

**REPUBLICA BOLIVARIANA DE VENEZUELA
MINISTERIO DE EDUCACION
UNIVERSIDAD EXPERIMENTAL SIMON RODRIGUEZ
CONVENIO FIEC.
BARQUISIMETO ESTADO LARA.**

DATOS Y ESTRUCTURA DE DATOS



Integrantes:

Reyes S. Heydimar E.	CI 14.001.472
Granadillo Luz A.	CI. 12.594.739
Macgyory S. Delgado	C.I 14.696.614
Stanlyn Sanchez	C.I 12.244.983

Facilitador:

Manuel Mújica

Barquisimeto, Junio 2007

DATOS

Datos son los hechos que describen sucesos y entidades. "Datos" es una palabra en plural que se refiere a más de un hecho. A un hecho simple se le denomina "data-ítem" o elemento de dato. Los datos son comunicados por varios tipos de símbolos tales como las letras del alfabeto, números, movimiento de labios, puntos y rayas, señales con la mano, dibujos, etc. Estos símbolos se pueden ordenar y reordenar de forma utilizable y se les denomina información. Los datos son símbolos que describen condiciones, hechos, situaciones o valores. Los datos se caracterizan por no contener ninguna información. Un dato puede significar un número, una letra, un signo ortográfico o cualquier símbolo que represente una cantidad, una medida, una palabra o una descripción. La importancia de los datos está en su capacidad de asociarse dentro de un contexto para convertirse en información. Por sí mismos los datos no tienen capacidad de comunicar un significado y por tanto no pueden afectar el comportamiento de quien los recibe. Para ser útiles, los datos deben convertirse en información para ofrecer un significado, conocimiento, ideas o conclusiones.

Los Datos a diferencia de la información son utilizados como diversos métodos para comprimir la información a fin de permitir una transmisión o almacenamiento más eficaces.

Aunque para el procesador de la computadora hace una distinción vital entre la información entre los programas y los datos, la memoria y muchas otras partes de la computadora no lo hace. Ambos son registradas temporalmente según la instrucción que se le de. Es como un pedazo de papel no sabe ni le importa lo que se le escriba: un poema de amor, las cuentas del banco o instrucciones para un amigo. Es lo mismo que la memoria de la computadora. Sólo el procesador reconoce la diferencia entre datos e información de cualquier programa. Para la memoria de la computadora, y también para los dispositivos de entrada y salida (E/S) y almacenamiento en disco, un programa es solamente más datos, más información que debe ser almacenada, movida o manipulada.

La cantidad de información de un mensaje puede ser entendida como el número de símbolos posibles que representan el mensaje. "los símbolos que representan el mensaje no son más que datos significativos.

En su concepto más elemental, la información es un mensaje con un contenido determinado emitido por una persona hacia otra y, como tal, representa un papel primordial en el proceso de la comunicación, a la vez que posee una evidente función social. A diferencia de los datos, la información tiene significado para quien la

recibe, por eso, los seres humanos siempre han tenido la necesidad de cambiar entre sí información que luego transforman en acciones. "La información es, entonces, conocimientos basados en los datos a los cuales, mediante un procesamiento, se les ha dado significado, propósito y utilidad"

TIPO DE DATOS

El tipo de un dato es el conjunto de valores que puede tomar durante el programa. Si se le intenta dar un valor fuera del conjunto se producirá un error.

La asignación de tipos a los datos tiene dos objetivos principales:

- Por un lado, detectar errores en las operaciones
- Por el otro, determinar cómo ejecutar estas operaciones

De Pascal se dice que es un lenguaje fuertemente tipeado. Esto quiere decir que todos los datos deben de tener un tipo declarado explícitamente, y además que existen ciertas restricciones en las expresiones en cuanto a los tipos de datos que en ellas intervienen.

Una ventaja de los lenguajes fuertemente tipeados es que se gasta mucho menos esfuerzo en depurar (corregir) los programas gracias a la gran cantidad de errores que detecta el compilador.

Los tipos de datos, como casi todos los objetos de Pascal, se pueden declarar. La declaración de tipos ya se comentó en el tema correspondiente a la estructura de un programa.

ENTRE ELLOS PODEMOS NOMBRAR

TIPOS ESTÁTICOS

Casi todos los tipos de datos son estáticos, la excepción son los punteros y no se tratarán debido a su complejidad.

Que un tipo de datos sea estático quiere decir que el tamaño que ocupa en memoria no puede variar durante la ejecución del programa. Es decir, una vez declarada una variable de un tipo determinado, a ésta se le asigna un trozo de memoria fijo, y este trozo no se podrá aumentar ni disminuir.

TIPOS DINÁMICOS

Dentro de esta categoría entra sólomente el tipo puntero. Este tipo te permite tener un mayor control sobre la gestión de memoria en tus programas. Con ellos puedes manejar el tamaño de tus

variables en tiempo de ejecución, o sea, cuando el programa se está ejecutando.

