



Facultad de Ciencias
de la **Administración**

Licenciatura en Sistemas

Algoritmos y Programación

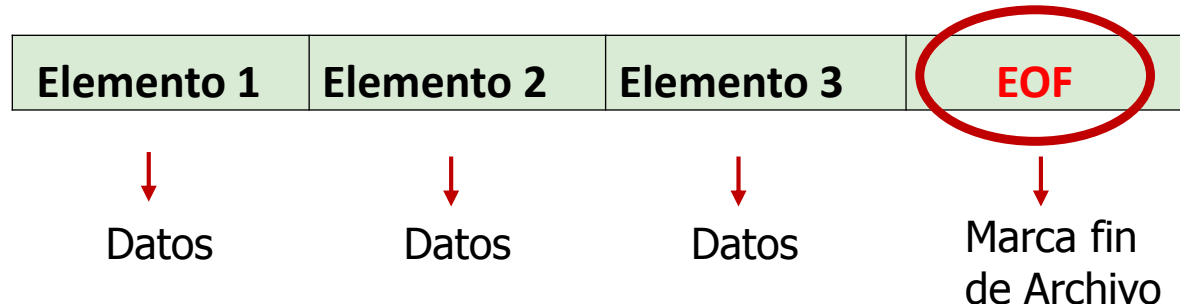
Estructura de Datos
Archivos

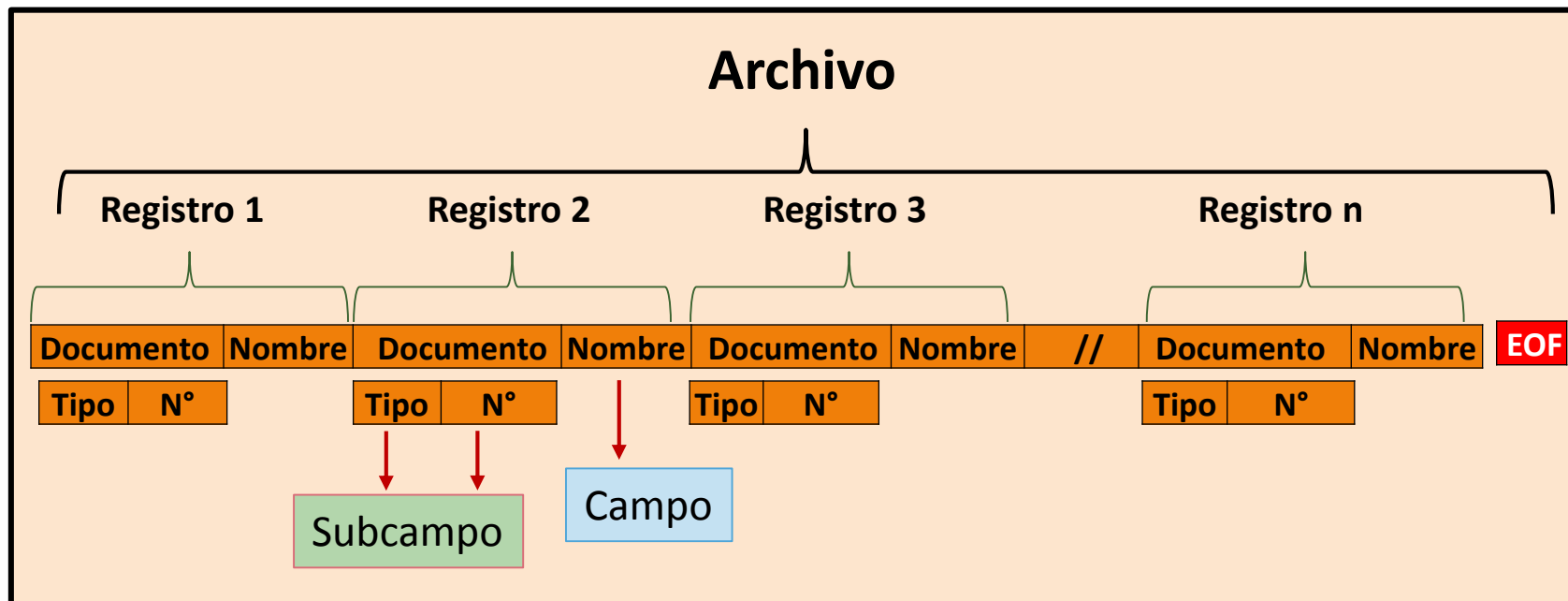
Archivos

Un archivo o fichero (file) es una estructura de datos que reside en memoria secundaria.

Consiste en un conjunto de informaciones estructuradas en unidades de acceso denominadas **registros**, todos del mismo tipo y en número indeterminado.

