

# Archivos en C

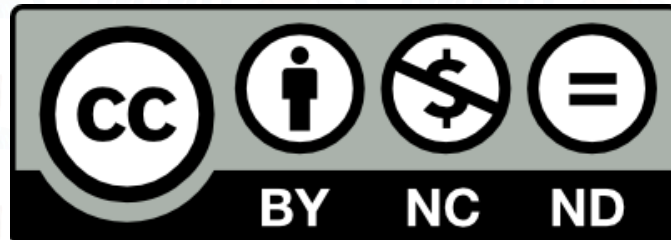
## Charly Cimino

Este documento se encuentra bajo Licencia Creative Commons 4.0 Internacional (CC BY-NC-ND 4.0). Usted es libre para:

- **Compartir** — copiar y redistribuir el material en cualquier medio o formato.

Bajo los siguientes términos:

- **Atribución** — Usted debe darle crédito a esta obra de manera adecuada, proporcionando un enlace a la licencia, e indicando si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo del licenciante.
- **No Comercial** — Usted no puede hacer uso del material con fines comerciales.
- **Sin Derivar** — Si usted mezcla, transforma o crea nuevo material a partir de esta obra, usted no podrá distribuir el material modificado.



# ¿Qué es un archivo?

Un archivo (también llamado fichero) es un conjunto de datos guardado en un medio de almacenamiento permanente (como un disco rígido, pendrive, DVD, cinta magnética, etc).

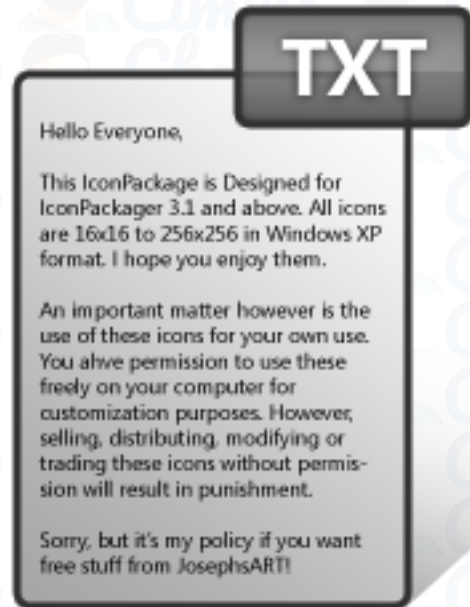
A través de ellos podemos manipular datos aun habiendo reiniciado o apagando la computadora.



# Tipos de archivos

Los datos en un archivo son guardados como bytes (conjuntos de 8 bits)

Según cómo se codifican y qué representan esos bytes, se desprenden dos tipos de archivos:



Archivos de texto



Archivos binarios



# Archivos de texto

## TXT

Hello Everyone,

This IconPackage is Designed for IconPacker 3.1 and above. All icons are 16x16 to 256x256 in Windows XP format. I hope you enjoy them.

An important matter however is the use of these icons for your own use. You have permission to use these freely on your computer for customization purposes. However, selling, distributing, modifying or trading these icons without permission will result in punishment.

Sorry, but it's my policy if you want free stuff from JosephsART!

- Sus bytes se codifican como caracteres que representan texto legible.
- Es fácil detectar errores y manipularlos, dado que son legibles por los humanos.
- Son inseguros, pues se pueden ver y manipular con cualquier editor de texto.
- A la hora de obtener los datos, todo es visto como texto: números, valores de verdad, estructuras, etc.
- Suelen tener la extensión **.txt**

# Archivos binarios

01010101

01011100101001001001010101000  
10101001010010010010101110010  
10010010010101010001010100101  
00100100101011100101001001001  
01010100010101001010010010010  
10111001010010010010101010001  
010100101001001001

01011100101001001001010101000  
10101001010010010010101110010  
10010010010101010001010100101  
00100100101011100101001001001  
0101010001010100101001001001

01011100101001001001010101000  
1010100101001001001

- Sus bytes se codifican como datos personalizados (números, texto, *arrays*, estructuras).
- El más pequeño error corrompe al archivo, siendo difícil de detectar.
- Son más seguros, pues su contenido, al no ser directamente legible por un humano, no se puede procesar si no se conoce la estructura del mismo.
- A la hora de obtener los datos, se mantiene los tipos de datos específicos, tal y como fueron guardados originalmente.
- Suelen tener la extensión **.bin**

