc] "-"
31003 FOR XER% = 0 TO net + SISTEMA%: IF LEN(RE$(XER%)) > len THEN len = LEN(RE$(XER%))
31005 NEXT XER%: IF colet = 0 THEN colet = 40 - ((len + 2) / 2)
31006 FIRSTLINE% = renet: LASTLINE% = renet + net + 2: IF SISTEMA% THEN 31009
31008 CALL QBSAVE(renet, colet, net + 3, len + 6, ve$)
31009 COLR% = (f1% AND 15) + (f2% * 16) + 8 * (f1% AND 16): RY3$ = "": IF fore% = 0 THEN
FORE4% = 112 ELSE FORE4% = 7
31010 FOR XER% = colet + len + 5 TO colet + 2 STEP -1: RY3$ = CHR$(SCREEN(renet + net + 2,
XER%)) + RY3$: NEXT XER%
31015 FOR XER% = 0 TO net + 1: LOCATE renet + XER%, colet
31020 IF XER% = 0 THEN r$ = CHR$(201) + STRING$(len + 2, 205) + CHR$(187): GOTO 31050
31030 IF XER% = net + 1 THEN f1% = fore2: f2% = back2: r$ = CHR$(200) + STRING$(len + 2,
205) + CHR$(188): GOTO 31050
31040 f1% = fore2: f2% = back2: r$ = CHR$(186) + " " + RE$(XER% - 1) + SPACE$(len -
LEN(RE$(XER% - 1))) + " " + CHR$(186)

```

```

31050 COLOR f1%, f2%: PRINT r%: : IF XER% < 0 THEN COLOR 7, 0: RY2% = CHR$(SCREEN(renet +
XER%, colat + lnet + 5)): RY1% = CHR$(SCREEN(renet + XER%, colat + lnet + 4)): PRINT RY1% +
RY2%:
31060 NEXT XER%
31062 SOUND 100, 3
31063 COLOR 7, 0: LOCATE renet + net + 2, colat + 2: PRINT RY3%:
31065 IF SISTEMAT THEN 31120
31080 INK$ = "": WHILE INK$ < CHR$(13) AND INK$ < CHR$(27): INK$ = INKEY$: WEND: IF
SISTEMAT THEN 3500
31100 CALL QBrest(renet, colat, net + 3, lnet + 6, ve%)
31120 LOCATE RRRR%, CCCC%: f1% = FIR%: f2% = F2R%: r% = RWQK%: IF SISTEMAT = -1 THEN 3500
ELSE RETURN
SUB QBrest (r%, C%, LR%, LC%, r%) STATIC
O% = 1
FOR J% = 0 TO LR% - 1
FOR I% = 0 TO LC% - 1
CO% = ASC(MID$(r%, O% + 1, 1))
COLOR CO% - CINT(CO% / 15 - 1) * 16, CINT(CO% / 15 - 1)
LOCATE r% + J%, C% + I%: PRINT MID$(r%, O%, 1): :
O% = O% + 2
NEXT I%, J%
END SUB

SUB QBSAVE (r%, C%, LR%, LC%, r%) STATIC
r% = ""
FOR J% = 0 TO LR% - 1
FOR I% = 0 TO LC% - 1
r% = r% + CHR$(SCREEN(r% + J%, C% + I%)) + CHR$(SCREEN(re + J%, C% + I%, 1))
NEXT I%, J%
END SUB

```

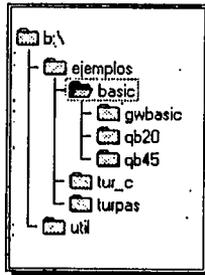
APÉNDICE F

CONTENIDO Y DISTRIBUCIÓN DEL DISCO ANEXO.

Al final de cada original de este trabajo se colocó un disco de 3 1/2" HD, que contiene todo el software desarrollado en este trabajo.

Hasta el momento se desconoce el éxito que pueda tener la intencion de anexas el disco, es decir si la biblioteca y coordinación de la carrera lo permitan y en su caso se hagan responsables de su cuidado, de no tener éxito la idea, el material original se encuentra impreso a lo largo del trabajo.

El siguiente diagrama muestra los directorios que contiene el disco anexo:



A continuación se presenta el contenido y explicación de cada directorio:

En el directorio raíz se encuentra un archivo llamado LEEME.DOC que contiene notas finales sobre el contenido del disco.