Es una secuencia de elementos que pertenecen al mismo tipo ó estructura. Es decir, puede ser una secuencia de caracteres, números ó registros, por lo que su representación lógica puede hacerse como una secuencia de módulos de igual tamaño.





Características de los Archivos

- ☐ Residen en soportes de información externos
- ☐ Son independientes respecto de los programas
- ☐ Permanencia de las informaciones almacenadas
- ☐ Gran capacidad de almacenamiento
- ☐ Posibilidad de variar la cantidad de elementos durante la ejecución.

Clasificación de los Archivos

I. En función del contenido:

- a) Archivo de **datos**
- b) Archivo de **programa**
- c) Archivo de **texto**

II. En función del tipos de acceso:

a) Acceso secuencial

Los registros se graban uno a continuación del otro. De la misma manera se leen. Es decir que para leer el registro 'n' es necesario leer los 'n - 1' registros anteriores.

b) Acceso directo

Permiten el acceso directo a un registro.

Sintaxis de las operaciones fundamentales

1. Apertura y cierre:

ABRIR <nombre de archivo>

CERRAR <nombre de archivo>

2. Lectura:

LEER <nombre de Archivo>

3. Grabación:

GRABAR <nombre de Archivo>

4. Re-Grabación :

REGRABAR <nombre de Archivo>

5. EOF(nombre de archivo):

retorna TRUE si se llegó al final de un archivo y FALSE en caso contrario.



Transferir un registro del disco a la memoria RAM.

Transferir un registro de la memoria RAM al disco.

Transferir un registro de la memoria RAM al disco.

Pero... ¿ Siempre se transfieren los datos de esta manera ?

Tipos de registros.

➤ Registro **lógico**:

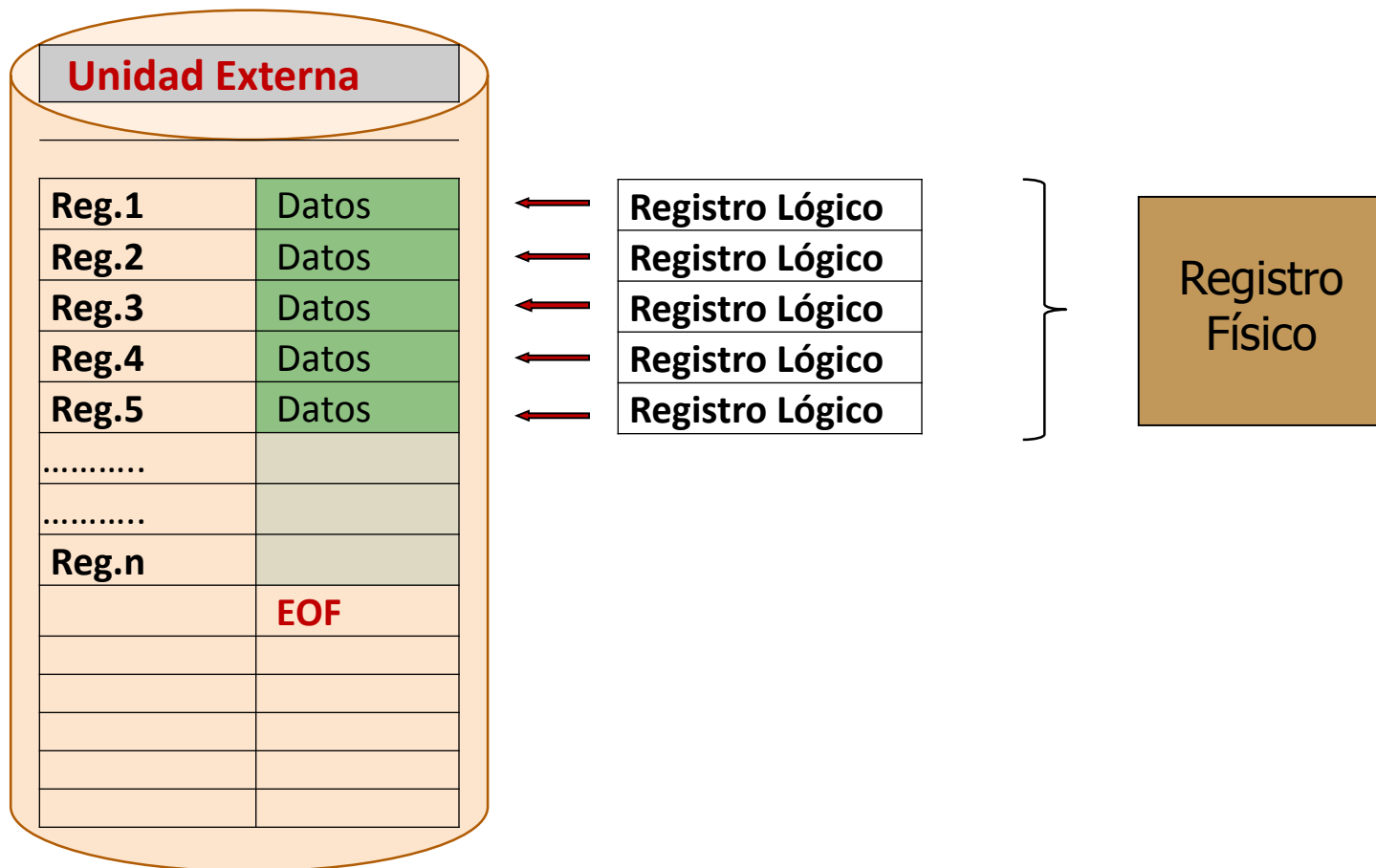
- Cada uno de los componentes del archivo, conteniendo el conjunto de informaciones que se tratan de manera unitaria.
- Está constituido por uno o más elementos denominados campos, que pueden ser de diferentes tipos y que a su vez pueden estar compuestos por subcampos.