Los punteros quizás sean el concepto más complejo a la hora de aprender un lenguaje de programación, sobre todo si es el primero que aprendes. Debido a esto, no lo trataremos. Además, lenguajes que están muy de moda (por ejemplo Java) no permiten al programador trabajar con punteros.

TIPOS SIMPLES

Como su nombre indica son los tipos básicos en Pascal. Son los más sencillos y los más fáciles de aprender. Por todo esto, serán en los que nos centremos.

Los tipos simples más básicos son: entero, lógico, carácter y real. Y la mayoría de los lenguajes de programación los soportan, no como ocurre con los estructurados que pueden variar de un lenguaje a otro.

TIPOS ESTRUCTURADOS

Mientras que una variable de un tipo simple sólo referencia a un elemento, los estructurados se refieren a colecciones de elementos.

Las colecciones de elementos que aparecen al hablar de tipos estructurados son muy variadas: tenemos colecciones ordenadas que se representan mediante el tipo array, colecciones sin orden mediante el tipo conjunto, e incluso colecciones que contienen otros tipos, son los llamados registros.

TIPOS ORDINALES

Dentro de los tipos simples, los ordinales son los más abundantes. De un tipo se dice que es ordinal porque el conjunto de valores que representa se puede contar, es decir, podemos establecer una relación uno a uno entre sus elementos y el conjunto de los números naturales.

Dentro de los tipos simples ordinales, los más importantes son:

1. El tipo entero (integer)
2. El tipo lógico (boolean)
3. El tipo carácter (char)

TIPOS NO-ORDINALES

Simplificando, podríamos reducir los tipos simples no-ordinales al tipo real. Este tipo nos sirve para declarar variables que pueden tomar valores dentro del conjunto de los números reales.

A diferencia de los tipos ordinales, los no-ordinales no se pueden contar. No se puede establecer una relación uno a uno entre ellos y los números naturales. Dicho de otra forma, para que un conjunto se considere ordinal se tiene que poder calcular la posición, el anterior elemento y el siguiente de un elemento cualquiera del conjunto. ¿Cuál es el sucesor de 5.12? Será 5.13, o 5.120, o 5.121, ...

PROPIEDADES DE LOS DATOS

Las propiedades son miembros que ofrecen un mecanismo flexible para leer, escribir o calcular los valores de campos privados. Se pueden utilizar las propiedades como si fuesen miembros de datos públicos, aunque en realidad son métodos especiales denominados descriptores de acceso. De este modo, se puede tener acceso a los datos con facilidad, a la vez que proporciona la seguridad y flexibilidad de los métodos.

La ventana Propiedades de base de datos es una ventana anclada que muestra las categorías utilizadas para asignar propiedades a los elementos de un diagrama de modelo de base de datos.

Las categorías que muestra la ventana Propiedades de base de datos dependen del elemento que seleccione en el diagrama de modelo de base de datos.

Por ejemplo, si selecciona una tabla del diagrama, la ventana Propiedades de base de datos muestra las categorías que se utilizan para modificar las propiedades de la tabla. Si selecciona una relación del diagrama, la ventana muestra las categorías que se utilizan para modificar las propiedades de la relación.

Para obtener detalles acerca de una categoría, haga clic en ella y presione F1.

La ventana Propiedades de base de datos se puede cerrar, abrir o abrir y mantener oculta. El estado de la ventana Propiedades de base de datos determina la manera de comenzar a modificar las propiedades de cualquier elemento del diagrama de modelo de base de datos.

- ▶ Si la ventana Propiedades de base de datos está cerrada
 1. Para abrir la ventana Propiedades de base de datos, haga doble clic en el elemento cuyas propiedades desea modificar.
 2. En la ventana Propiedades de base de datos, haga clic en la categoría que contenga la información que desea modificar y realice los cambios deseados.
- ▶ Si la ventana Propiedades de base de datos está abierta

1. Haga clic en el elemento del modelo de base de datos cuyas propiedades desea modificar.

La ventana Propiedades de base de datos muestra las categorías asociadas al elemento seleccionado.

2. En la ventana Propiedades de base de datos, haga clic en la categoría que contenga la información que desea modificar y realice los cambios deseados.

El diagrama de modelo de base de datos se actualiza automáticamente.

- ▶ Si la ventana Propiedades de base de datos está abierta pero oculta
 1. Haga clic en el elemento del modelo de base de datos cuyas propiedades desea modificar.
 2. Mueva el puntero sobre la barra de título Propiedades de base de datos para que aparezca la ventana.
 3. En la ventana Propiedades de base de datos, haga clic en la categoría que contenga la información que desea modificar y realice los cambios deseados.

