



# FUNDAMENTOS DE ORGANIZACION DE DATOS

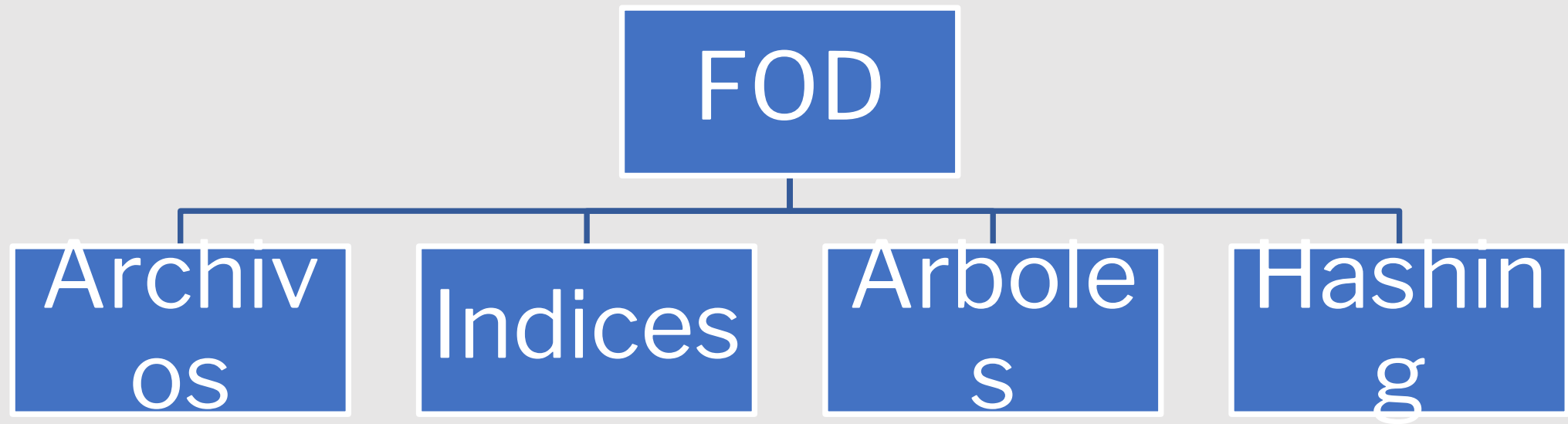
Prof. Titular Rodolfo Bertone

Prof. Asociado Pablo Thomas

Prof. Adjunto Luciano Marrero

Facultad de Informatica UNLP

# La materia



# 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

Que  
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.

# 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 <b>registros</b> guardados en <b>almacenamiento secundario</b>
--------------	---

---

	Colección de <b>datos</b> almacenados en <b>dispositivos secundarios de memoria</b>
--	---

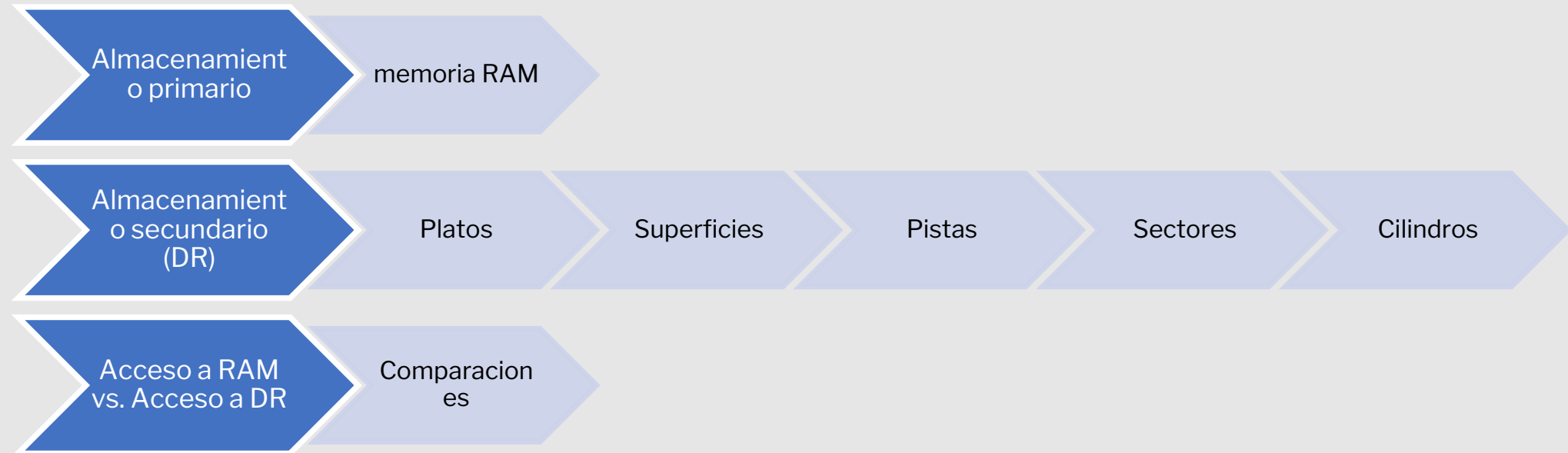
---

	Colección de <b>registros</b> que abarcan entidades con un aspecto común y originadas para algún propósito particular
--	---

