

**CONTINUACIÓN DE ARCHIVOS**

# **ARCHIVOS BINARIOS**

**SEGUNDA PARTE**

Este tipo de archivos almacenan los datos numéricos con su representación binaria. Pueden ser archivos que contienen instrucciones en lenguaje máquina listas para ser ejecutadas.

Por ejemplo, cuando escribimos un programa en un lenguaje en particular (como en C) tenemos las instrucciones almacenadas en un archivo de texto llamado programa fuente, pero una vez que lo sometemos a un proceso de compilación y ejecución nuestro programa es convertido en un programa ejecutable (en lenguaje máquina), que es directamente entendido por la computadora y se crea un archivo binario.

En este tipo de archivos también se pueden almacenar diferentes tipos de datos incluyendo datos numéricos; sin embargo, cabe destacar que los datos numéricos se graban con su representación binaria (no con su representación ASCII), por tal razón, cuando se despliegan con un editor de textos o por medio de comandos del sistema operativo, aparecen caracteres raros que no se interpretan.

## Funciones de manejo de archivos

fopen() Abre un archivo.

fclose() Cierra un archivo.

fseek() Busca un byte específico de un archivo.

fwrite() Escribe una salida con formato en el archivo.

fread() Lee una entrada con formato desde el archivo.

feof() Devuelve cierto si se llega al final del archivo.

ftell() obtiene el valor actual del indicador de posición del archivo

ferror() Devuelve cierto si se produce un error.

remove() Borra un archivo.

rename() renombra un archivo

fflush() Vacía un archivo.

**SEEK\_SET:** mueve la posición del puntero del archivo al comienzo del archivo

**SEEK\_CUR:** mueve la posición del puntero del archivo a una ubicación determinada

**SEEK\_END:** mueve la posición del puntero del archivo al final del archivo.

Modos de apertura en archivos binarios	Operación
rb	Apertura en modo sólo lectura, el archivo debe existir
wb	Apertura en modo escritura, si el archivo existe se sobrescribe y pierde el contenido anterior. Si no existe lo crea.
ab	Apertura en modo agregar, si el archivo existe agrega los datos al final del archivo, si no existe lo crea.
rb+ ó r+b	Apertura en modo lectura / escritura, el archivo debe existir.
wb+ ó w+b	Apertura en modo lectura / escritura, el archivo debe existir, si el archivo existe se sobrescribe y pierde el contenido anterior. Si no existe lo crea.
ab+ ó a+b	Apertura en modo lectura / agregar, si el archivo existe agrega los datos al final del archivo, si no existe lo crea.

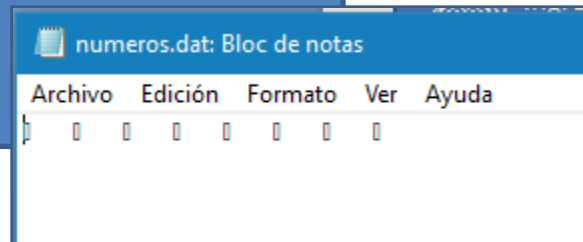
```
//ejemplo 1
#include <stdio.h>

int main(){
int num;
FILE *pArchivo;
pArchivo=fopen("numeros.dat","wb");
if(pArchivo!=NULL){
do{
printf("Ingrese un numero, para terminar 0 (cero):
");
scanf("%d",&num);
fwrite(&num,sizeof(int),1,pArchivo);
}while (num > 0);
fclose(pArchivo);
}
else{printf("Error en la apertura del archivo");}
getchar();
return 0;
}
```

Tanto a fread() como fwrite() tenemos que pasarle, además de otros parámetros, el tamaño del tipo de datos en bytes, para ello podemos utilizar el operador sizeof.