ATRIBUTOS

Los atributos constituyen un medio apropiado para asociar información declarativa con código (tipos, métodos, propiedades, etc.). Una vez asociado a una entidad del programa, el atributo se puede consultar en tiempo de ejecución mediante una técnica denominada reflexión.

Existen dos formas de atributos: atributos que se definen en la biblioteca de clases base de Common Language Runtime y atributos personalizados que se pueden crear para agregar información adicional al código. Esta información se puede recuperar después mediante programación.

ARCHIVOS

Es un conjunto de datos que se almacenan bajo un determinado formato; puede ser guardado en el disco duro de la computadora o en algún otro medio de almacenamiento como disquete, disco compacto, unidad ZIP, etc

Es un programa, un conjunto de datos utilizados por el programa o un documento creado por los usuarios.

Los archivos son las unidades básicas de almacenamiento que permiten a la computadora distinguir entre los diversos conjuntos de información. Aunque no siempre es el caso, un archivo se suele

encontrar en un formato legible por los usuarios. Aun así, en un archivo se agrupan instrucciones, números, palabras o imágenes en unidades coherentes que el usuario puede recuperar, modificar, eliminar, guardar o enviar a un dispositivo de salida.

TIPOS DE ARCHIVOS:

ARCHIVOS SECUENCIALES:

Se refiere al procesamiento de los registros, no importa el orden en que se haga, para eso los registros están organizados en forma de una lista y recuperarlos y procesarlos uno por uno de principio a fin.

Dependiendo del dispositivo de almacenamiento utilizado el archivo se puede mostrar el usuario como si fuera un sistema secuencial.

Al finalizar un archivo secuencial se denota con una marca de fin de archivo. (End end-of-file)

El usuario de un archivo secuencial puede ver los registros en un orden secuencial simple.

La única forma de recuperar registros es comenzar al principio y extraerlos en el orden contemplado.

La manipulación de los archivos se hace en el contexto de la programación en un lenguaje por procedimientos de alto nivel. Estos lenguajes tienden a expresar la manipulación de archivos mediante subrutinas que se definen como parte del lenguaje formal o se incluyen como extensiones del lenguaje en una biblioteca estándar.

La mayor parte de los lenguajes por procedimiento de alto nivel cuenta con características que ayudan a detectar la marca de fin de archivo.

ARCHIVOS DE TEXTO

También conocidos como (Stream File) son utilizados para almacenar documentos que consisten en texto; En ellos, cada registro es un solo símbolo o código de control.

El leer estos archivos recibimos la información en orden secuencial en el que aparece cuando lo vemos en un monitor.

Los archivos de texto son una secuencia de líneas separadas por marcas de fin de línea.

El usuario escribe los archivos de textos mediante un procesador de palabras que le permitirá almacenar la información pero no estrictamente en forma secuencial.

El procesador también nos permite desplazarnos por todo el bloque de información y permitirnos realizar modificaciones.

Mientras el usuario avance rápidamente en la lectura de registro lograra ver mas archivos.

Casi todos los entornos de programación por procedimientos de alto nivel cuentan con subrutinas para manipular los archivos de texto.

Estas subrutinas pueden formar parte de la definición formal del lenguaje o que se ofrezca en biblioteca como extensiones del mismo

ARCHIVOS INDIZADOS

Es la aplicación de incluir índices en el almacenamiento de los archivos; de esta forma nos será más fácil buscar algún registro sin necesidad de ver todo el archivo.

Un índice en un archivo consiste en un listado de los valores del campo clave que ocurren en el archivo, junto con la posición de registro correspondiente en el almacenamiento masivo.

a.- La colocación de un listado al inicio del archivo: para la identificación del contenido.

b.- La presentación de un segundo índice: para reflejar la información de cada punto principal del índice anterior.

c.- La actualización de los índices: Cuando se insertan y eliminan archivos, es preciso actualizar los índices para evitar contratiempos actualizando un archivo.

d.- La organización de un índice: Nos evita examinar archivo por archivo para recuperar algún registro buscado; por lo tanto ahorraríamos tiempo si tenemos una adecuado organización de los índices.

Algunos lenguajes de alto nivel cuentan con subtítulos para manipular los archivos de un registro indizado.

Valiéndose de las subrutinas es posible escribir programas sin tener que preocuparse por la estructura real del sistema de índices que se aplique.

ARCHIVOS DISPERSOS

También llamados (Hashed Files) representan un sistema de almacenamiento de archivos que solo ofrece acceso directo, y permiten calcular la posición de un registro en el almacenamiento masivo.

El usuario debe dividir el área de almacenamiento asignando al archivo en varias secciones llamadas cubetas para poder ingresar los datos.

La distribución de la información en las cubetas es problemática debido a que la estructura de los archivos es dispersa.

Dentro de los archivos se presentan colisiones de información debido al agrupamiento de los registros ingresados.

