

Fundamentos de Organización de Datos – Cursada 2025

Bibliografía

- Introducción a las Bases de Datos. Conceptos básicos (Bertone, Thomas).
- Estructuras de Archivos (Folk-Zoellick).
- Files & Databases: An introduction (Smith-Barnes).

1) Archivos

Persistencia de datos

Una **base de datos** es una colección de datos relacionados, específicamente de archivos diseñados para servir a múltiples aplicaciones. Estos datos representan hechos conocidos que pueden registrarse y que tienen un resultado implícito.

Propiedades implícitas de una BD

Una base de datos...

1. ...representa algunos aspectos del mundo real, a veces denominado Universo de Discurso.
2. ...es una colección coherente de datos con significados inherentes. Un conjunto aleatorio de datos no puede considerarse una BD. O sea los datos deben tener cierta lógica.
3. ...se diseña, construye y completa de datos para un propósito específico. Está destinada a un grupo de usuarios concretos y tiene algunas aplicaciones preconcebidas en las cuales están interesados los usuarios.
4. ...está sustentada físicamente en archivos en dispositivos de almacenamiento persistente de datos.

Definiciones de archivo

Un archivo es una colección de...

1. ...registros guardados en almacenamiento secundario.
2. ...datos almacenados en dispositivos secundarios de memoria.
3. ...registros que abarcan entidades con un aspecto común y originadas para algún propósito particular.

Hardware

1. Almacenamiento primario: Memoria RAM

2. Almacenamiento secundario (DR): platos, superficies, pistas, sectores, cilindros.

Organización de un archivo

1. En una **secuencia de bytes**: archivos de texto dónde se leen o recuperan caracteres sin formato previo. Una palabra se determina por un conjunto de caracteres finalizados en blanco (convención).
2. En **registros y campos**: un campo es la unidad más pequeña lógicamente significativa de un archivo; un registro es un conjunto de campos agrupados que definen un elemento del archivo.

Acceso de archivos

1. Secuencial físico: acceso a los registros uno tras otro y en el orden físico en el que están guardados (sin orden específico, simplemente en como llegaron).
2. Secuencial indizado (lógico): acceso a los registros de acuerdo al orden establecido por otra estructura o criterio.
3. Directo: se accede a un registro determinado sin necesidad de haber accedido a los predecesores.

Tipos de archivos

Se determinan de acuerdo a su forma de acceso:

1. Serie: cada registro es accesible solo luego de procesar su antecesor, simples de acceder (acceso secuencial físico).
2. Secuencial: los registros son accesibles en orden de alguna clave (acceso secuencial indizado/secuencial lógico).
3. Directo: se accede al registro deseado (acceso directo).

Operaciones de archivos

Los **archivos físicos** aparecen en el disco y están a cargo del SO. En cambio, los **archivos lógicos** se definen dentro del programa. El archivo se puede definir de dos formas:

- Como variable:
`Var archivo: file of Tipo_de_dato;`
- Como tipo:
`Type archivo: file of Tipo_de_dato;`
`Var arch: archivo;`

Relación con el SO: se debe asignar la correspondencia entre el nombre físico y el lógico:

```
Assign(n_logico, n_fisico);
```

Rewrite: de solo escritura (creación):

```
Rewrite(n_logico);
```

Reset: lectura/escritura (apertura del archivo):

```
Reset(n_logico);
```

Close: cierre de archivo. Luego del ultimo dato del archivo se coloca la marca EOF (End Of file), es decir, al final del archivo:

```
Close(n_logico);
```

Read: leer un archivo sobre una variable del mismo tipo del archivo:

```
Read(n_logico, variable);
```

Write: escribo en el buffer:

```
Write(n_logico, variable);
```

Estas operaciones (read y write) leen y/o escriben sobre los buffers relacionados a los archivos No se realizan directamente sobre la memoria secundaria.

```
Program Generar_Archivo;
type archivo = file of integer; {definición del tipo de dato para el archivo }
var
  arc_logico: archivo; {variable que define el nombre lógico del archivo}
  nro: integer; {nro será utilizada para obtener la información de
  teclado}
  arc_fisico: string[12]; {utilizada para obtener el nombre físico del archivo
  desde teclado}

begin
  write( 'Ingrese el nombre del archivo:' );
  read( arc_fisico ); { se obtiene el nombre del archivo}
  assign( arc_logico, arc_fisico );
  rewrite( arc_logico ); { se crea el archivo }
  read( nro ); { se obtiene de teclado el primer valor }
  while nro <> 0 do begin
    write( arc_logico, nro ); { se escribe en el archivo cada número }
    read( nro );
  end;
  close( arc_logico ); { se cierra el archivo abierto oportunamente con la
instrucción
  rewrite }
end.
```

EOF: fin del archivo (es un booleano):

```
EOF(n_logico); // true o false
```

FileSize: tamaño del archivo:

```
FileSize(n_logico); // función
```

FilePos: posición dentro del archivo:

```
FilePos(n_logico); // función
```

Seek: ir a una posición del archivo. Siempre se cuenta desde el comienzo del archivo (principio = 0):

```
Seek(n_logico, posicion); // procedimiento
```