El directorio EJEMPLOS contiene los programas muestra que se discutieron en el capítulo IV, así como también los archivos de datos ya generados para cada caso. A continuación se muestran los archivos que contiene cada subdirectorio:

Subdirectorío BASIC

ALUMNOS.BTR (Es el archivo de datos Btrieve que funciona para todos los ejemplos de Basic)
ALUMNOS.DES

Subdirectorío GWBASIC

BTRV.BAS (Programa muestra que corre en GWBASIC)

Subdirectorío QB20

BTRVQB20.BAS (Programa muestra que corre con Quick Basic V2.0)

BTRVQB20.EXE (Programa ejecutable)

Subdirectorío QB45

BTRV45.BAS (Programa muestra que corre con Quick Basic V4.5)

BTRV45.EXE (Programa ejecutable)

Subdirectorío TURPAS

ALUM_PAS.DES (Archivo de descripción del archivo de datos)

ALUM_PAS.BTR (Archivo de datos Btrieve)

BTRV.PAS (Programa muestra que corre con Turbo Pascal V5.0)

BTRV.EXE (Programa ejecutable)

Subdirectorío TUR_C

ALUM_C.BTR (Archivo de datos Btrieve)

ALUM_C.DES (Archivo de descripción del archivo de datos)

BTRV.C (Programa muestra que corre con Turbo C V2.0)

BTRV.EXE (Programa ejecutable)

Subdirectorío UTIL

AYUDA2.BAS (Programa de utilería descrito en el apéndice E, que corre con QBASIC.EXE del sistema operativo)

AYUDAB.BAS (Programa de utilería descrito en el apéndice E, que corre con Quick Basic V4.5. correr QB AYUDAB /L UTIL.QLB)

AYUDAB.EXE (Programa ejecutable)

MOUSE.BAS (Programa ejemplo para manejar el mouse en Quick Basic V45.)

UTIL.QLB (Funciones adicionales para el programa ayudab.exe)

UTIL.LIB (Archivo para compilar las funciones adicionales.)

GLOSARIO

Ambiente

Se refiere a las condiciones de trabajo dentro de una computadora, normalmente el tipo de sistema operativo.

Antecedentes o fundamentos

Son definiciones y descripciones de las cualidades y términos que son necesarios para comprender y aprovechar las siguientes secciones del trabajo.

Aplicación

Se refiere a un programa creado en cualquiera de los diferentes lenguajes que maneja Btrieve. Por ejemplo; un programa en BASIC o TURBO C. Se da por hecho que dicho programa trabaja con una base de datos manipulada o manejada por Btrieve, es decir que el programa utiliza las operaciones de Btrieve para acceder la información, y de manera inherente este programa sirve para resolver algún problema de informática.

Archivo Btrieve

Se refiere a un archivo especial, porque un archivo Btrieve, tiene grabado el inicio, las características generales del archivo y las características de los índices. Btrieve sólo puede manejar archivos de este tipo. Éstos son definidos y creados por el programador, por medio de una utilería que provee el paquete de Btrieve o por medio de una pequeña aplicación desarrollada para ese efecto.

Bloqueo o cerradura

Un bloqueo o cerradura es la forma de controlar el acceso a uno o más registros dentro de un archivo. Esto es útil cuando varias computadoras accesan un archivo al mismo tiempo. (Ver cap. V para mayor información sobre su uso)

Brequest

Es un programa que se utiliza únicamente en la modalidad Btrieve de *basado en servidor* que permite comunicar a una aplicación con el servidor de archivos de la red.

Bserver

Es el programa que se utiliza únicamente en la modalidad de Btrieve de *basado en servidor* en redes Novell 2.11 hasta antes de la 3.0 y es el programa que contiene las rutinas de Btrieve, es equivalente al programa BTRIEVE.NLM, que se utiliza en las redes Novell de la versión 3.0 en adelante.

Btrieve

Se refiere al manejador de registros indexados, que se presenta como un programa ejecutable, que al ponerlo en marcha deja las rutinas en la memoria de la computadora, para que cualquier aplicación pueda hacer uso de ellas.

Buffer

Ver memoria intermedia.

Consistencia

Es una característica de las bases de datos que se refiere a que todas las referencias de un archivo hacia otro existan.

DOS

Se refiere al sistema operativo MS-DOS, (MicroSoft Disk Operating System) que es el más utilizado en micro-computadoras.

E/S

Son las siglas de Entrada y Salida, es decir, procesos de grabar y leer información a un disco, a la memoria, o a un archivo específico.

Estación de trabajo

Una red de micro-computadoras está compuesta de manera básica, de un servidor y de una o más computadoras que están conectadas al servidor y utilizan de forma comunitaria la información que se encuentra dentro del servidor. A las computadoras conectadas al servidor se les denomina estaciones de trabajo.