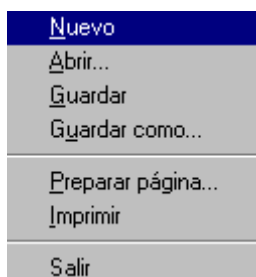
Casi ninguno de los lenguajes de programación por procedimientos en la actualidad ofrece implantaciones directas de archivos dispersos; esto es debido a las cuestiones dependientes de la aplicación implicadas en el diseño de estos archivos.

CARACTERÍSTICAS DE LOS ARCHIVOS

Las principales características de esta estructura son:

- Independencia de las informaciones respecto de los programas
- La información almacenada es permanente
- Un archivo puede ser accedido por distintos programas en distintos momentos
- Gran capacidad de almacenamiento.
- El nombre identifica a cada archivo. "CALC"
- La extensión son los caracteres que van al final del nombre separados de éste por un punto. Permite conocer que tipo de información almacena un archivo y suele ser de tres caracteres. "EXE"
- En una misma carpeta no puede haber dos archivos con el mismo nombre y extensión.
- El nombre y extensión de un archivo puede tener hasta 255 caracteres incluyendo espacios.
- Además a la izquierda o encima del nombre aparece un icono gráfico representando al archivo.

A continuación veremos los procedimientos cotidianos a emplear en cualquier programa que encontramos en el menú de "Archivo": NUEVO, ABRIR, GUARDAR y GUARDAR COMO.



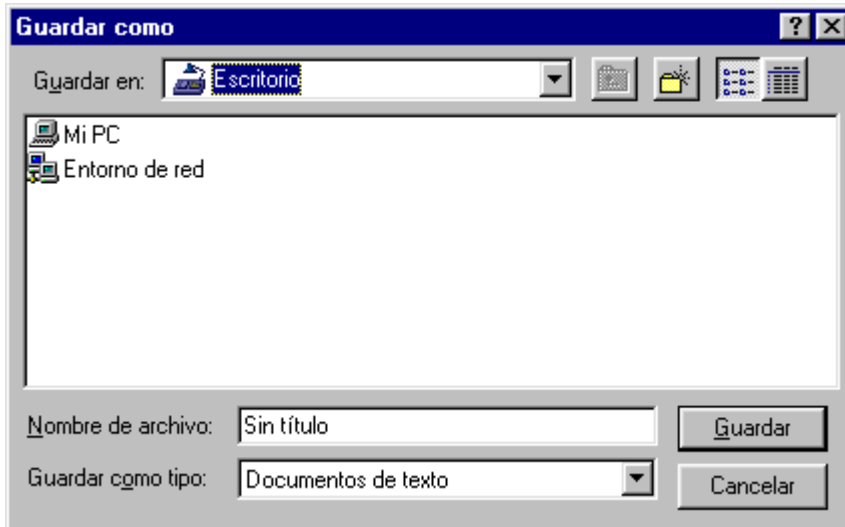
Para desarrollar el tema utilizamos el programa "Bloc de notas" situado en el menú de "Accesorios". Este programa es un editor de texto muy sencillo.

GUARDAR

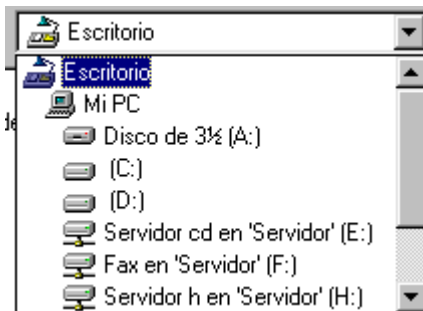
El concepto de guardar un documento se realiza como en la oficina de nuestro ejemplo: hemos escrito una carta en nuestra mesa y después la almacenamos en nuestro armario archivador dentro de la carpeta donde se guardan las cartas.

En el ejemplo de nuestro ordenador este proceso es similar:

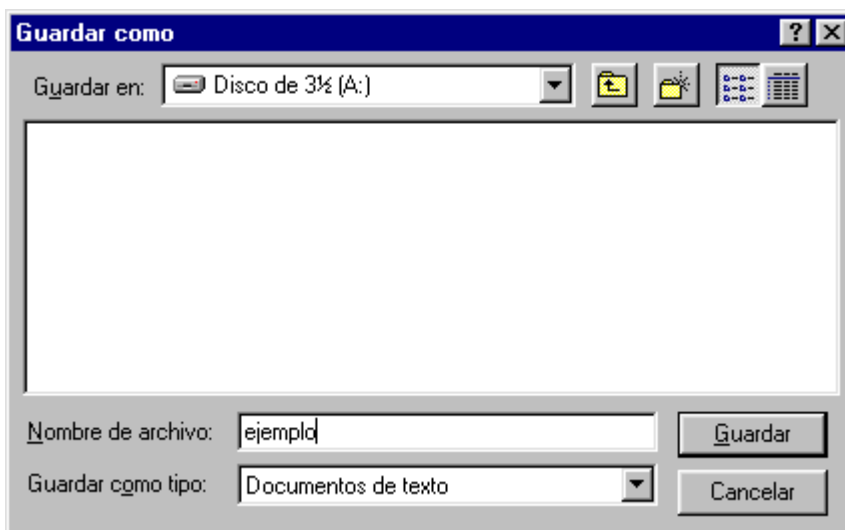
1. Ejecutamos el bloc de notas colocándose en el escritorio
2. Escribimos un pequeño texto (Ej nombre y apellidos)
3. Acabado el texto hay que almacenarlo, porque sino al apagar el ordenador la información se perderá (en la oficina no ocurre esto ya que podemos dejar la mesa con documentos, en el ordenador no).
4. Picamos la opción "Guardar" del menú y aparece este cuadro de diálogo. (El cuadro es el de GUARDAR COMO ya que el archivo no había sido guardado anteriormente).