➤ Registro **físico**: o *bloque*

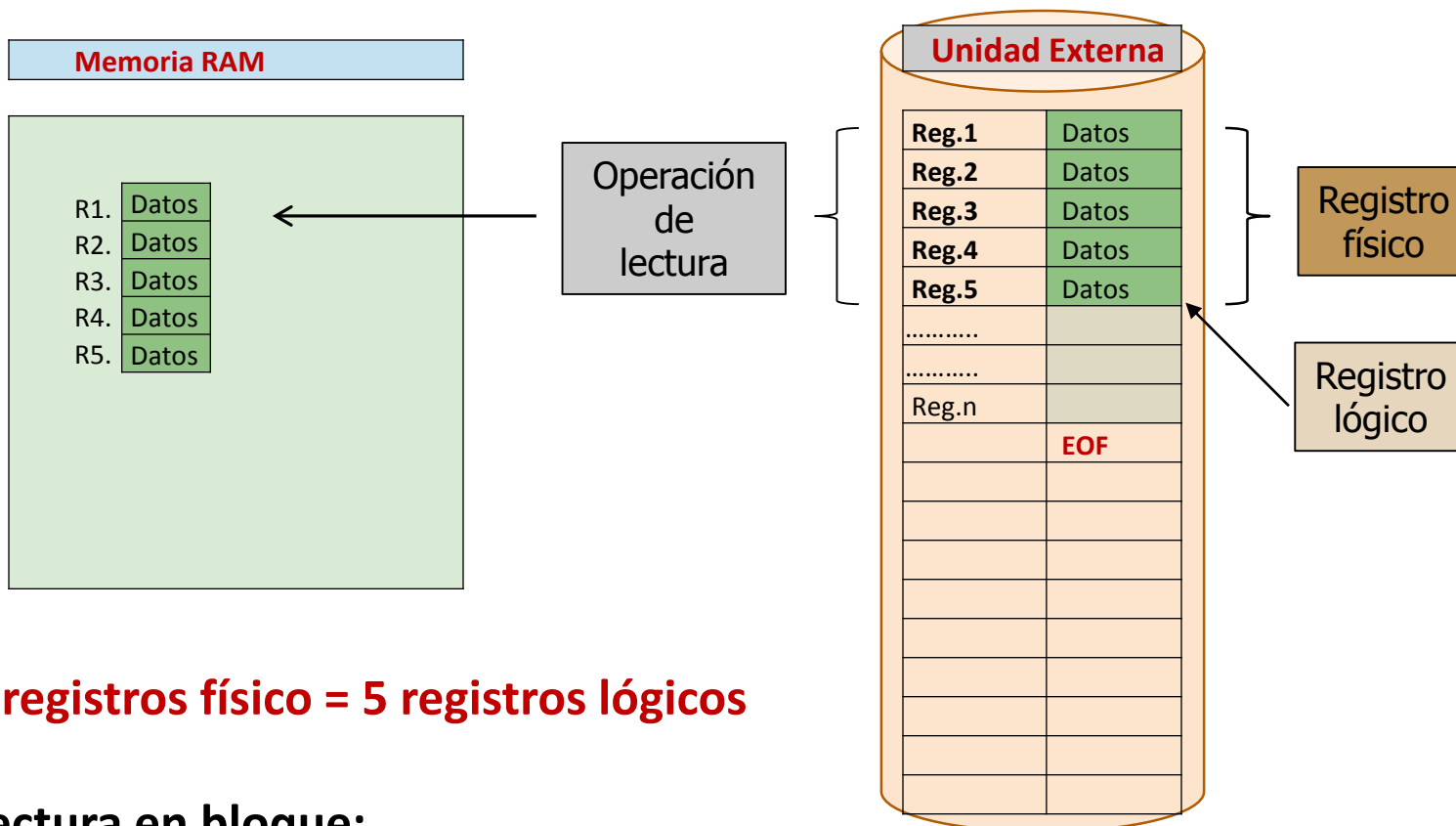
- Corresponde a la cantidad de información que se transfiere en cada operación de acceso (lectura o escritura).
- Contiene uno o mas registros lógicos.

