



**UNRN**

Universidad Nacional  
de Río Negro



| unrionegro

# Errores en archivos

**UNRN**

Universidad Nacional  
de Río Negro

**9**

**2023**



---

# Archivos

## #include <stdio.h>

## parte 3

---

# apertura

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

**Si hay problemas  
para abrir el  
archivo**

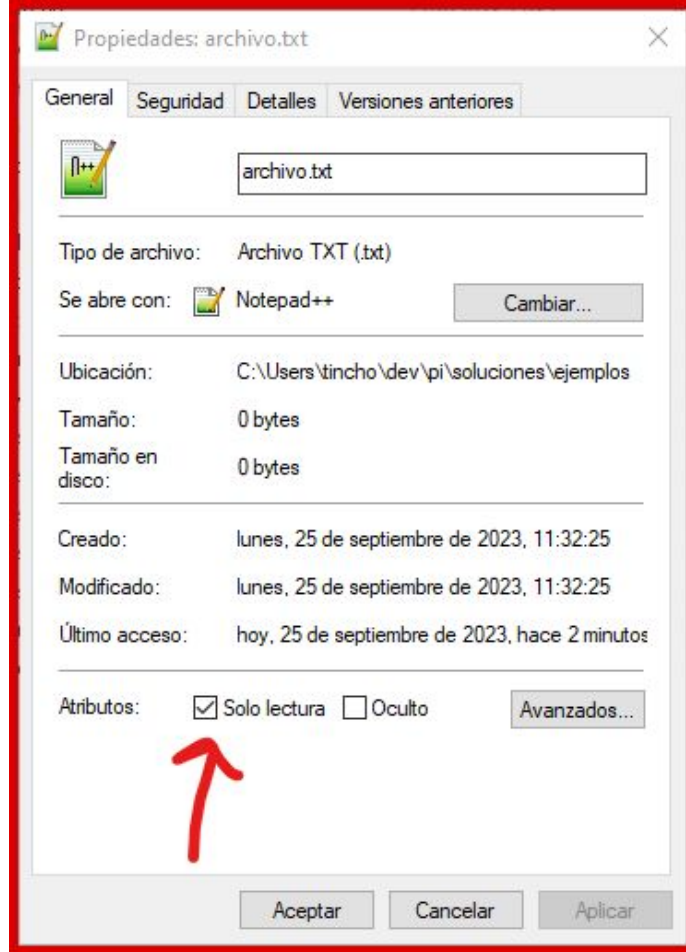
# El puntero apuntará a NULL

**¿Qué problemas  
puede tener la  
apertura?**

# El archivo no existe



# Problemas de permisos (en modo “w”)



**Incluyendo los de seguridad de sistema**

# El archivo ya estaba abierto\*

```
a1 = fopen("archivo.txt", "r");  
a2 = fopen("archivo.txt", "w");
```

**\*en especial si olvidan el `fclose` y necesitan del archivo de nuevo**

# El puntero apuntará a NULL

**Pero, ¿no podemos  
saber más?**

# perro en stdio.h

```
void perro(char* mensaje);
```

*Imprime* por consola (stderr) el mensaje sobre el último error junto a un mensaje propio.

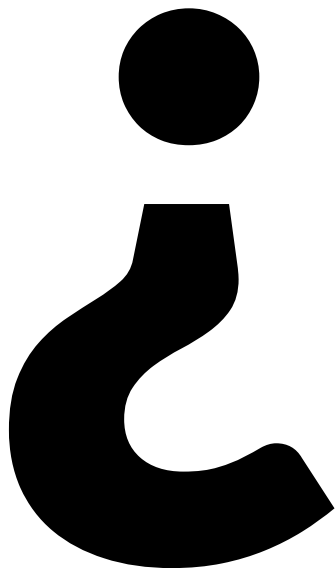
# Archivo inexistente

```
#include <stdio.h>

int main()
{
    FILE *archivo = fopen("no_existi.txt", "r");
    if (archivo == NULL)
    {
        perror("fopen() fallo");
    }
    else
    {
        fclose(archivo);
    }
}
```

# Y veremos por consola

```
fopen() fallo: No such file or directory
```



**Podemos decidir  
como seguir**





# errno en errno.h

Una *variable global* con el código de error producido por la **última llamada a función**.