5. Picamos en la lista desplegable "Guardar en" y seleccionamos la unidad A: donde guardaremos el documento.



6. Después escribimos el nombre que vamos a dar al archivo EJEMPLO



7. Llegados a este punto solo hay que picar sobre el botón de Guardar.

Ya está almacenado en el disquete el archivo EJEMPLO.TXT, cuya extensión TXT ha sido añadida automáticamente por el bloc de notas. La próxima vez que picamos sobre la opción GUARDAR no aparece el cuadro de diálogo sino que directamente se actualiza el contenido en el archivo y se almacena.

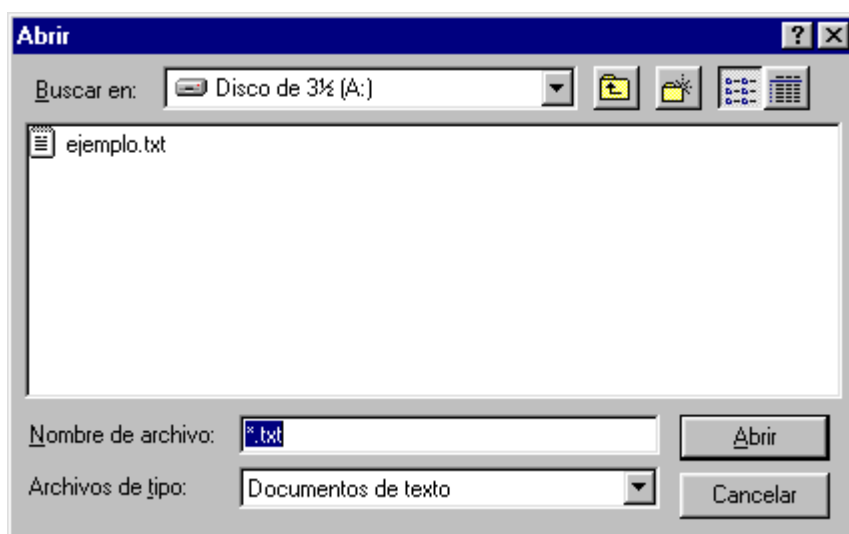
Resumiendo todo el proceso queda organizado así:

- Comenzamos a escribir un documento
- La primera vez que picamos GUARDAR aparece el cuadro en el que indicamos donde y que nombre va a tener el documento.
- Continuamos escribiendo el documento y de vez en cuando picamos guardar (actualizando en el disco los últimos cambios realizados).

ABRIR

La opción abrir permite recuperar un archivo almacenado anteriormente. En nuestro símil equivale a coger del armario un documento y dejarlo sobre la mesa.

1. Aparece este cuadro de diálogo



2. Seleccionamos la unidad y carpeta donde está el archivo que queremos abrir en la lista "Buscar en"
3. Marcamos con el ratón el archivo de entre los que aparecen en la lista
4. Picamos sobre el botón "Abrir"

Nota: Si el archivo que queremos abrir (o guardar en su caso) se encuentra dentro de una carpeta el proceso lleva un proceso

intermedio, en el cual para activar la carpeta hay que hacer doble click sobre ella en el cuadro de diálogo.

GUARDAR COMO

La opción guardar como implica siempre la aparición inmediata del cuadro de diálogo pudiendo indicar un nuevo nombre para el archivo y/o una nueva ubicación donde guardarlo.

