

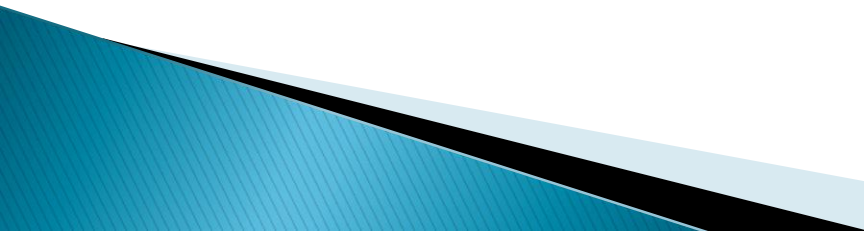
# Computación I. CI-2125

## Tema VIII

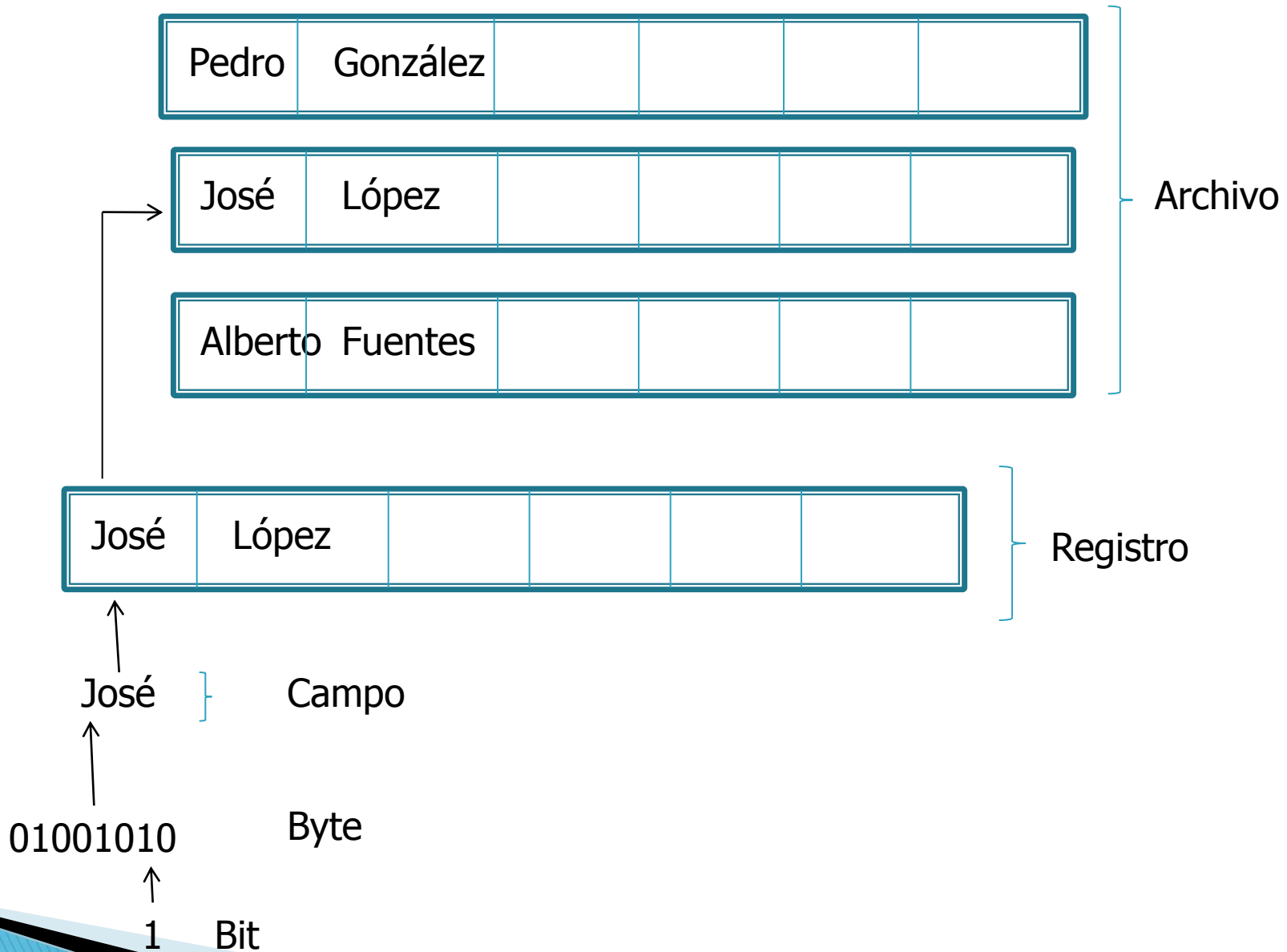
Prof. Mireya Morales



# CONTENIDO

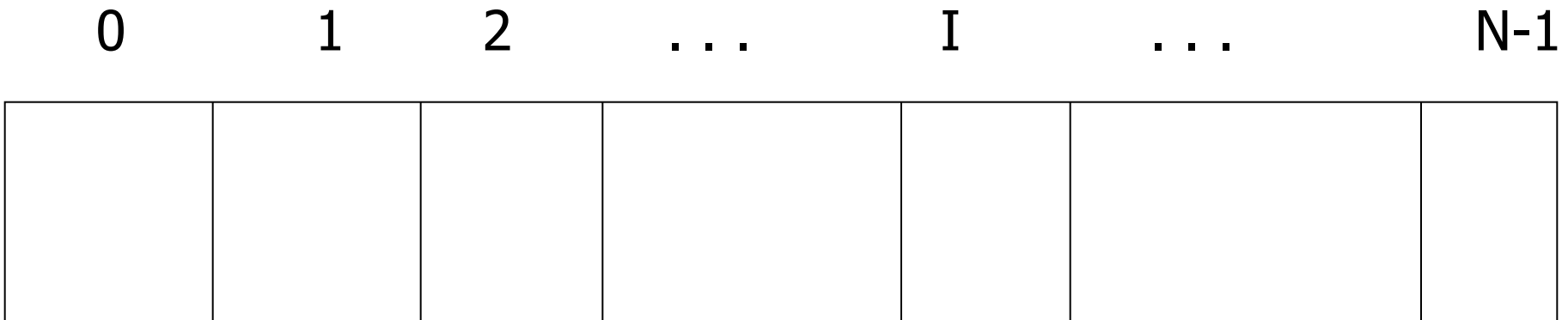
- ▶ *¿Por qué el uso de archivos?*
  - ▶ *Jerarquía de datos*
  - ▶ *Archivos y Flujos*
  - ▶ *Secuencias*
  - ▶ *Funciones del sistema de archivo*
  - ▶ *Apuntador a un archivo*
  - ▶ *Apertura de un archivo*