**fwrite(dirección de la variable, tamaño en bytes, cantidad, archivo);**

```
Ingrese un numero, para terminar 0 (cero): 5
Ingrese un numero, para terminar 0 (cero): 4
Ingrese un numero, para terminar 0 (cero): 3
Ingrese un numero, para terminar 0 (cero): 4
Ingrese un numero, para terminar 0 (cero): 5
Ingrese un numero, para terminar 0 (cero): 6
Ingrese un numero, para terminar 0 (cero): 7
Ingrese un numero, para terminar 0 (cero): 8
Ingrese un numero, para terminar 0 (cero): 9
Ingrese un numero, para terminar 0 (cero): 0
```



```
//ejemplo 2
#include <stdio.h>

int main(){
    int num;
    FILE *pArchivo;
    pArchivo=fopen("numeros.dat","rb");

    if(pArchivo!=NULL){
        fread(&num,sizeof(int),1,pArchivo);
        while(!feof(pArchivo)){
            printf("%d\n", num);
            fread(&num,sizeof(int),1,pArchivo);
        }
        fclose(pArchivo);
    }
    else{printf("Error en la apertura del archivo");}
    getchar();
    return 0;
}
```

Tanto a fread() como fwrite() tenemos que pasarle, además de otros parámetros, el tamaño del tipo de datos en bytes, para ello podemos utilizar el operador sizeof.

fread(dirección de la variable, tamaño en bytes, cantidad, archivo);

5  
4  
3  
4  
5  
6  
7  
8  
9  
0

```

//ejemplo 3
#include <stdio.h>

int main(){
    int num, nReg;
    FILE *pArchivo;
    pArchivo=fopen("numeros.dat","ab");
    if(pArchivo!=NULL){
        do{
            printf("Ingrese un numero, para terminar 0 (cero): ");
            scanf("%d",&num);
            fwrite(&num,sizeof(int),1,pArchivo);
        }while (num > 0);
        fclose(pArchivo);
    }
    else{printf("Error en la apertura del archivo");}
    getchar();
    return 0;
}

```

```

Ingrese un numero, para terminar 0 (cero): 78
Ingrese un numero, para terminar 0 (cero): 34
Ingrese un numero, para terminar 0 (cero): 23
Ingrese un numero, para terminar 0 (cero): 12
Ingrese un numero, para terminar 0 (cero): 0

```

```

5
4
3
4
5
6
7
8
9
0
78
34
23
12
0

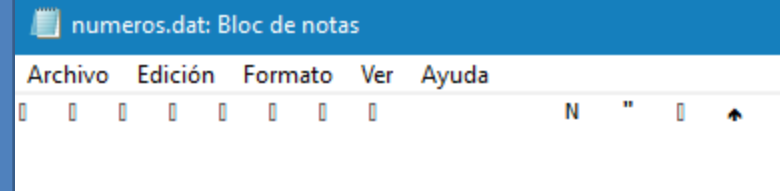
```

```
//ejemplo 4
#include <stdio.h>

int main(){
    int num, nReg;
    FILE *pArchivo;
    pArchivo=fopen("numeros.dat","r+b");
    if(pArchivo!=NULL){
        fseek(pArchivo,0,SEEK_END);
        nReg=ftell(pArchivo)/sizeof(int);
        printf("No de registros en el archivo = %d\n",nReg);
        fclose(pArchivo);
    }
    else{printf("Error en la apertura del archivo");}
    getchar();
    return 0;}

```

```
No de registros en el archivo = 15
```



numeros.dat

	5
	4
	3
	4
	5
	6
	7
	8
	9
	0

**fseek(pArchivo,0,SEEK\_END),**  
**Posiciono el puntero al final**  
**del archivo de esta manera**  
**obtengo el tamaño para ftell()**