Un ejemplo típico del empleo de GUARDAR COMO:

1. Disponemos de un documento almacenado con el nombre FELICITACION.TXT dirigido a una persona.
2. Queremos felicitar a otra persona para lo cual abrimos el citado documento.
3. Modificamos parte del documento como el nombre del destinatario, la fecha etc...
4. Por último queremos almacenar el documento, pero si utilizamos la opción GUARDAR machacamos la primera felicitación, por lo que utilizaremos la opción GUARDAR COMO
5. En el cuadro de diálogo sustituyo el nombre FELICITACION por OTRA FELICITACION
6. De esta manera conservo el documento original FELICITACION.TXT y almaceno el nuevo en el archivo OTRA FELICITACION.TXT

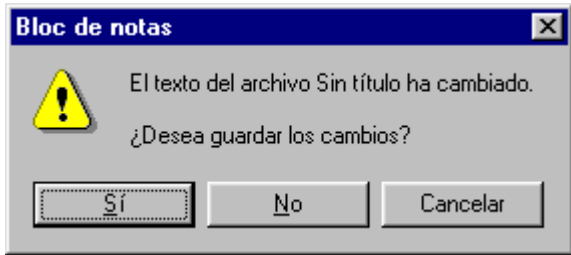
NUEVO

La opción nueva en el Bloc de Notas, implica que el documento actual se cierra, y se comienza un nuevo documento vacío.

Este hecho, ocurre también con la opción abrir, de manera que si abrimos un documento, el anterior ha de cerrarse.

En nuestro símil la situación equivale a que en la mesa no cabe más que un documento, y por lo tanto si deseamos comenzar uno nuevo o recuperar uno del armario, primero "cerraremos" (almacenando ó desechando) el documento que se encuentra en nuestra mesa.

Este cuadro de diálogo aparece en el bloc de notas cuando pulsamos nuevo sin haber guardado previamente los cambios.



Este proceso que acabamos de explicar es de gran importancia, ya que no todos los programas funcionan así. De este modo, programas como Word y Excel permiten abrir a la vez más de un archivo. A estos programas se les llama programas multidocumento y presentan dentro de su ventana, otras ventanas internas en las que se muestra cada documento. En estos programas la opción nuevo o Abrir no implica que el documento actual haya de cerrarse, sino que permanece como está y aparece una nueva ventana donde se muestra el nuevo documento. En el símil equivale a que en nuestra mesa si cabe más de un documento, pudiendo entonces tener unos cuantos sobre ella.

REGISTROS

Los registros del procesador se emplean para controlar instrucciones en ejecución, manejar direccionamiento de memoria y proporcionar capacidad aritmética. Los registros son espacios físicos dentro del microprocesador con capacidad de 4 bits hasta 64 bits dependiendo del microprocesador que se emplee. Los registros son direccionables por medio de una viñeta, que es una dirección de memoria. Los bits, por conveniencia, se numeran de derecha a izquierda (15,14,13.... 3,2,1,0), los registros están divididos en seis grupos los cuales tienen un fin específico. Los registros se dividen en:

- Registros de segmento
- Registros de apuntadores de instrucciones
- Registros apuntadores
- Registros de propósitos generales
- Registro índice
- Registro de bandera.

CAMPOS

El campo consiste de un grupo de caracteres unidos tratados como una sola unidad. Define la localización física de almacenamiento de una unidad de datos o información. Describen una propiedad de una entidad o sujeto para el cual los datos han sido almacenados. Un ejemplo es: la nómina. Hay un campo para el nombre, un campo de horas trabajadas, un campo de cuotas, un campo para la tasa de

impuestos, etc. Durante el proceso, los caracteres de cada uno serán usados como una unidad.

CLAVE

Tener claro el concepto de palabra clave, es fundamental para lograr una buena posición dentro de los buscadores. Los buscadores y algunos directorios de búsqueda utilizan las palabras clave que se ingresan para localizar los enlaces dentro de su base de datos y presentarlos como resultados.

Estos resultados aparecen listados en páginas que contienen 10, 15 o 20 de estos enlaces. Esta forma de presentación varía de acuerdo a cada Buscador. El quedar colocado entre los primeros significa una correcta utilización de las etiquetas (metatags). Esto significa que nuestro título (title), descripción (description) y palabras clave (keywords) deben contener la información más apropiada y contundente relacionada con nuestro sitio Web.

TABLA DE BASE DE DATOS

Tabla en las bases de datos, se refiere al tipo de modelamiento de datos, donde se guardan los datos recolectados por un programa. Su estructura general se asemeja a la vista general de un programa de Hoja de cálculo.

Las tablas se componen de dos estructuras:

- Campo: Corresponde al nombre de la columna. Debe ser único y además de tener un tipo de dato asociado.
- Registro: Corresponde a cada fila que compone la tabla. Allí se componen los datos y los registros. Eventualmente pueden ser nulos en su almacenamiento.

En la definición de cada campo, debe existir un nombre único, con su tipo de dato correspondiente. Esto es útil a la hora de manejar varios campos en la tabla, ya que cada nombre de campo debe ser distinto entre sí.

A los campos se les puede asignar, además, propiedades especiales que afectan a los registros insertados. El campo puede ser definido como índice o autoincrementable, lo cual permite que los

datos de ese campo cambien solos o sean el principal indicar a la hora de ordenar los datos contenidos.