  - ▶ *Valores válidos de modo*
  - ▶ *Ejemplo de apertura de archivo*
- 

# *Jerarquía de datos*



# Archivos y Flujos

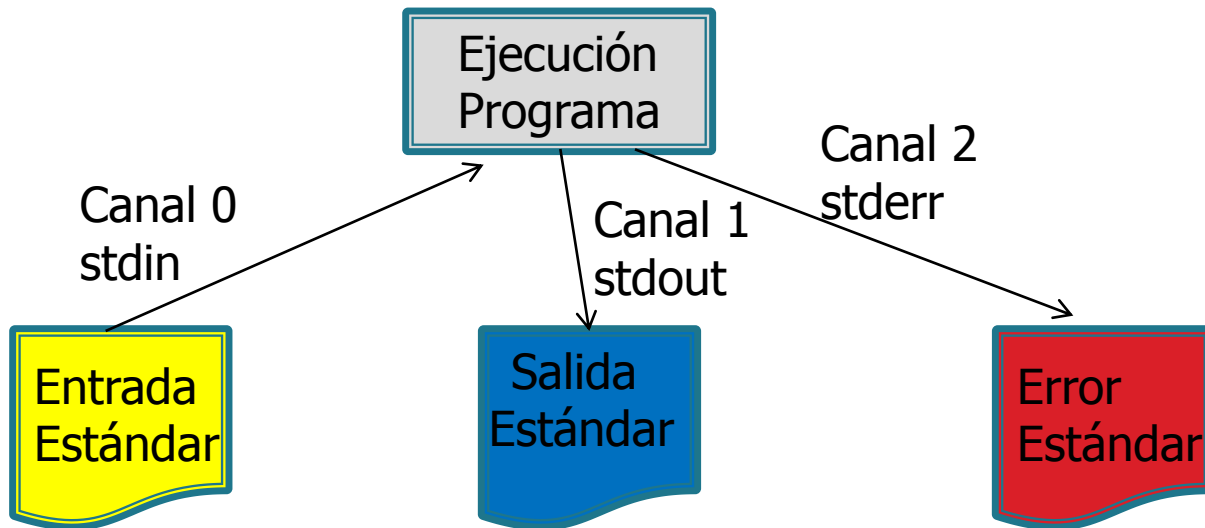
- En C un archivo es un concepto lógico que se aplica a cualquier cosa, desde un archivo de disco, a un terminal o a una impresora.
- El lenguaje C ve cada uno de los archivos simplemente como un flujo secuencial de bytes



