

# Laboratorio de Computación II

Archivos

# Archivos

Conjunto de bytes que son almacenados en un dispositivo con un nombre identificador.



Archivos de sistema



Archivos ejecutables



Archivos de datos

# Archivos de datos

Datos con un formato y/o estructura específica que son interpretados por programas para poder crearlos, visualizarlos, editarlos y/o borrarlos.



Archivos de datos



Planilla de cálculo



Código fuente



Archivo PDF



Archivo de clientes

# Archivos de datos

Un registro de información de una determinada entidad debe cumplir una serie de reglas para poder ser almacenado en un archivo



Archivo de clientes

- Registros deben tener longitud fija.
- Registros deben ser identificados por un valor único e irrepetible.
- El archivo, en consecuencia al ítem anterior, no debiera admitir registros duplicados.



Alumno objAlumno;

2000Angel Simon 38



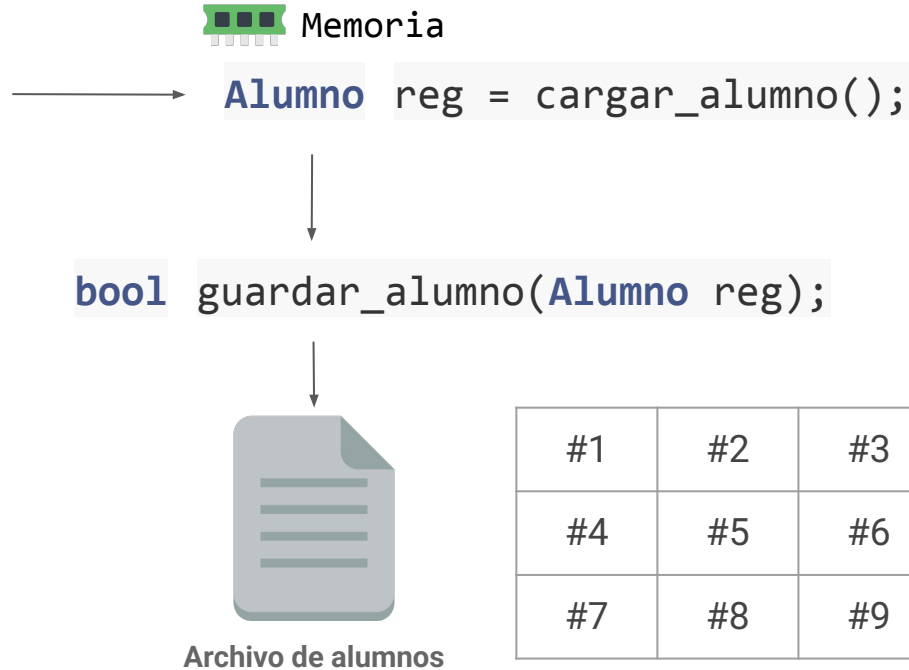
Archivo alumnos.dat

2000Brian Lara 323000Maxi Wenner 304000Daniel Kloster 22



# Archivos de datos - Escritura

```
class Alumno{  
    private:  
    int legajo;  
    char apellido[50];  
    char nombres[50];  
    int edad;  
};
```



# Archivos de datos - Lectura



Archivo de alumnos

ID	Apellido	Nombre	Edad
100	Santos	Mario	40
200	Medina	Gabriel	48
300	Ravenna	Emilio	44
400	Lamponne	Pablo	47

**Alumno** leer\_alumno(**int** nroRegistro);


Memoria




200	Medina	Gabriel	48
-----	--------	---------	----




## Pero...



Las funciones de archivos no admiten trabajar con "números de registros".



Las funciones de archivos trabajan con cantidad de bytes a escribir o leer.



# Lectura desde archivo en C

```
int fread( &reg, sizeof (Alumno), 1, p)
```

Lee del inicio del archivo apuntado por **p**, una cantidad de 120 bytes (el tamaño de un registro de Alumno) y los almacena en la variable **reg** de tipo Alumno.

Puntero FILE. Tipo de dato que "apuntará" a un archivo en disco

Cantidad registros a leer a la vez. Generalmente será de a uno por vez.

Cantidad de bytes de cada registro a leer. Hipotéticamente 120 bytes por registro.

Registro de tipo Alumno donde se va a guardar la información proveniente del archivo.

# Escritura desde archivo en C

```
int fwrite( &reg, sizeof (Alumno), 1, p)
```

Escribe al final del archivo apuntado por **p**, un registro de tamaño 120 bytes (el tamaño de un registro de Alumno) proveniente desde la dirección de **reg**.

Puntero FILE. Tipo de dato que "apuntará" a un archivo en disco

Cantidad registros a escribir a la vez. Generalmente será de a uno por vez.

Cantidad de bytes de cada registro a escribir. Hipotéticamente 120 bytes por registro.

Registro de tipo Alumno de cual proviene la información a guardar en el archivo.

# Puntero FILE

Un tipo de dato que nos permitirá "conectar" nuestro programa a un archivo en el dispositivo que indiquemos.

```
FILE *p;  
p = fopen("archivo.dat", "rb");
```

*/\* Todas las acciones a realizar con el archivo \*/*

```
fclose(p);
```

Ruta relativa  
"datos/archivo.dat"  
"alumnos.dat"



C:/MiApp/app.exe

E:/App/App.exe

Ruta absoluta  
"C:/datos/archivo.dat"  
"E:/alumnos.dat"



# fopen

Una función que nos permite conectar un archivo de datos a un puntero FILE. Además, le indicamos el modo de apertura.

```
p = fopen("archivo.dat", "rb");
```

Modo	Nombre	Descripción
<u>wb</u>	write-binary	Destruye el archivo a 0 bytes y lo abre como escritura. Si no existe el archivo, lo crea.
<u>rb</u>	<u>read-binary</u>	Abre el archivo como lectura desde el byte 0. No admite escritura.
<u>ab</u>	append-binary	Abre el archivo desde el final y lo abre como escritura. No admite modificar lo existente, sólo agregar. Si no existe el archivo, lo crea.
rb+	read-binary plus overwrite	Abre el archivo desde el byte 0 y lo abre como escritura. Admite modificar registros existentes.

# fclose

Una función que nos permite liberar la conexión de un archivo de datos a un puntero FILE.

```
fclose(p);
```

- Es clave porque no se puede abrir el mismo archivo más de una vez simultáneamente.

- Confirma los datos en aperturas de escritura.

# Ejemplo

[Link](#)

# ABML Postulantes

Postulante

- id
- apellidos
- nombres
- edad
- puesto
- xp
- contratado

PostulanteMenu

- PostulanteManager \_manager;
- + mostrar()

PostulanteManager  
- PostulanteArchivo \_archivo;

- + cargarNuevoPostulante
- + listarTodos
- + buscarPostulantes

PostulanteArchivo  
- nombreArchivo

- + leerUno (int pos)
- + leerMuchos (int \*, int cant)
- + guardar (Postulante reg)
- + modificar (Postulante reg, int pos)
- + eliminar (Postulante reg)
- + getCantidadRegistros

PostulanteArchivo arch("postulante.dat")

PostulanteArchivo  
arch2("postulantesContratados.dat")