Cada tabla creada debe tener un nombre único en la cada Base de Datos, haciéndola accesible mediante su nombre o su sinónimo (dependiendo del tipo de base de datos elegida).

La estructura de las tablas viene dado por la forma de un archivo plano, los cuales en un inicio se componían de un modo similar.

ALMACENAMIENTO DE DATOS

La única estructura que ofrece la memoria es la organización de sus elementos en bytes. Por tanto, para almacenar los datos que manipula un procesador es imprescindible saber de antemano su tamaño. El tamaño de algunos datos básicos viene definido por la arquitectura del propio procesador. Por ejemplo, el lenguaje máquina del Intel Pentium contiene instrucciones máquina para operar enteros de 32 bits. Esto no quiere decir que el procesador no pueda manejar enteros de otros tamaños, sino que el procesador manipula estos de forma mucho más rápida y eficiente. Números de otros tamaños pueden ser manipulados igualmente pero con un coste mayor en tiempo de ejecución.

Los lenguajes de programación de alto nivel como Java definen un conjunto de datos denominados “básicos” y un conjunto de mecanismos para definir datos complejos en base a ellos. Como los programas escritos en estos lenguajes deben ejecutar en diferentes equipos con diferentes procesadores, es difícil definir el tamaño de los datos tal que se ajuste a todos ellos. El compilador se encarga de transformar las operaciones escritas en lenguaje de alto nivel en las instrucciones más adecuadas para manipular los datos en el procesador pertinente.

MODELOS DE ALMACENAMIENTO DE DATOS

En general existen dos modelos de almacenamiento de datos en los sistemas de información.

a) El modelo tradicional de archivos que se construye con los siguientes elementos:

1.- Variables Registros que como ya se indico son variables que permiten almacenar conjuntos de datos de diverso tipo.

También se pueden definir como representaciones simbólicas y programáticas de entidades lógicas de información ejemplos de variables registros son alumnos, empleados, clientes, proveedores, productos, autos, etc.

Estas variables registros también ocupan programas o rutinas de programas para procesarlas por ejemplo un procedimiento, modulo o subrutina se encargara de capturar los datos que contendrá la variable registro otro procedimiento para corregir los datos que ya contiene, otro procedimiento para desplegarlos en pantalla ya cuando ha sido capturada y así sucesivamente.

2.-Archivos, que en principio pueden entenderse como una especie de almacenes o bodegas para almacenamiento de datos en forma permanente en disco es decir un archivo de empleados en disco contiene todos los datos de todos los empleados de una empresa.

Igualmente los archivos ocupan su propios programas o subrutinas o procedimientos especializados por ejemplo, procedimientos para crear los archivos, para almacenar o dar de altas los registros en el archivo, procedimientos para buscar un registro determinado, procedimiento para dar de baja un registro, etc.

3.- Una aplicación, que es un programa que se encarga de coordinar todos los programas descritos y presentar a usuarios de manera clara, fácil, accesible y entendible.

Salta a la vista que construir un sistema de información por ejemplo para una tienda de vídeo o para un refaccionaria, etc. involucra un gran cantidad de trabajo de programación puesto que hay que programar muchas variables registros, muchos archivos en disco y construir una o muchas aplicaciones.

Este modelo se usa todavía en la actualidad pero es obvio que mejores maneras, mas rápidas, seguras y eficientes existen en la actualidad para resolver estos problemas, y esto nos lleva al segundo modelo de datos.

b) Modelo de Bases de Datos Relacionales, este modelo intenta simplificar la construcción de sistemas de información como los antes descritos, este modelo solo incluye en forma simple los siguientes elementos:

b.1) Tablas, es una combinación de las variables registro y de los archivos del modelo anterior.

Es decir cuando un programador moderno define o declara una tabla en un programa, realmente esta haciendo dos cosas por el precio de una, es decir crea una variable registro en memoria que almacenara los datos y al mismo tiempo ya esta creando un archivo en disco que se llamara igual que la tabla o variable registro y que automáticamente se convertirá en un espejo de la tabla en memoria, es decir cuando se cargan los datos en la tabla en memoria, también se estarán cargando en disco.

Otra vez cuando el programador escribe código para capturar los datos y mandarlos a la tabla en pantalla-memoria, realmente también lo esta haciendo para darlos de alta en disco.

b.2) Aplicación, que tiene la misma función que en el modelo anterior.

No confundir este concepto de tablas en base de datos con el concepto de tablas vistos en el capítulo de arreglos.

Como se observa en este modelo es mas sencillo construir sistemas de información puesto que la parte programática se reduce ampliamente.

ORGANIZACIÓN DE ARCHIVOS (SECUENCIAL, DIRECTO, CON INDICE)

Existen dos maneras de acceder y buscar registros de un archivo:

Secuencial

Características

Consecutivamente

Respetando el orden de aparición en el archivo

El orden de complejidad será $O(n)$ lo cual implica que es demasiado lento para grandes volúmenes de datos, $O(n/2)$ en promedio.