**1** funciones como  
**printf** puede  
cambiar su valor



**Copien el valor de  
errno antes de  
llamar a otra  
función**

# 2

**el valor *solo* tiene  
sentido si la función  
retorna valores  
como NULL o -1**

# Por lo que se utiliza

```
FILE *archivo = fopen("no_existi.txt", "r");
if (archivo == NULL)
{
    printf("%s\n\n", strerror(errno));
    perror("fopen() fallo");
}
else
{
    // todo OK
}
```

# Algunos valores relevantes de `errno.h`

ENOENT - El archivo no existe

EEXIST - El archivo ya existe

ENAMETOOLONG - Nombre muy largo

EACCES - Error de permisos de acceso



**Todos comienzan  
con E**

**Y nos dan lugar a  
decidir como seguir**



# Por ejemplo

Si el archivo ya existía, pueden agregar un “-copia”



**¿Preguntas?**

**Pero, si  
necesitamos el  
mensaje**

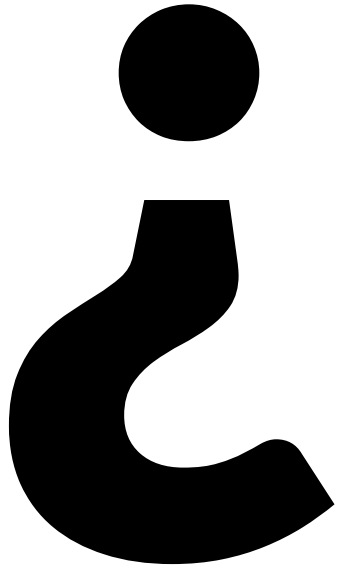
**También lo podemos  
obtener igual que el  
de perro**

# strerror en string.h

```
char* strerror( int errnum );
```

Retorna un puntero a la cadena que describe el error indicado por `errnum`.

Idéntico al que saldría por `perror`.



