

ARCHIVOS

Conjunto de información almacenada en un dispositivo externo.

DATOS → RAM (almacenamiento interno)

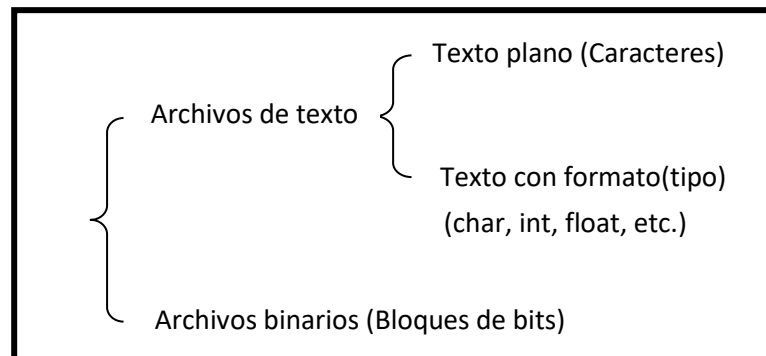
(Velocidad de acceso, Baja capacidad (64 Gb), volátil)

DATOS → ARCHIVO (HD, CD, DVD, BRD, SSD, USB, etc.)

(Velocidad baja de acceso, Alta capacidad(1 Tb),
permanente)

- Tipos de archivos

De manera general se cuenta con 2 tipos de archivos: *Archivos de texto* y *archivos binarios*.



*Manejo de archivos con formato(tipo) en Lenguaje C

El lenguaje C ofrece elementos de programación (Tipos e instrucciones) que permiten implementar el manejo de archivos en los programas.

- Declarar el archivo

FILE *idVarArch [;]

Ejemplo(s): // Como variable

FILE *miArch;

FILE *arch1, *arch2;

```
//Como parámetro por valor  
void manejaArch(FILE *arch, ...);  
//Como parámetro referencia  
void manejarArch(FILE **arch, ...);
```

- (apertura) Abrir el archivo

```
idVarArchivo = fopen(nombreArch, modoApertura);
```

Donde:

Nombre del archivo es el nombre que tiene/tendrá la información en el dispositivo externo. Este nombre puede ser una **cadena directa** o una **variable de cadena**.

El modo de apertura es una cadena que indica la operación a realizar con el archivo. Los modos disponibles para archivos de texto(tipeados) son:

w → (write) Escritura/ creación del archivo. (Llevar datos de la memoria al archivo).

r → (read) Lectura del archivo. (Llevar datos del archivo a la memoria).

a → (append) Agregar datos a un archivo existente.

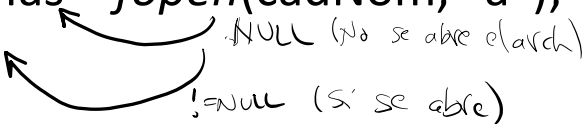
Ejemplo(s):

```
FILE *miArch;  
miArch = fopen("datos.txt", "w");
```

```
FILE *otroArch;  
otroArch = fopen("otro.txt", "r");
```

```
FILE *otroMas;  
char cadNom[]={ "misdatos.txt"};  
char cadModo[]={ "r"};  
otroMas = fopen(cadNom, cadModo);
```

```
FILE *otroArchMas;  
char cadNom[] = { "otro.txt"};  
otroArchMas = fopen(cadNom, "a"); // ~ Inicialización
```



Cada vez que “abrimos” un archivo nos podemos encontrar diversos motivos por los cuales el archivo podría no estar disponible: Disco lleno, sectores dañados, archivo no existente, etc. Debido a esto, cuando llamamos a la función fopen nos contesta con un valor de **NULL** cuando no se puede abrir el archivo y un valor **!= NULL** cuando sí se puede abrir el archivo, este valor es recibido por la variable asignada al llamado, y siempre debemos verificar la apertura del archivo.

*Verificación de la apertura del archivo

```
varArch = fopen("arch.txt", "r"); // Intento  
if(varArch != NULL) //Verificación  
    printf("Archivo disponible, a trabajar\n");  
else  
    printf("Archivo no disponible, a descansar\n");
```

** Manejo del archivo: Escritura /Lectura

*** Escritura: RAM → Disp. Externo

```
fprintf(idArch, "cad. formato", var1, var2, ...);
```

Diagram illustrating the sequential writing process:

- A box labeled "Archivo" contains a sequence of characters: A B C D ... Z.
- Arrows point from the characters A, B, C, and D to the word "escritura" (writing) below the box.
- To the left of the box, the words "puntero de lectura" (reading pointer) and "oscilatoria" (oscillating) are written vertically, with an arrow pointing towards the box.

```
for(c=0; c < n; c++)
    fprintf(arch,"%f ",arr[c]);
```

```
fscanf(idArch, "cad. Fmto", &var(s));
```

```
fscanf(arch, "%d", &edad);
printf("Así que tienes %d años\n", edad);
```

$$\{$$

}

Cadena: Hacer a todos '0'
↑
terminador

Archivo: ABCD...Z<EOF>
End of File

feof(idArch) => Nos notifica si hemos llegado o no al final de los datos en un archivo. (TRUE / FALSE)

Por ejemplo:

```
while( !feof(miArch) ) // Mientras NO se termine el archivo
{
    fscanf(miArch, "%c", &letra);
    printf("Letra = %c\n", letra);
}
```

```
scanf("%d %d %d", &v1, &v2, &v3);
```

El fscanf nos avisa cuantos datos leyó del archivo y cuando no puede leer nos avisa con un 0.

```
res = fscanf(arch, "%c", &letra);
```

Si puede leer el dato, res recibe 1, si no recibe 0.

```
res = fscanf(arch, "%c %d %f", &letra, &edad, &suelo);
```

Si lee los **tres** valores, res será igual a 3

Si lee **letra y edad**, res será igual a 2

Si solo lee **letra** res valdrá 1

Y si no lee **ninguno**, res valdrá 0

Otra forma:

<pre>while(fscanf(miArch, "%c", &letra) > 0) // == 1 o 0 printf("Letra = %c\n", letra);</pre>
--

**** Cerrar el archivo**

```
fclose(idArch);
```

Ejemplo:

```
fclose(miArchivo);
```