



Fundamentos de Organización de Datos

Curso 2023

Profesores Rodolfo Bertone y Pablo Thomas

La cátedra

➤ Clases

- Teóricas
- Explicaciones de Prácticas (donde se presentan ejemplos)
- Prácticas
- Se utilizará la plataforma <https://asignaturas.info.unlp.edu.ar> (Moodle Facultad)

➤ Para aprobar la cursada

- Un Parcial (con 3 temas)
- Dos recuperatorios

La Materia



	Semana	Teoría	Práctica
	06/03	Introducción. Archivos, Algorítmica Básica	Sin Actividad
	13/03	Archivos, Algorítmica Básica, Algorítmica Clásica	Archivos Algorítmica Básica
	20/03	Archivos, Algorítmica Clásica	Archivos Algorítmica Básica
		Proceso de Baja en Archivos	
	27/03	Archivos con Registros de Longitud Variable	Archivos Algoritmica Clasica
	03/04	Semana Santa	Archivos Algorítmica Clásica
	10/04	Búsqueda de información en Archivos. Indices	Archivos Algorítmica Clásica Bajas, Registros de Longitud Variable
	17/04	Arboles Binarios, AVL. Introducción a Arboles B	Bajas, Registros de Longitud Variable
	24/04	Arboles B, Creación Búsqueda Eliminación, Performance (solo el jueves)	Arboles,
	01/05	Arboles B *	Arboles
	08/05	Arboles B+	Arboles
	15/05	Hashing	Hashing
	23/05	Sin teoría (feriado)	Hashing
	29/05	Hashing	Consulta
	05/06	Consulta	Parcial → Martes 06-06-2023 → Fecha tentativa
	12/06	Consultas	Consulta
	19/06	Consultas	Muestra de examen, consulta
	26/06	Consulta	Recuperatorio→ Martes 27-06-2023 → Fecha tentativa
	03/07	Consulta	Muestra de examen, consulta
	10/07	Examen teórico (viernes 14/07/2023 → fecha tentativa)	Recuperatorio→ Martes 11-07-2023 → Fecha tentativa
	17/07	Vacaciones	Vacaciones
	26/07	Vacaciones	Vacaciones

Horarios de Práctica

5

► Turno 1

► Martes de 11 a 14

► Turno 2

► Martes de 18 a 21

► Comienza semana del 13/03

► Inscripción en Turnos <https://fod.info.unlp.edu.ar/>
el viernes 10/03/2022 a partir de las 12 hs

Fechas de Examen

Fechas tentativas de parcial:

1º Fecha → Martes 06-06-2023

2º Fecha → Martes 27-06-2023

3º Fecha → Martes 11-07-2023

Fecha tentativa de evaluación teórica:

Fecha → 14-07-2023

Evaluación teórica - Cursada 2023

- ▶ Podrán acceder al examen todos los alumnos que se encuentren inscriptos en la asignatura.
- ▶ Se realizará un examen teórico.
- ▶ La aprobación del examen es con nota 5 o superior
- ▶ El examen NO tiene recuperatorio.
- ▶ Se deberá aprobar la cursada durante el semestre en que rinde los exámenes teóricos.
- ▶ Deberá anotarse para pasar el final de la asignatura como máximo hasta la mesa de marzo de 2024 inclusive.

Para el Redictado FOD (segundo semestre)

- ▶ Podrán acceder al redictado de FOD:
 - ▶ aquel alumno que no apruebe los tres temas de la materia y se presente como mínimo en dos de los tres fechas de parcial y
 - ▶ En cada uno de los tres temas de la materia deben tener, en al menos una de las fechas presentadas, una calificación diferente a blanco (es decir, demuestre intención de resolver el tema)

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

Clase 1

Agenda

Conceptos básicos de BD

- Definiciones
- Características

Archivos

- Introducción
- Operatoria básica

Conceptos básicos

Qué es una Base de Datos?

Es una colección de datos relacionados.