**nReg=ftell(pArchivo)/sizeof(int);**  
**De esta manera obtengo la cantidad de**  
**enteros en el archivo.**

```
//ejemplo 5
```

```
#include <stdio.h>
```

```
int main(){
```

```
int num, nReg;
```

```
FILE *pArchivo;
```

```
pArchivo=fopen("numeros.dat","r+b");
```

```
if(pArchivo!=NULL){
```

```
    fseek(pArchivo,0,SEEK_END);
```

```
    nReg=ftell(pArchivo)/sizeof(int);
```

```
    printf("No de registros en el archivo = %d\n",nReg);
```

```
    fseek(pArchivo,0,SEEK_SET);
```

```
    fseek(pArchivo,3*sizeof(int),SEEK_SET);
```

```
    fread(&num,sizeof(int),1,pArchivo);
```

```
    printf("4to valor: %d\n",num);
```

```
    fclose(pArchivo);
```

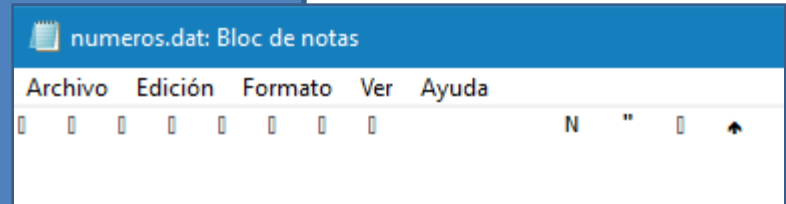
```
}
```

```
else{printf("Error en la apertura del archivo");}
```

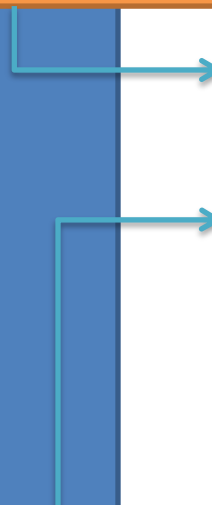
```
getchar();
```

```
return 0;}
```

```
No de registros en el archivo = 15
4to valor: 4
```



**fseek(pArchivo,0,SEEK\_SET)**



5
4
3
4
5
6
7
8
9

**fseek(pArchivo,3\*sizeof(int),SEEK\_SET);**  
**Coloca el puntero donde comienza el**  
**cuarto dato**



```
//ejemplo 6
#include <stdio.h>
int main(){
    int num, valor, nReg;
    FILE *pArchivo;
    pArchivo=fopen("numeros.dat","r+b");
    if(pArchivo!=NULL){
        printf("Ingrese un valor\n");
        scanf("%d", &valor);

        fseek(pArchivo,0,SEEK_SET);
        fseek(pArchivo,3*sizeof(int),SEEK_CUR);

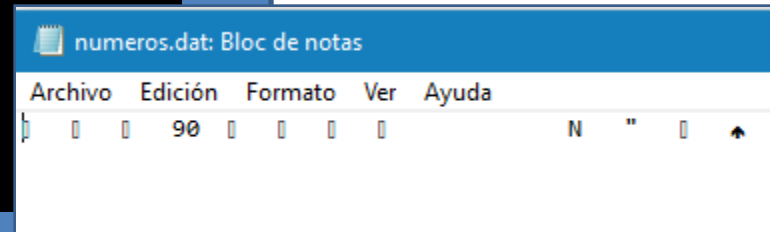
        fread(&num,sizeof(int),1,pArchivo);
        num=valor;

        fseek(pArchivo, - sizeof(int),SEEK_CUR);

        fwrite(&num,sizeof(int),1,pArchivo);
        printf("4to valor: %d\n",num);
        fclose(pArchivo);
    }
    else printf("Error en la apertura del archivo");
    getchar();
    return 0;}

```

```
Ingrese un valor
2345
4to valor: 2345
5
4
3
2345
5
6
7
8
9
0
78
34
23
12
0
```