Archivos

1 registro físico = 5 registros lógicos



Archivos



1 registros físico = 5 registros lógicos

Lectura en bloque:

Se leen 5 registros lógicos juntos y se los “copian” en memoria.

¿ Entonces... Cómo trabajamos en nuestros programas ?



¡ Nosotros trabajaremos
a nivel “lógico” !

Es decir: Con las primitivas de transferencias (leer, grabar, regrabar) se indicará que se está transfiriendo (ya sea de memoria a dispositivo o viceversa) **UN** solo registro lógico (aunque sepamos que físicamente se realizan transferencias en bloque).

Se desea crear un archivo el que se guarden el Nro.de Cuenta, Apellido y nombres, y saldo que adeuda un grupo de clientes de una empresa.

1. Se define el registro y el archivo de tipo registro:

TYPE

R-CLI = RECORD

Cuenta : Integer;

Nombre : String;

Saldo : Real

END;

2. El siguiente paso es declarar las variables del tipo archivo y tipo registro

VAR

CLI : file of R-CLI;

**No es necesario declarar
la cantidad de registros**

3. Se le asigna un nombre real para guardarlo en el disco, mediante una primitiva.

Assign (CLI, 'Clientes.dat');

Caso 1.

Generar un archivo con los datos de los clientes de una empresa, siguiendo el ejemplo anterior.

Program Clientes

Type

R-CLI = record
Cuenta : Integer;
Nombre: string;
Saldo : real
end;

Var

CLI : file of R-CLI;
NCta : integer;
Esta : boolean;

Procedimiento Ingreso

Inicio

Assign (CLI, "Cliente.dat")

Abrir (CLI)

Ingresar CLI.Cuenta

Mientras CLI.Cuenta <> 0

Ingresar CLI.Nombre, CLI.Saldo

Grabar (CLI)

Ingresar CLI.Cuenta

FinMientras

Cerrar (CLI)

FIN

Fin Procedimiento Ingreso

Caso 2:

Se desea imprimir el contenido del archivo generado en el ejemplo anterior:

Procedimiento Listado

Inicio

Assign (CLI, "Cliente.dat ")

Abrir (CLI)

Leer (CLI)

Mientras NO EOF (CLI)

Imprimir CLI.Cuenta, CLI.Nombre, CLI.Saldo

Leer (CLI)

Finmientras

Cerrar (CLI)

FIN

Fin Procedimiento Listado

Procedimiento Consulta INICIO

Assign (CLI, "Cliente.dat ")

Ingresar Ncta

Mientras NCTa <> 0

Abrir (CLI)

 ESTA := False

Leer (CLI)

Mientras ESTA = False **and** **NO EOF** <CLI>

 Si NCTa = CLI.Cuenta

 Mostrar CLI.Nombre, CLI.Saldo

 ESTA := True

 Sino

Leer (CLI)

 FinSi

FinMientras

Si ESTA = False

 Mostrar "No existe Cuenta"

FinSi

Cerrar (CLI)

 Ingresar NCTa

FinMientras

FIN

Fin Procedimiento Consulta.

Caso 3:

Se desea mostrar los datos de un empleado del archivo generado en el ejemplo anterior, para lo cual el operador ingresa el N° de Cuenta.

INICIO

Ingresar TNCta

Mientras TNCta <> 0

Abrir (CLI)

ESTA := False

Leer (CLI)

Mientras ESTA = False **and** NO EOF (CLI)

Si TNCta = CLI.Cuenta

Ingresar TCOD, Timp

Si TCOD = 1

CLI.Saldo := CLI.Saldo + Timp

Sino

CLI.Saldo := CLI.Saldo - Timp

FinSi

Regrabar (CLI)

ESTA := True

Sino

Leer (CLI)

FinSi

FinMientras

Si ESTA = False

Mostrar "No existe Cuenta"

FinSi

Cerrar (CLI)

Ingresar TNCta

FinMientras

FIN

Archivos



Caso 4

Se desea actualizar los saldos de los clientes.

Por cada operación ingresan por teclado:

- **TNCta** - Nro.de Cuenta
- **TCOD** - Código de Operación
[1=Compra, 2=Pago]
- **Timp** - Importe



Facultad de Ciencias
de la **Administración**

Algoritmos y Programación

Estructura de Datos
Archivos Secuenciales



**FIN DE LA
CLASE**