# Puntero a archivo


Para poder trabajar con un archivo en C, es necesario declarar una referencia (puntero) hacia él.

```
#include <stdio.h>

int main () {

    FILE* pArchivo; // Declara un puntero a un archivo

    return 0;
}
```



Por ahora, su contenido es “basura”

**FILE** es un tipo de dato (como lo es el **char**, el **int**, etc) que representa un archivo.

[https://www.w3schools.com/c/c\\_files.php](https://www.w3schools.com/c/c_files.php)

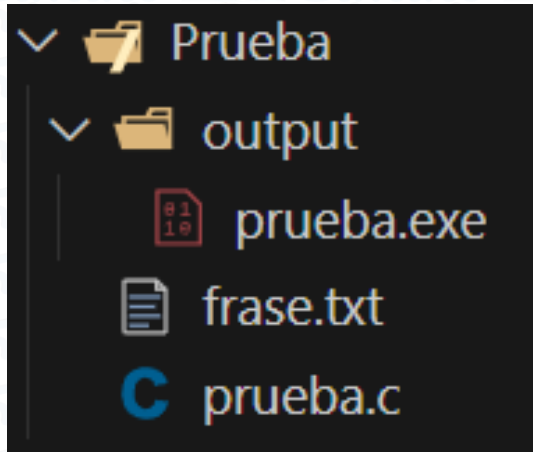
# Abrir un archivo

Antes de manipular un archivo, primero hay que abrirlo.

**pArchivo = fopen(*rutaHaciaElArchivo*, *modo*)**

La función **fopen()** recibe como argumentos la ruta hacia el archivo y el modo de apertura (detallado más adelante).

Retorna un puntero a un archivo. Si este es **NULL**, significa que no fue posible abrirlo.



```
#include <stdio.h>
int main () {
    FILE *pArchivo;
    pArchivo = fopen("../frase.txt", "r");
    if (pArchivo != NULL) {
        printf("ARCHIVO '%s' ABIERTO CORRECTAMENTE\n");
    }
    else {
        printf("NO SE PUDO ABRIR EL ARCHIVO\n");
    }
    printf("El puntero al archivo es %p\n", pArchivo);
    return 0;
}
```

**Atención:** La ruta es con respecto al ejecutable (.exe) que no necesariamente está en la misma carpeta que el código fuente (.c)

Modo de lectura de datos de texto.  
Más adelante se detallan otros...

# Cerrar el archivo

Después de manipular el archivo, **hay que cerrarlo.**

```
#include <stdio.h>
int main()
{
    FILE *pArchivo;
    pArchivo = fopen("../frase.txt", "r");
    if (pArchivo != NULL) {
        printf("ARCHIVO '%s' ABIERTO CORRECTAMENTE\n");
        // Procesar el archivo...
        fclose(pArchivo); // Cierra el archivo
    }
    else {
        printf("NO SE PUDO ABRIR EL ARCHIVO\n");
    }
    return 0;
}
```



Cerrar el archivo tras su uso es una buena práctica, porque asegura:

- Que los cambios se guarden correctamente.
- Permitir el uso del archivo a otros programas.
- Liberar espacio en la memoria RAM.



# Modos de apertura de archivos

`pArchivo = fopen(rutaHaciaELArchivo, modo)`

Modo	Nombre	Descripción
"r"	<code>read</code>	Permite abrir un archivo de texto solo para leerlo. El archivo debe existir previamente.
"w"	<code>write</code>	Permite abrir un archivo de texto para escribir en él. Si el archivo no existe, lo crea y escribe, de lo contrario, escribe "pisando" el contenido anterior.
"a"	<code>append</code>	Permite abrir un archivo de texto para escribir en él. Si el archivo no existe, lo crea y escribe, de lo contrario, escribe a continuación del contenido anterior.
"r+"	<code>read+</code>	Permite abrir un archivo de texto para leerlo o para escribir en él, a continuación del contenido anterior. El archivo debe existir previamente.
"w+"	<code>write+</code>	Permite abrir un archivo de texto para leerlo o para escribir en él. Si el archivo no existe, lo crea y escribe, de lo contrario, escribe "pisando" el contenido anterior.
"a+"	<code>append+</code>	Permite abrir un archivo de texto para leerlo o para escribir en él. Si el archivo no existe, lo crea y escribe, de lo contrario, escribe a continuación del contenido anterior.

# Más contenido pendiente...

# FIN DE LA PRESENTACIÓN

Encontrá más como estas en mi [sitio web](#).