---



# Repaso de conceptos de Hardware



# Como se organiza un archivo?

## Secuencia de bytes

- Archivos de texto
- Se leen o recuperan caracteres.
- No hay formato previo.
- Una palabra se termina por un conjunto de caracteres que termina con blanco. Pero eso es una “convención”

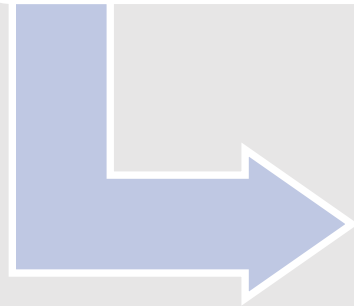
## Conjunto de registros

- Hay estructura.
- Los registros pueden estar conformados por campos

# Archivos organizacion

Registro

- Conjunto de campos agrupados que definen un elemento del archivo



Campo

- Unidad más pequeña, lógicamente significativa de un archivo

# Archivos



Definen  
de que  
tipo de  
archivo  
se trata



# Formas de acceso a archivos

## **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

# Tipos de archivos

## ***Serie***

- cada registro es accesible solo luego de procesar su antecesor, simples de acceder

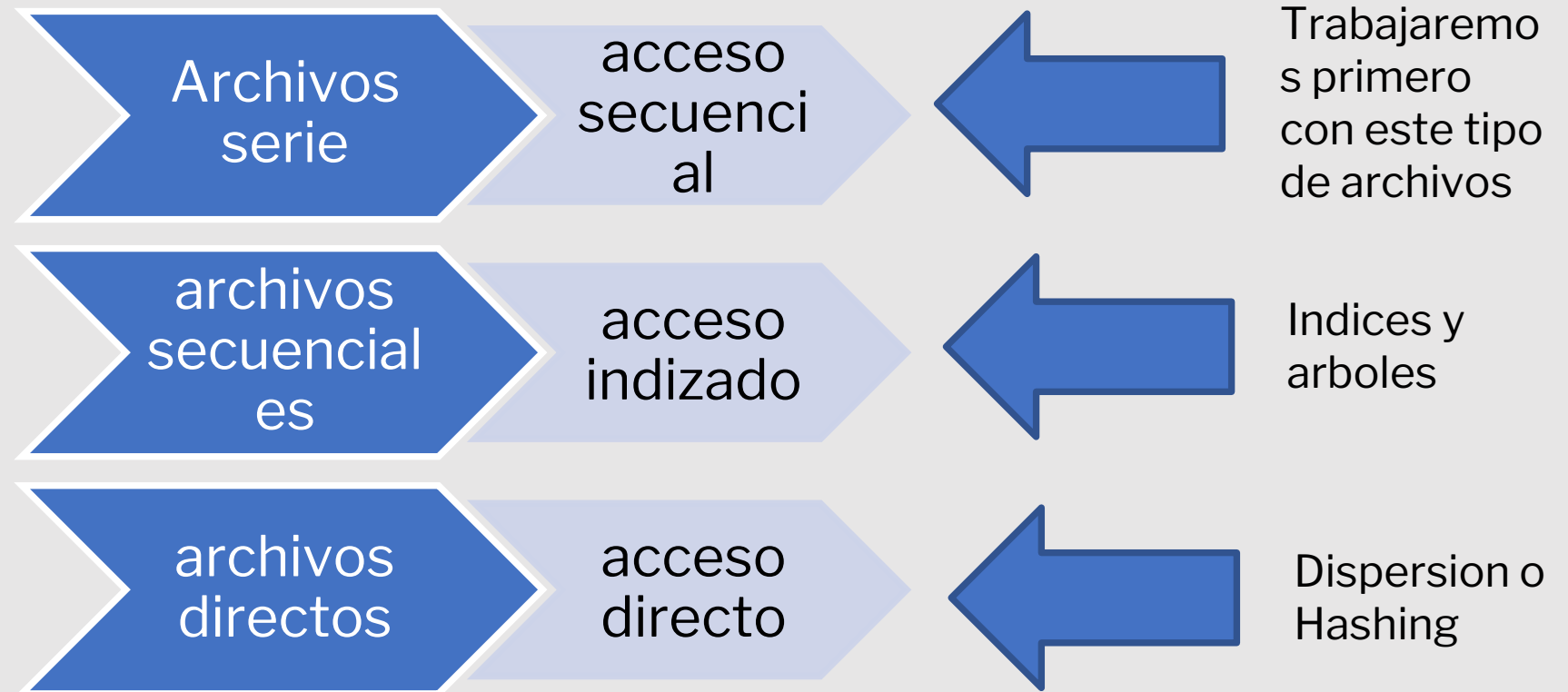
## ***Secuencial*** los registros

- son accesibles en orden de alguna clave

## ***Directo***

- se accede al registro deseado

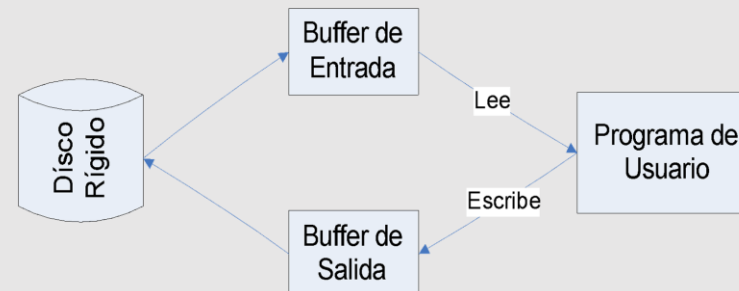
# Resumiendo



# Algunos conceptos

## buffer

- 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? Buffer de entrada y Buffer de salida

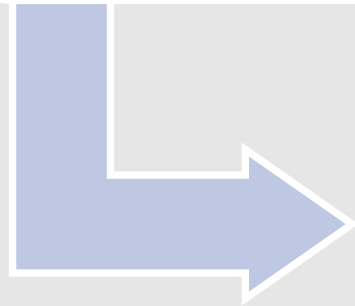




# Archivos en los lenguajes de programación

Archivos  
Físicos

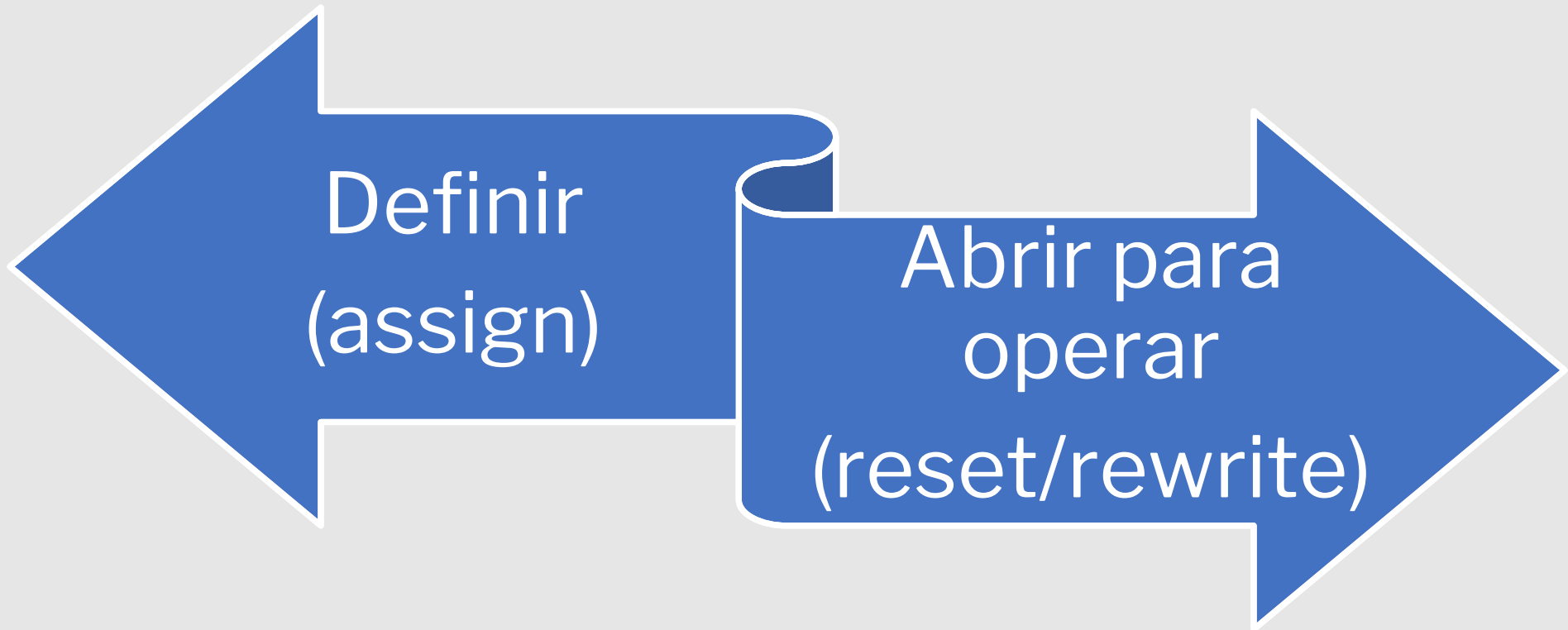
- como aparecen en el disco rígido
- a cargo del sistema operativo



