

CONCEPTOS DE BASES DE DATOS

Introducción a Archivos

PRACTICA – MODALIDAD VIRTUAL

- Consulta sobre la explicación de práctica, los martes de 14hs a 15hs.
- Existirá un horario de consulta práctica online (martes 15hs-17hs). Se dividirán los alumnos en 4 grupos, cada uno con un auxiliar docente asignado. Al inicio del curso serán asignados a un grupo determinado.
- Por el foro se puede realizar consultas puntuales.
- El foro va a estar dividido por practica, y para cada enunciado de la practica existirá un posteo sobre el que se podrá realizar las consultas. De esta forma quedan todas las consultas de un enunciado en un solo posteo.

ARCHIVOS EN PASCAL

Tipos de archive:

- **De longitud fija (File of <tipo_info>)**
- **De unidades indeterminadas (File)**
- **De Texto (Text):** caracteres estructurados en líneas
→ lectura/escritura con conversión automática de tipos, con acceso secuencial

ARCHIVOS EN PASCAL

Operaciones:

Asignación (Assign): una variable de tipo archivo representa a cualquier archivo del tipo definido → se le debe asociar la ubicación física real del archivo para poder utilizar dicha variable.

Apertura: establece la dirección de transferencia entre el programa y el archivo y la posición inicial del programa en el archivo

- **Rewrite:** crea un archivo con el nombre asignado a la variable para escribir, y si ya existe borra su contenido, posicionando al programa al comienzo del archivo
- **Reset:** abre el archivo con el nombre asignado a la variable tanto para leer como para escribir, posicionando al programa al comienzo del archivo
- **Append:** abre el archivo de texto con el nombre asignado a la variable para escribir, posicionando al programa al final del archivo

ARCHIVOS EN PASCAL

Operaciones (cont.)

Posicionamiento (Seek): posiciona al programa en el número relativo de unidad que se especifica (no se permite en archivos de texto)

Lectura: copia bytes desde archivo a la variable que se proporciona, en la posición donde está el programa dentro del archivo y posiciona al programa en la posición siguiente → **Read** (archivos de registros y de texto), **ReadLn** (archivos de texto), **BlockRead** (archivo de unidades indeterminadas)

Escritura: copia bytes desde la variable que se proporciona al archivo, en la posición donde está el programa en el archivo, y posiciona al programa en la siguiente posición → **Write** (archivos de registros y de texto), **WriteLn** (archivos de texto), **BlockWrite** (archivo de unidades indeterminadas)

ARCHIVOS EN PASCAL

Operaciones (cont.)

Cierre (Close): desactiva la dirección de transferencia entre el programa y el archivo y el posicionamiento del programa en el archivo.

Consultas: permiten obtener características del archivo o de la posición del programa en el archivo

- **FileSize** (tamaño en unidades del archivo)
- **FilePos** (número relativo de unidad sobre la que está posicionado el programa en el archivo)
- **EoF** (si el programa está posicionado a continuación de la última unidad del archivo)
- **EoL** (si el programa está posicionado a continuación de la última línea del archivo -sólo archivos de texto)

Truncamiento (Truncate): trunca el archivo a partir de la posición actual

ACCESO A LA INFORMACIÓN

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

Secuencial indizado: El acceso a los elementos de un archivo se realiza mediante algún tipo de organización previa, independiente del orden físico.

Directo: Un solo acceso para recuperar cada elemento.

EJEMPLO-ARCHIVO DE TEXTO- BINARIO

Type

```
tRegistroVotos=Record
    codProv: integer;
    codLoc: integer;
    nroMesa: integer;
    desc:string;
    cantVotos: integer;
End;
tArchVotos=File of tRegistroVotos;
```

Var

```
opc: Byte;
nomArch, nomArch2: String;
arch: tArchVotos;
```

carga: Text;*// este archivo tiene los datos de los votos, se lee de el y genera el binario. Además, luego se muestra como a partir del archivo de registros se escribe en el texto*

```
votos: tRegistroVotos;
```


EJEMPLO-ARCHIVO DE TEXTO- BINARIO

```
begin
WriteLn('VOTOS');
WriteLn;
WriteLn('0. Terminar el Programa');
WriteLn('1. Crear un archivo');
WriteLn('2. Abrir un archivo existente');
Repeat
Write('Ingrese el nro. de opcion: '); ReadLn(opc);
If (opc=1) or (opc=2) then begin
    WriteLn;
    Write('Nombre del archivo de votos: ');
    ReadLn(nomArch);
    Assign(arch, nomArch);
end;
```

EJEMPLO-ARCHIVO DE TEXTO- BINARIO

Case opc of

1: begin

Write('Nombre del archivo de carga: ');

ReadLn(nomArch2);

Assign(carga, nomArch2); Reset(carga); Rewrite(arch);

Repeat

With votos do ReadLn(carga, codProv, codLoc, nroMesa, cantVotos, desc);

Write(arch, votos);

until eof(carga);

Write('Archivo cargado. Oprima tecla de ingreso para continuar...');

ReadLn;

Close(arch); Close(carga)

end;

Qué sucede si hay más de un campo string en el archivo de texto?

EJEMPLO-ARCHIVO DE TEXTO- BINARIO

2: begin

Reset(arch);

Rewrite(carga);

WriteLn;

Repeat

Read(arch, votos);

With votos do WriteLn(codProv:5, codLoc:5, nroMesa:5, cantVotos:5, desc:5);

With votos do WriteLn(carga, codProv, codLoc, nroMesa, cantVotos,desc);

until eof(arch);

Close(arch); Close(carga);

WriteLn;

Write('Oprima cualquier tecla para continuar...');

ReadLn

end;