Colección de **archivos** diseñados para servir a múltiples aplicaciones

Un dato representa hechos conocidos que pueden registrarse y que tienen un resultado implícito.

Conceptos básicos

Propiedades implícitas de una BD

Una BD representa algunos aspectos del mundo real, a veces denominado Universo de Discurso.

Una BD 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.

Una BD 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

Una BD está sustentada físicamente en **archivos** en dispositivos de almacenamiento persistente de datos

Archivos

Definiciones

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

Archivos → algunos conceptos

Hardware

Almacenamiento
primario

Almacenamiento
secundario (DR)

Acceso a RAM
vs. Acceso a DR

Platos

Superficies

Pistas

Sectores

Cilindros

Comparaciones

Archivos → organización

Secuencia de bytes

- No se puede determinar fácilmente comienzo y fin de cada dato.
- Archivos de texto

Registros y Campos

- **Campo:** Unidad más pequeña, lógicamente significativa de un archivo
- **Registro:** Conjunto de campos agrupados que definen un elemento del archivo

Archivos → Acceso

Secuencial Físico: acceso a los registros uno tras otro y en el orden físico en el que están guardados

Secuencial indizado (lógico): acceso a los registros de acuerdo al orden establecido por otra estructura

- Ej: una guía telefónica, o índice temático del un libro

Directo: se accede a un registro determinado sin necesidad de haber accedido a los predecesores

De acuerdo a la forma de acceso

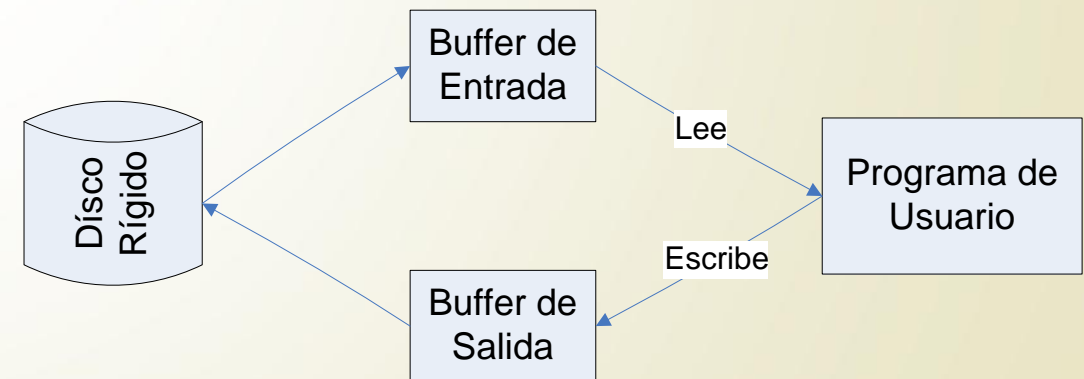
- **Serie** cada registro es accesible solo luego de procesar su antecesor, simples de acceder (*acceso secuencial físico*)
- **Secuencial** los registros son accesibles en orden de alguna clave (*acceso secuencial lógico*)
- **Directo** se accede al registro deseado (*acceso directo*)

Archivos

► Buffers

Memoria intermedia entre un archivo y un programa, donde los datos residen provisoriamente hasta ser almacenados definitivamente en memoria secundaria o donde los datos residen una vez recuperados de dicha memoria secundaria.

- Los buffers ocupan lugar en RAM
- SO encargado de manipular los buffers
- Como trabaja?