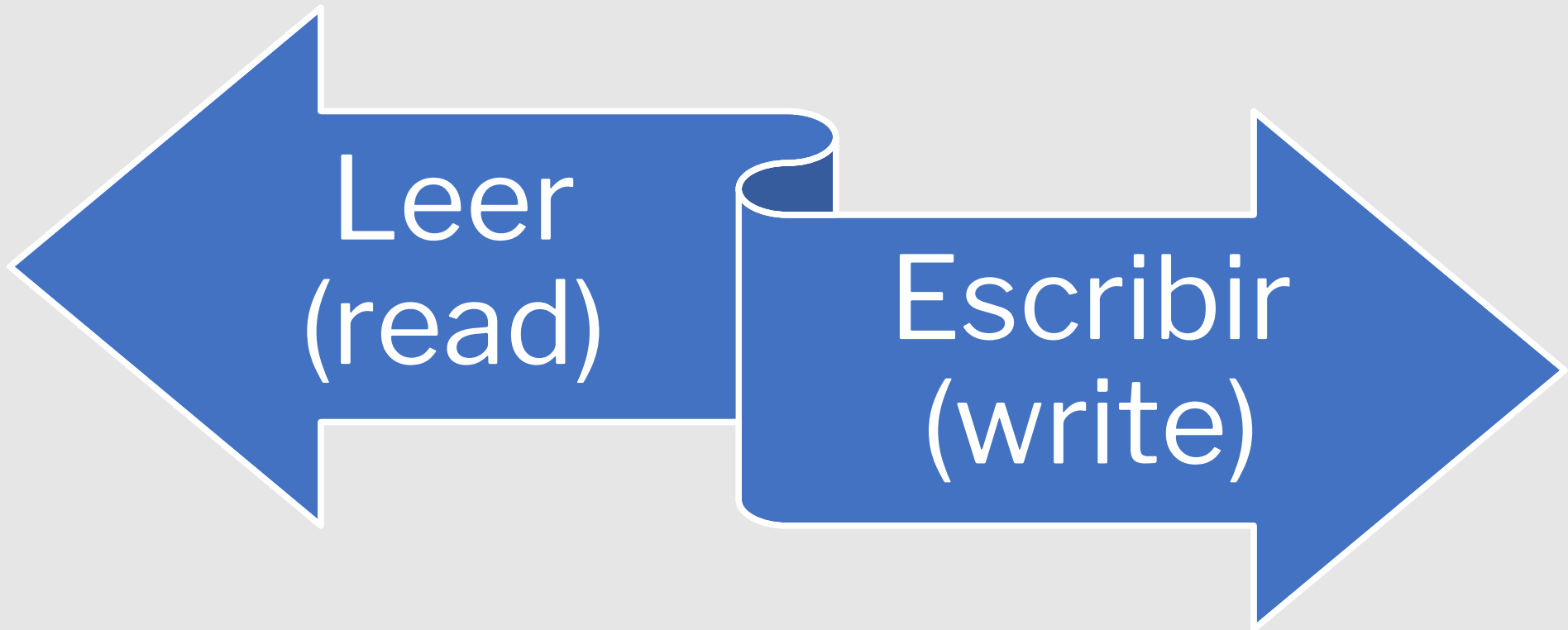
Archivos  
lógicos

- como se manipulan desde el lenguaje de programación

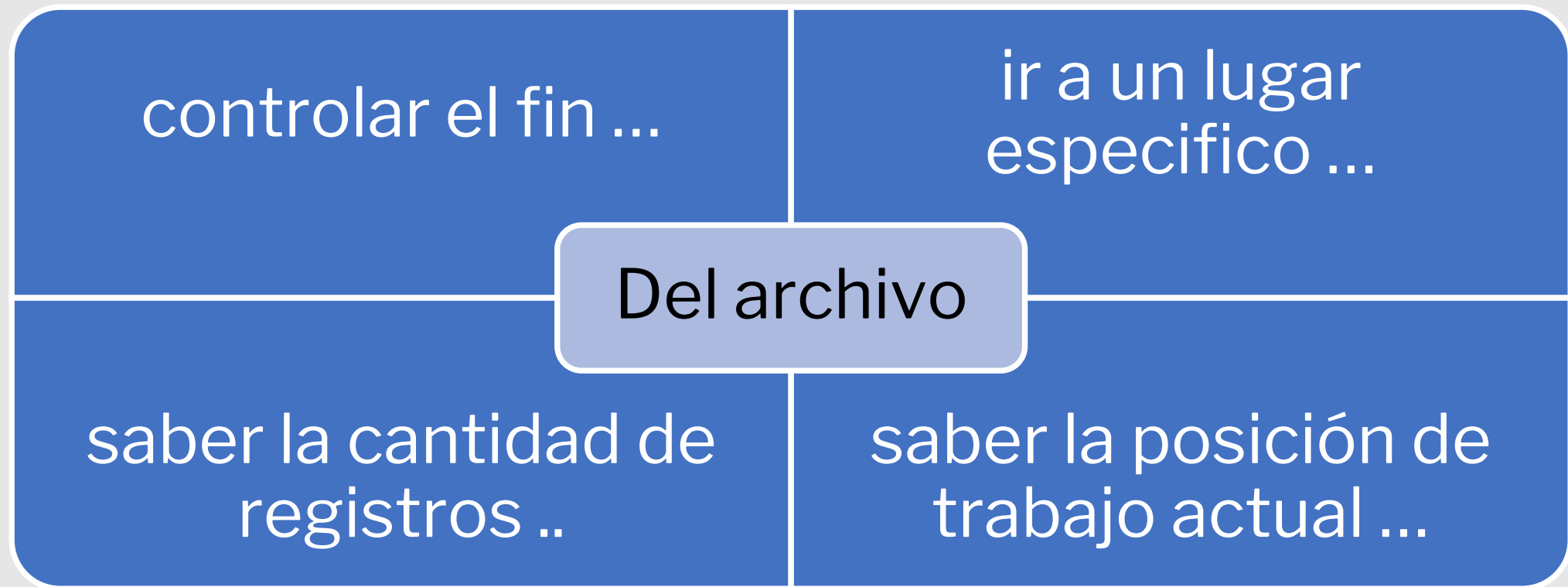
# Archivos: Operaciones iniciales



# Archivos: Operaciones básicas



# Archivos: operaciones adicionales



# Archivos: Definir/declarar (notación Pascal)

## ■ Variable

- *Var archivo: file of Tipo\_de\_dato;*

## ■ Tipo

- *Type archivo: file of Tipo\_de\_dato;*
- *Var arch: archivo*

# Archivos definir/declarar

## ■ 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 definir/declarar

## ■ 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 ejemplo

```
...
type emple = record
    nombre:string[20];
    dirección: string[30];
    edad:integer;
end;
numero = file of integer;
empleado = file of emple;
Var arch_num: numero;
    arch_emp: empleado;

...
begin
    ...
    assign (arch_num, 'pepe.dat');
    assign (arch_emp, 'pipo.dat');
    ...
End.
```

# Archivos operar

```
Rewrite (nombre_logico);
```

- De solo escritura (creación)

```
Reset (nombre_logico);
```

- Lectura Escritura (apertura)

```
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.

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



# 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 el DR

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

# Program **Generar\_Archivo;**

```
type archivo = file of integer;
var arc_logico: archivo;
    nro: integer;
    arc_fisico: string[12];
begin
    write( 'Ingrese el nombre del archivo:' );
    read( arc_fisico );
    assign( arc_logico, arc_fisico );
    rewrite( arc_logico );
    read( nro );
    while nro <> 0 do begin
        write( arc_logico, nro );
        read( nro );
    end;
    close( arc_logico );
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 mostrar el archivo del ej anterior

```
Procedure Recorrido (var arc_logico: archivo );  
    var  nro: integer;  
  
    begin  
        reset( arc_logico );  
        while not eof( arc_logico) do begin  
            read( arc_logico, nro );  
            write( nro );  
        end;  
        close( arc_logico );  
    end;
```

# Archivos archivo

## Ej 3 Modificación un

- 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 tipos en el programa ppal

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

```
Procedure actualizar (Var Emp:empleados);
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;
```