Implementar

Se refiere al método o forma específica en que se resuelve un problema. Y un problema tiene diferentes formas de ser resuelto, es decir, que la solución se puede implementar de distintas maneras.

Índice

Es una lista ordenada de valores de llaves y cada valor tiene consigo la dirección donde se puede localizar el registro completo con el valor de llave correspondiente. Por ejemplo; dentro de un archivo de alumnos, se pueden tener índices ordenados por nombre, número de cuenta, grupo, semestre, etc.

Integridad

Se refiere a una cualidad de bases de datos que consiste en que los campos de referencia a otros archivos que deban tener un valor, lo tengan. O si un campo debe tener la información en cierto formato, así se encuentre.

Interfaz

Se refiere a un programa que permite comunicar a una versión y marca de un lenguaje específico con Btrieve. En algunos casos en un archivo con el código necesario y en otros es un archivo objeto que se liga con la aplicación al compilar.

Liberar o desbloquear

Es el proceso que permite a otra estación de trabajo realizar un bloqueo sobre el registro liberado. Esto es útil cuando varias computadoras accesan un archivo al mismo tiempo. (Ver Cap. V para mayor información sobre su uso)

Llave

Se refiere a un campo dentro de un registro que es utilizado para crear un índice.

Manejador de registros

Se refiere al programa que facilita el mantenimiento a un archivo desde un programa realizado en algún lenguaje.

Memoria intermedia o buffer

Es un espacio predestinado o apartado para poder contar con él para procesos específicos.

Modelo relacional de bases de datos

Se refiere a un conjunto de archivos que se relacionan entre sí. Y su característica principal es que se relacionan por medio de una llave común a uno o más de los archivos.

Multi-acceso

El Multi-acceso se presenta cuando una o más computadoras accesan la misma información, normalmente parte de un archivo. Esto se presenta en redes de computadoras.

Nlm

Son las siglas de NetWare Loadable Module. Se refiere a programas que son ejecutados desde la consola del servidor de una red. Normalmente estos programas están relacionados con la operación e instalación de una red. Estos tipos de programas se encuentran a partir de la versión 3.0 de Novell NetWare.

Novell NetWare

Es el nombre del sistema operativo creado por Novell, Inc, que permite controlar una red.

Novell NetWork

Son los productos que intervienen en la instalación de una red. También se refiere a los componentes que conforman una red en conjunto. Es el área de trabajo de una red.

Operaciones

Una operación es a una función o proceso específico que realiza Btrieve. Por ejemplo una consulta, una alta, una actualización. Btrieve cuenta con 40 operaciones diferentes, éstas pueden ser ejecutadas por medio de una instrucción del lenguaje que se utiliza. El capítulo III describe las operaciones de Btrieve.

Página

Se refiere a un espacio de memoria o disco, con un tamaño determinado. Se utilizan para separar lógicamente un archivo, un índice o memoria.

Paquete

Un paquete es conocido como el conjunto de manuales y software que se incluye con la compra de un programa específico. En este trabajo se habla normalmente del paquete de Btrieve.

Red

Se refiere al conjunto de computadoras (pensantes) conectadas de alguna manera, con una o más computadoras que son los servidores. Los servidores controlan el Multi-acceso a diferentes periféricos y dispositivos, como: discos duros, impresoras, unidades de respaldo, modems, etc. Las computadoras conectadas a los servidores se conocen como estaciones de trabajo.

Registro multi-indexado

Un registro multi-indexado se refiere a que uno o más de los campos que se encuentran en el registro, están ordenados dentro de un índice. Esto permite revisar o consultar la información que se encuentra dentro del archivo de diferentes maneras, o buscar un registro específico por medio del valor de un campo llave.

Segmento y offset

El espacio de trabajo en memoria o disco, se divide lógicamente en segmentos y éstos a su vez se dividen en offset. Esto se ha utilizado para definir unidades de espacio que tienen una dirección específica, es decir que una área de memoria se puede localizar por medio de su número de segmento y su número de offset.

Servidor

Se refiere a la computadora que controla y administra los recursos de una red.

Vap

Son las siglas de Value Added Process. Se refiere a programas que son ejecutados cuando se pone en marcha el servidor de la red además del sistema operativo. Normalmente estos programas están relacionados con la operación de una red. Estos tipos de programas se encuentran a partir de la versión 2.1 hasta antes de la 3.0 de Novell NetWare.