Archivos → Operaciones básicas

Dos niveles

- Físico (almacenamiento secundario)
- Lógico (dentro del programa)
 - Operaciones
 - Crear
 - Abrir
 - Read/Write
 - Eof
 - Seek(localización)

Archivos → Declaraciones

Utilizaremos la notación Pascal

Declaración de archivos

- Variable
 - `Var archivo: file of Tipo_de_dato;`
- Tipo
 - `Type archivo: file of Tipo_de_dato;`
 - `Var arch: archivo`

Archivos – Operaciones Básicas

▶ Ejemplos

```
Type emple = record
    nombre: string [20];
    direccion: string [40];
    edad: integer;
end;

numero = file of integer;
empleado = file of emple;
```

```
Var arch_num: numero;
```

```
Var arch_emp: empleado,
```

Archivos → Operaciones Básicas

► Relación con el sistema operativo

- Se tiene que establecer la correspondencia entre el nombre físico y nombre lógico

Assign (n_lógico, N_físico)

► Ejemplo

```
Program archivos;
...
Type emple = record
    nombre: string [20];
    direccion: string [40];
    edad: integer;
end;
numero = file of integer;
empleado = file of emple;
Var arch_num:numero;
Var arch_emp:empleado;
...
Begin
    ...
    Assign( arch_num, 'numeros.dat' )
    Assign( arch_emp, 'Personas_empleados.dat' )
    ...
End;
```

Archivos – Operaciones Básicas

```
Rewrite (nombre_logico);
```

- De solo escritura (creación)

```
Reset (nombre_logico);
```

- Lectura Escritura (apertura)

Nombre lógico representa una variable de tipo archivo sobre la que se realizó la asignación.

```
Close(nombre_logico);
```

- Cierre de archivo
- Esta instrucción indica que no se va a trabajar más con el archivo. Significa poner una marca de EOF (end of file) al final del mismo.

Archivos → Operaciones Básicas

```
Read(nombre_logico, variable);
```

```
Write(nombre_logico, variable);
```

Estas operaciones leen y/o escriben sobre los buffers relacionados a los archivos

No se realizan directamente sobre la memoria secundaria

En ambos casos la variable debe ser del mismo tipo que los elementos que se declararon como parte del archivo

Archivos → EJ 1 Crear un archivo

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.
```

Archivos → Operaciones adicionales

```
EOF( nombre_logico); (función)
```

- Fin de archivo
- Como trabaja?
- Hay que preguntar primero!!!

```
FileSize(nombre_logico); (función)
```

- Tamaño del archivo

```
FilePos( nombre_logico); (función)
```

- Posición dentro del archivo

```
Seek( nombre_logico, posición); (Procedimiento)
```

- Ir a una posición del archivo
- La posición se cuenta siempre desde el comienzo del archivo
- El primer lugar es el cero .

Archivos → EJ 2 Presentar en pantalla el archivo generado en ej1

```
Procedure Recorrido(var arc_logico: archivo );  
  var  nro: integer;  { para leer elemento del archivo}  
  begin  
    reset( arc_logico ); {archivo ya creado, para operar debe abrirse como de lect/escr}  
    while not eof( arc_logico) do begin  
      read( arc_logico, nro ); {se obtiene elemento desde archivo }  
      write( nro );           {se presenta cada valor en pantalla}  
    end;  
    close( arc_logico );  
  end;
```

Archivos → Ej 3 Modificación de Datos de un archivo

- ▶ Este caso involucra un archivo de datos previamente generado y consiste en cambiar sus datos.
- ▶ El archivo debe ser recorrido desde su primer elemento y hasta el último, siguiendo un procesamiento secuencial

{declaración de los tipos de datos necesarios para el problema.
Esta declaración se hace efectiva en el programa principal que
tiene al proceso Actualizar como uno de sus módulos }

```
Type registro = record  
    Nombre: string[20];  
    Direccion: string[20];  
    Salario: real;  
End;  
Empleados = file of registro;
```

Archivos → Ej 3 Modificación de Datos de un archivo (continuación)

```
Procedure actualizar (Var Emp:empleados); {se recibe como parám.  
por referencia}  
var E: registro;  
begin  
    Reset( Emp );  
    while not eof( Emp ) do begin  
        Read( Emp, E );  
        E.salario:=E.salario * 1.1;  
        Seek( Emp, filepos(Emp) -1 );  
        Write( Emp, E );  
    end;  
    close( Emp );  
end;
```