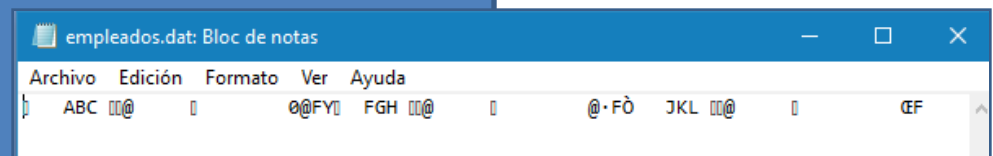


	5
	4
	3
	4
	5
	6
	7
	8
	9

**fseek(pArchivo, -sizeof(int),SEEK\_CUR);**  
**Coloca el puntero donde debo modificar el dato**

```
//ejemplo 7
#include <stdio.h>
struct persona{
    int leg;
    char ape[20];
    float suel; };
int main(){
    struct persona unapersona;
    FILE *pArchivo;
    pArchivo=fopen("empleados.dat","wb");
    if(pArchivo!=NULL){
        printf("Ingrese un numero, para terminar 0 (cero): ");
        scanf("%d",&unapersona.leg);
        fflush(stdin);
        while(unapersona.leg > 0){
            printf("Ingrese un apellido: ");
            scanf("%s",unapersona.ape);
            fflush(stdin);
            printf("Ingrese un sueldo: ");
            scanf("%f",&unapersona.suel);
            fflush(stdin);
            fwrite(&unapersona,sizeof(struct persona),1,pArchivo);
            printf("Ingrese un numero, para terminar 0 (cero): ");
            scanf("%d",&unapersona.leg);
            fflush(stdin);
        }
        fclose(pArchivo); }
    else{printf("Error en la apertura del archivo");}
    getchar();
    return 0; }
```

```
Ingrese un numero, para terminar 0 (cero): 3
Ingrese un apellido: ABC
Ingrese un sueldo: 12300
Ingrese un numero, para terminar 0 (cero): 345
Ingrese un apellido: FGH
Ingrese un sueldo: 23456
Ingrese un numero, para terminar 0 (cero): 722
Ingrese un apellido: JKL
Ingrese un sueldo: 18000
Ingrese un numero, para terminar 0 (cero): 0
```



```

//ejemplo 8
#include <stdio.h>

struct persona{
int leg;
char ape[20];
float suel;
};

int main(){
struct persona unapersona;
FILE *pArchivo;
pArchivo=fopen("empleados.dat","rb");

if(pArchivo!=NULL){
fread(&unapersona,sizeof(struct persona),1,pArchivo);
while(!feof(pArchivo)){
printf("%d - %10s - %10.2f\n",
unapersona.leg,unapersona.ape,unapersona.suel);
fread(&unapersona,sizeof(struct persona),1,pArchivo);
}
fclose(pArchivo);
}
else{printf("Error en la apertura del archivo");}
getchar();
return 0;
}

```

```

3 -      ABC -    12300.00
345 -      FGH -    23456.00
722 -      JKL -    18000.00

```

```
//ejemplo 9
#include <stdio.h>
struct persona{
    int leg;
    char ape[20];
    float suel; };
int main(){
    struct persona unapersona;
    FILE *pArchivo;
    pArchivo=fopen("empleados.dat","ab");
    if(pArchivo!=NULL){
        printf("Ingrese un numero, para terminar 0 (cero): ");
        scanf("%d",&unapersona.leg);
        fflush(stdin);
        while(unapersona.leg > 0){
            printf("Ingrese un apellido: ");
            scanf("%s",unapersona.ape);
            fflush(stdin);
            printf("Ingrese un sueldo: ");
            scanf("%f",&unapersona.suel);
            fflush(stdin);
            fwrite(&unapersona,sizeof(struct persona),1,pArchivo);
            printf("Ingrese un numero, para terminar 0 (cero): ");
            scanf("%d",&unapersona.leg);
            fflush(stdin);
        }
        fclose(pArchivo); }
    else{printf("Error en la apertura del archivo");}
    getchar();
    return 0; }
```

```
Ingrese un numero, para terminar 0 (cero): 35
Ingrese un apellido: XYZ
Ingrese un sueldo: 23456
Ingrese un numero, para terminar 0 (cero): 37
Ingrese un apellido: TGB
Ingrese un sueldo: 12345
Ingrese un numero, para terminar 0 (cero): 45
Ingrese un apellido: ERT
Ingrese un sueldo: 23789
Ingrese un numero, para terminar 0 (cero): 54
Ingrese un apellido: RDS
Ingrese un sueldo: 43256
Ingrese un numero, para terminar 0 (cero): 0
```