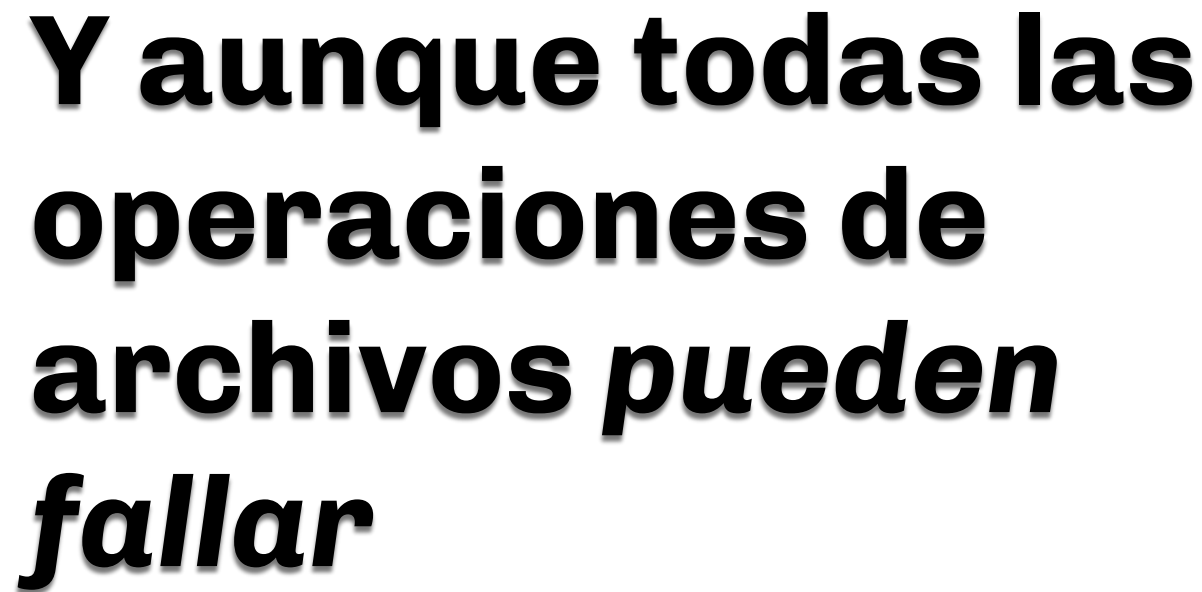
**Que retorna  
fprintf**



# `fprintf` (y `printf`)

La cantidad de caracteres escritos

Con valores negativos si hay problemas



**Y aunque todas las  
operaciones de  
archivos *pueden*  
*fallar***



**atajarlos en fopen es lo  
más importante**

---

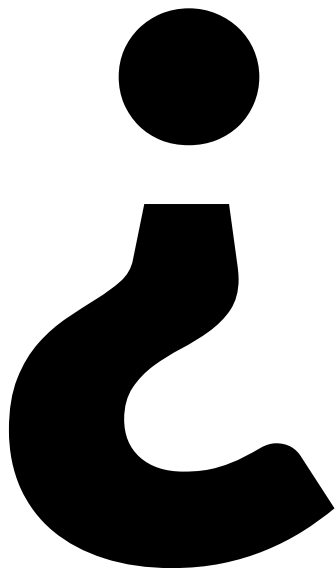
# Otras situaciones de error como para ver

- Abrir o cerrar dos veces un archivo
- Operar con un FILE en NULL
- No cerrar el archivo luego enviar información
- El archivo deja de estar disponible



**¿Preguntas?**

***PLUS  
ULTRA***

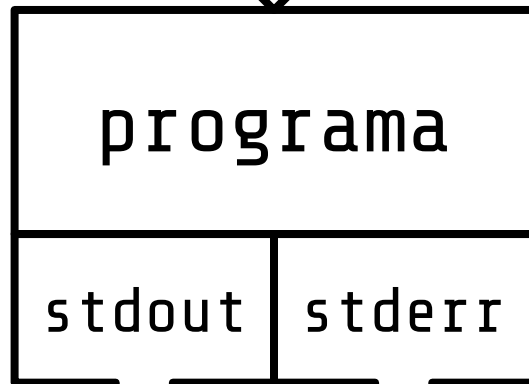


**Qué es eso de  
stderr**



**Otro canal de salida  
como el usado por  
printf (stdout)**

scanf  
gets



printf\*



stdout

stderr



fprintf\*

**Lo normal es que  
ambos esten  
conectados a la  
consola**



**Y permite separar  
los mensajes de  
error de la salida  
normal del  
programa**

**Se usa para los  
programas  
conectados por la  
terminal**

## Para escribir a 'error estándar'

```
fprintf(stderr, "Esto es un error!\n");
```

**Aunque este  
'superpuesto' con  
los printf  
tradicionales**

**\$> ./miprograma  errores.txt**

**Envía solo los errores al archivo  
errores.txt**



**Hay más combinaciones  
y posibilidades, pero  
escapan a este curso**

# **Le darán uso en la materia Sistemas Operativos**

# Lo mismo que printf y scanf

```
fprintf(stdout, "Esto es un printf!\n");  
scanf(stdin, "%d");
```



**Lo interesante es que los  
tres**

---

**son *archivos***

---

**y aplican todos los  
códigos de error**



**¿Preguntas?**

---

# Posibles errores con punteros

**unrn.edu.ar**

**UNRN**

Universidad Nacional  
de **Río Negro**



| **unrionegro**