¡SE UTILIZA CUANDO!

Se está buscando en un archivo de texto algún patrón (pattem)

Archivos con pocos registros

Archivos que no necesitan "búsquedas" por ejemplo los respaldos en cintas

Cuando de antemano se sabe que se recuperarán muchos resultados (vale la pena la espera)

Existe una técnica que permite aumentar la velocidad de estos accesos llamado "Blocking". Se basa en leer bloques de registros en lugar de leer uno por uno. El leer un bloque es más lento porque se traen en cada viaje al disco más datos, pero nuevamente la separación se hace en memoria donde la velocidad es mucho mayor y ahí se gana tiempo.

Aunque el rendimiento mejora considerablemente en realidad no es algo considerable ya que el número de las comparaciones para buscar el patrón o valor que se requiere sigue siendo el mismo.

DIRECTO

CARACTERÍSTICAS

El orden de complejidad será $O(1)$

Se basa en las funciones de seek

Para obtener un buen rendimiento se deben hacer los registros de una longitud cuyo múltiplo sea del tamaño de un sector del disco. Si el sector es de 512 bytes y nuestro registro mide 30, lo más adecuado es que mida 32, ya que $32 \times 16 = 512$ (16 registros en un viaje al disco)

Se utiliza cuando:

Los registros son de longitud fija

Tenemos una manera de saber en que posición del archivo está un registro,

Ej. Campo id=número del registro en el archivo

FORMAS DE ACCESOS

ACCESO SECUENCIAL: Exige el tratamiento de elemento, para esto es necesario una exploración secuencial comenzando desde el primer momento (Pascal permite este acceso)

Secuenciales: archivo de texto que debe ser leído del principio hasta el final.

ACCESO DIRECTO: Permite procesar o acceder a un elemento determinado y referencia directamente por su posición en el soporte de almacenamiento (Turbo Pascal permite este acceso).

Aleatorios: es un archivo con registros de un mismo largo. Un programa puede acceder directamente cualquier registro sin tener que leer los registros previos.

BINARIOS: es un archivo que lee byte por byte sin asumir ninguna estructura. Los archivos Binarios no son un nuevo tipo de archivo, pero sí una nueva forma de manipular cualquier tipo de archivo. Las técnicas de archivo binarios permiten leer o cambiar cualquier byte de un archivo. Son herramientas extremadamente potentes, pero como toda herramienta potente debe manejarse con cuidado

Entre otras características, a las técnicas de archivo binario no les preocupa los caracteres EOF intercalados (Control+Z = Chr\$(26)) que pueda tener un archivo.

Ejemplo Open NombreArchivo for Binary as #NumeroArchivo.

Instrucciones para manejar archivos

OPEN: reserva un espacio del buffer para la data que moverá entre el programa y los archivos. La estructura es:

```
OPEN filespec FOR {Append|Binary|Input|Output|Random} As #filename
```

Por ejemplo:

```
OPEN "C:\Windows\AddrBook.ini" FOR Input As #1
```

Filespec: Es la localización de archivo en el que se trabajará, incluyendo usualmente el drive y path.

```
"C:\Windows\AddrBook.ini"
```

{Append| Binary| Input| Output| Random} El programador tiene que seleccionar uno. Binary y Random se utiliza para archivos binarios y aleatorios. Append, Input y Output son usados con archivos secuenciales. Un archivo secuencial no puede ser abierto para leer y escribir simultáneamente. Output es usado para escribir en el archivo. Input es usado para leer del archivo. Append es usado para colocar data al final de un archivo existente.

#filename: es necesario asignar un número al archivo. El número puede estar en el rango de #1 a #511 y es usado por Visual Basic para identificar el archivo.

CLOSE: para cerrar un archivo. La estructura es:

CLOSE #filenumber

Por ejemplo:

CLOSE #1

WRITE: envía data del programa al archivo secuencial. La estructura es:

WRITE #filenumber, [OutputList]

Por ejemplo:

WRITE #1, UserName, UserCompany, SerialNumber

WRITE es la operación opuesta al INPUT. Las expresiones en el OutputList son separadas por comas. WRITE inserta comillas y comas a la data que envía al archivo.

INPUT: lee data del archivo. La estructura es:

INPUT #filenumber, InputList

Por ejemplo:

INPUT #1, UserName, UserCompany, SerialNumber

Declaración y asignación de archivos

La declaración de un archivo con tipo se efectúa con la ayuda de las palabras reservadas file of.

El procedimiento de asignación es idéntico al utilizado anteriormente.

Ejemplo:

Type

datos = record

clave : integer;

nombre : string[30];

puesto : string[20];

sueldo : real;

estado : boolean;

{true activo,false baja lógica}

end;

Var

archivo:file of datos;

begin

Assign(archivo,'empleado.dat');