35	-	ABC	-	12300.00
345	-	FGH	-	23456.00
722	-	JKL	-	18000.00
35	-	XYZ	-	23456.00
37	-	TGB	-	12345.00
45	-	ERT	-	23789.00
54	-	RDS	-	43256.00

```
//ejemplo 10
#include <stdio.h>
struct persona{
    int leg;
    char ape[20];
    float suel; };

int main(){
    struct persona unapersona;
    FILE *pArchivo;
    int registro, nReg;
    char apenuevo[20];
    pArchivo=fopen("empleados.dat","r+b");
    if(pArchivo!=NULL){
        fseek(pArchivo,0,SEEK_END);
        nReg=ftell(pArchivo)/sizeof(struct persona);
        printf("No de registros en el archivo =
%d\n",nReg);
        printf("Ingrese el número de registro que quiere
modificar: ");
        scanf("%d",&registro); //ingreso 5
        fseek(pArchivo,0,SEEK_SET);
        fseek(pArchivo,(registro-1)*sizeof(struct
persona),SEEK_SET);
        fread(&unapersona,sizeof(struct
persona),1,pArchivo);
        printf("%d%10s%10.2f\n",
unapersona.leg,unapersona.ape,unapersona.suel);
        printf("Ingrese el apellido nuevo:\n");
```

```
scanf("%s",apenuevo);
strcpy(unapersona.ape, apenuevo)
fseek(pArchivo,-sizeof(struct
persona),SEEK_CUR);
fwrite(&unapersona,sizeof(struct
persona),1,pArchivo);
    fclose(pArchivo);
    }
else { printf("Error en la apertura del archivo");}
getchar();
return 0;
}
```

```
3 -      ABC -    12300.00
345 -      FGH -    23456.00
722 -      JKL -    18000.00
35 -      XYZ -    23456.00
37 -      TGB -    12345.00
45 -      ERT -    23789.00
54 -      RDS -    43256.00
```

```
No de registros en el archivo = 7
Ingrese el n-mero de registro que quiere modificar: 5
37      TGB  12345.00
Ingrese el apellido nuevo:
APENUEVO
```

```
3      ABC  12300.00
345     FGH  23456.00
722     JKL  18000.00
35     XYZ  23456.00
37  APENUEVO 12345.00
45     ERT  23789.00
54     RDS  43256.00
```

`fseek(pArchivo,0,SEEK_SET)`

Cada vez que se utiliza **fread()** y **fwrite()** el puntero pasa a la posición siguiente por lo tanto, ante una modificación, hay que reubicarlo para escribir en el archivo.

<i>legajo</i>	<i>apellido</i>	<i>sueldo</i>
3	ABC	12300
345	FGH	23456
722	JKL	18000
35	XYZ	23456
37	TGB	12345
45	ERT	23789
54	RDS	43256

empleados.dat

`fseek(pArchivo, - sizeof(struct persona),SEEK_CUR);`

`fwrite(&unapersona,sizeof(struct persona),1,pArchivo);`

`fseek(pArchivo,(5-1)*sizeof(struct persona),SEEK_SET);`

`fread(&unapersona,sizeof(struct persona),1,pArchivo);`

```
3 -      ABC -    12300.00
345 -      FGH -   23456.00
722 -      JKL -   18000.00
35 -      XYZ -   23456.00
37 -      TGB -   12345.00
45 -      ERT -   23789.00
54 -      RDS -   43256.00
```

```
3      ABC  12300.00
345     FGH  23456.00
722     JKL  18000.00
35     XYZ  23456.00
37  APENUEVO 12345.00
45     ERT  23789.00
54     RDS  43256.00
```

**FIN**  
FIN