

Fundamentos de Organización de Datos

Archivos

La Cátedra

Profesores:

Mg. Rodolfo Bertone Viernes 8 – 11 hs

Mg. Thomas Pablo Jueves 14 – 17 hs

Lic. Marrero Luciano Redictado DBD Jueves 17 – 20 hs

Trabajos Prácticos:

JTP: Lic. Serveto Arturo Redictado DBD Jueves 17 – 20
hs

JTP: Lic. Sobrado Ariel Lunes 8 – 11 hs

JTP: Lic. Restelli Noelia Martes 11 – 14 hs

JTP: Lic. Nucilli Emanuel Martes 18 – 21 hs

La Cátedra

▮ Clases

- ▮ Teóricas
- ▮ Explicaciones de Prácticas (donde se presentan ejemplos)
- ▮ Prácticas
- ▮ Se utilizará la plataforma Ideas – Curso IBBDD / FOD

▮ Para aprobar la cursada

- Un Parcial - 1° Fecha Martes 06/06 18 hs
- Dos recuperatorios - 2° Fecha Martes 27/06 18 hs
- 3° Fecha Martes 11/07 18 hs

Bibliografía

- Introducción a las Bases de Datos. Conceptos Básicos (Bertone, Thomas)
- Estructuras de Archivos (Folk-Zoellick)
- Files & Databases: An Introduction (Smith-Barnes)
- Fundamentos de Bases de Datos (Korth Silvershatz)

Fundamentos de Organización de Datos

Archivos

Tipos de Archivos

Registros de longitud fija (File of <tipo_dato>)

Texto: Caracteres estructurados en líneas.

Lectura/escritura con conversión automática de tipos.

El acceso es exclusivamente secuencial.

Útiles para importar y exportar datos.

Bloques de bytes (File): Se verá más adelante en el curso.

Acceso a la información

Métodos de acceso

Secuencial: El acceso a cada elemento de datos se realiza luego de haber accedido a su inmediato anterior.

Directo: Se recupera un elemento de datos de un archivo en un solo acceso.

Secuencial indizado: El acceso a los elementos de un archivo se realiza mediante una estructura externa.

No tiene en cuenta el orden físico

Operaciones básicas

Definición de Archivos

Dos formas:

- **var** archivo_logico: **file of** tipo_de_dato;
- **type**
 archivo = **file of** tipo_de_datos;
var archivo_logico: archivo

type

```
persona = record
    dni: string[8];
    apellido: string[25];
    nombre: string[25];
    direccion: string[25];
    sexo: char;
end;
archivo_enteros = file of integer;
archivo_string = file of string;
archivo_personas = file of persona;
```

var

```
enteros: archivo_enteros;
texto: archivo_string;
personas: archivo_personas;
```

Ejemplo

Operaciones

```
assign(nombre_logico, nombre_fisico);
```

Realiza una correspondencia entre el archivo lógico y archivo físico.

Ejemplo:

```
assign(enteros, 'c:\archivos\enteros.dat');
```

```
assign(texto, ' c:\archivos\texto.dat');
```

```
assign(personas, 'c:\archivos\personas.dat');
```

Operaciones

Apertura y creación de archivos

rewrite(nombre_logico); → Crea un archivo

reset(nombre_logico); → Abre un archivo existente

Ejemplo:

rewrite(enteros);

reset(personas);

Operaciones

Cierre de archivos

close(nombre_logico);

Transfiere la información del buffer al disco.

Ejemplo:

close(enteros);

close(personas);

Operaciones

Lectura y escritura de archivos

```
read(nombre_logico, var_dato);  
write(nombre_logico, var_dato);
```

El tipo de dato de la variable `var_dato` es igual al tipo de datos de los elementos del archivo.

Ejemplo:

```
read(enteros, e);    ➡ e : integer;  
write(personas, p); ➡ p : persona;
```

Operaciones adicionales

EOF(nombre_logico);

Controla el fin de archivo.

fileSize(nombre_logico);

Devuelve el tamaño de un archivo.

filePos(nombre_logico);

Devuelve la posición actual del puntero en el archivo.

En longitud fija, los registros se numeran de 0..N-1.

seek(nombre_logico, pos);

Establece la posición del puntero en el archivo.

Ejercicio

```
program creacion_archivo;
type
    persona = record
        dni: string[8]
        apellidoyNombre : string[30];
        direccion : string[40];
        sexo      : char;
        salario   : real;
    end;
    archivo_personas = file of persona;

var
    personas: archivo_personas;
    nombre_fisico: string[12];
    per: persona;
```

begin

```
write('Ingrese el nombre del archivo: ');  
readln(nombre_fisico)
```

{enlace entre el nombre lógico y el nombre físico}

```
assign(personas, nombre_fisico);
```

{apertura del archivo para creación}

```
rewrite(personas);
```


{lectura del DNI una persona}

readln(per.dni);

while (per.dni <> '') do begin

{lectura del resto de los datos de la persona}

readln(per.apellidoyNombre);

readln(per.direccion);

readln(per.sexo);

readln(per.salario);

{escritura del registro de la persona en el archivo}

write(personas, per);

{lectura del DNI de una nueva persona}

readln(per.dni);

end;

{cierre del archivo}

close(personas);

end