↓  
Marcador de fin de archivo

# Archivos y Flujos

- Cuando se ejecuta un programa, automáticamente se abren tres archivos y sus flujos asociados



# Secuencias

- ▶ El sistema de E/S de C proporciona un nivel de abstracción entre el programador y el dispositivo que se está usando. A esta abstracción se le llama secuencia y al dispositivo real archivo.
- ▶ Se asocia una secuencia con un archivo específico realizando una operación de apertura.
- ▶ Cada secuencia que se asocia con un archivo, tiene una estructura de control de tipo FILE. Esta estructura se define en `stdio.h`.

# Funciones del sistema de archivos

- ▶ `fopen()`. Abre una secuencia
- ▶ `fclose()`. Cierra una secuencia
- ▶ `putc()`. Escribe un carácter en la secuencia
- ▶ `fputc()`. Igual que `putc()`
- ▶ `getc()`. Lee un carácter de una secuencia
- ▶ `fgetc()`. Igual que `getc()`
- ▶ `fseek()`. Busca en una secuencia el byte especificado.
- ▶ `fprintf()`. Es a una secuencia lo que `printf()` es a la salida estándar (pantalla)

# Funciones del sistema de archivos

- ▶ `fscanf()`. Es a una secuencia lo que `scanf()` es a la entrada estándar (teclado)
- ▶ `feof()`. Devuelve cierto si ha llegado al fin del archivo.
- ▶ `ferror()`. Devuelve cierto si se ha producido un error
- ▶ `rewind()`. Inicializa el indicador de posición al principio de este.
- ▶ `remove()`. Elimina un archivo
- ▶ `fflush()`. Vacía un archivo



# Apuntador a un archivo

- ▶ Es un apuntador a una información que define varias cosas sobre el archivo, incluyendo el nombre, el estado y la posición actual.
- ▶ Un apuntador a archivo es una variable del tipo apuntador al tipo FILE.
- ▶ Un programa utiliza apuntadores a archivos para leer o escribir en los mismos: Ejemplo:

```
FILE *ap;
```

# Apertura de un Archivo

- ▶ La función `fopen()` tiene 2 propósitos:
  - Abre una secuencia para que pueda ser utilizada y la asocia a un archivo.
  - Devuelve el apuntador al archivo asociado con ese archivo.

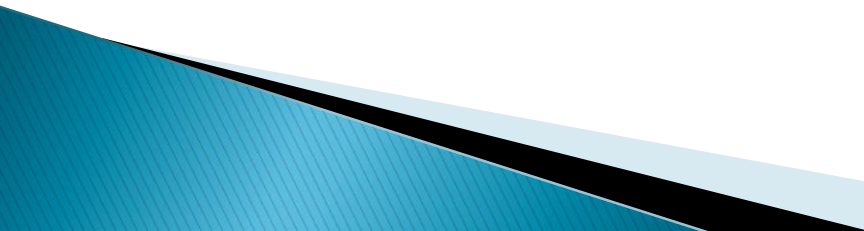
El prototipo es el siguiente:

```
FILE *fopen(char *nombre_archivo, char  
*modo);
```

# Valores válidos de modo

- ▶ “r” abre un archivo de texto para lectura
- ▶ “w” crea un archivo de texto para escritura
- ▶ “a” abre un archivo de texto para añadir
- ▶ “rb” abre un archivo binario para lectura
- ▶ “wb” crea un archivo binario para escritura
- ▶ “ab” abre un archivo binario para añadir
- ▶ “r+” abre un archivo de texto para lectura/escritura
- ▶ “w+” crea un archivo de texto para lectura/escritura

# Valores válidos de modo

- ▶ “a+” abre o crea un archivo de texto para lectura/escritura
  - ▶ “r + b” abre un archivo binario para lectura/escritura
  - ▶ “w + b” crea un archivo binario para lectura/escritura
  - ▶ “a + b” añadir en un archivo binario en modo lectura/escritura
- 

# Ejemplo Apertura de archivo

```
FILE *apf;
```

```
if ((apf = fopen("prueba","w"))==NULL)  
{  
    printf("El archivo no se puede abrir\n");  
    exit(1);  
}
```

# Ejemplo Apertura de archivo

