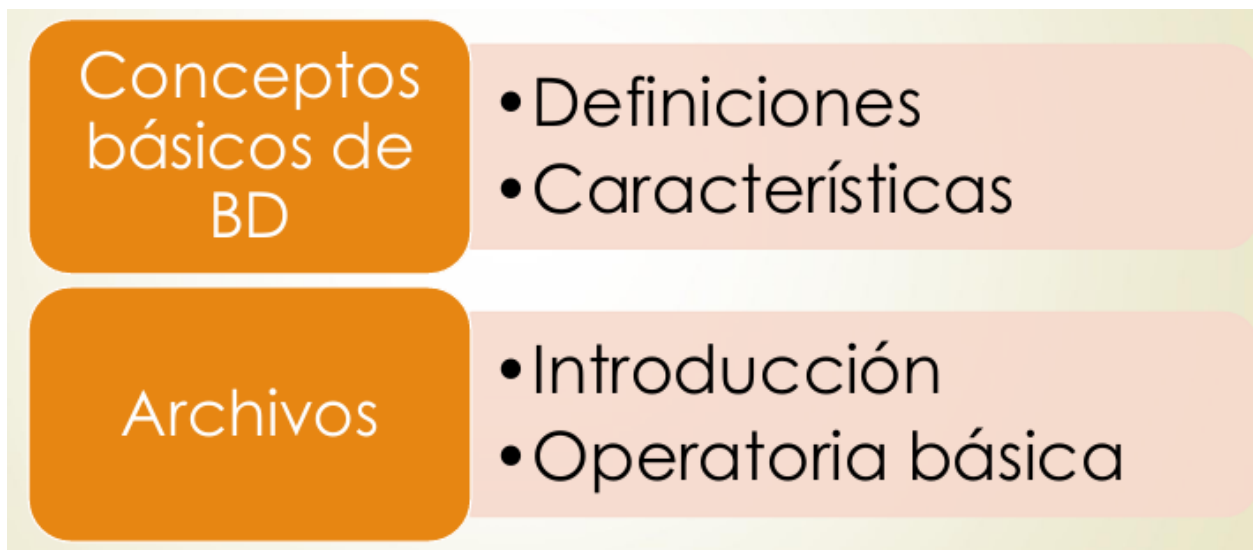


## FUNDAMENTOS DE LA ORGANIZACIÓN DE DATOS.

### CONTENIDOS



### CLASE 1:



### QUE ES UNA BASE DE DATOS?

- 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.

### Propiedades Implícitas de una BD.

- Representa algunos aspectos del mundo real, a veces denominado Universo de Discurso.
- Colección coherente de datos con significados inherentes (Deben tener cierta lógica).
- Una base de datos es diseñada, Construida y completa de datos para un propósito específico.}
- Sustentada físicamente en ARCHIVOS en dispositivos de almacenamiento **persistente de datos**.

**PERSISTENCIA DE DATOS** = Es la capacidad de mantener y almacenar datos a lo largo del tiempo.

### **ARCHIVOS :**

- 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 --ACCESO

**SECUENCIAL FISICO:** 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 con el 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.

**SERIE:** cada registro es accesible solo luego de procesar su antecesor, simples de accesos (**ACCESO SECUENCIAL FISICO**).

**SECUENCIAL:** Los registros son accesibles en orden de alguna clave (**ACCESO SECUENCIAL INDIZADO/SECUENCIAL LOGICO**).

**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

### ➤ 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;
```

FOD - Clase 1

Donde **n\_logico** vendría siendo el nombre de la variable, y el **n\_físico**: Es el nombre que le ponemos a un espacio en memoria cualquiera.

### 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 parte del archivo.

## Archivos// Operaciones adicionales .

**EOF( nombre\_logico); (función)**

- Fin de archivo

**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

30

## 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;
```

FDD - Clase 1

31

## 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;
```

```
procedure ingresar(var arch_log:archivo);
var
  num:integer;
begin
  Assign(arch_log,'hola'); //Asigno en memoria el archivo con el nombre hola
  rewrite(arch_log); //creo el archivo
  writeln ('Ingrese numeros');readln (num);
  while (num <> 0) do
    begin
      write(arch_log,num); //escribo en el archivo y el puntero del mismo avanza solo.
      writeln ('Ingrese numero');readln (num);
    end;
  close(arch_log); //cierro indicando que termine de escribirlo.
end;
procedure mostrar(var arch_log:archivo);
var
  num:integer;
begin
  reset (arch_log); //vuelve a abrirlo ya sea para escribirlo o solo mostrarlo
  while not eof (arch_log) do //el puntero se mueve solo desde el principio EOF= END OF FILE.
    begin
      read(arch_log,num); //si o si en una variable,escribo lo que esta en el archivo en una variable
      writeln (num); //imprimo
    end;
  close(arch_log); //cierro el archivo.
end;
```