

Universidad de San Carlos de Guatemala Centro Universitario de Oriente -
CUNORI-
Ingeniería en Ciencias y Sistemas

Maneja e Implementación de Archivos

Ing. Indira Valdez

Practica #4 Escritura y Lectura de Registros en Archivos Binarios

Mynor Bezaleel Ramos González

201944540

05 de septiembre de 2022

Ejercicio 1:

```
#include "stdio.h"
#include "stdlib.h"
#include "string.h"

typedef struct //Estructura de tipo Artículo
{
    int codART; //Codigo del artículo
    float precio; //Precio del artículo
    char descrip [50]; //Descripcion del artículo
} Artículo; //Tipo de dato Artículo

int main ()
{
    Artículo articulos [10]= { //Declaracion de un arreglo de tipo Artículo

        [0]: { .codART: 1, .precio: 10.0, .descrip: "Pelota de futbol"}, //Inicializacion de los artículos
        [1]: { .codART: 2, .precio: 20.0, .descrip: "Raqueta de tenis"},
        [2]: { .codART: 3, .precio: 30.0, .descrip: "Pelota de baloncesto"},
        [3]: { .codART: 4, .precio: 40.0, .descrip: "Bicicleta"},
        [4]: { .codART: 5, .precio: 50.0, .descrip: "Guantes de boxeo"},
        [5]: { .codART: 6, .precio: 60.0, .descrip: "Gafas de natacion"},
        [6]: { .codART: 7, .precio: 70.0, .descrip: "Zapatillas de correr"},
        [7]: { .codART: 8, .precio: 80.0, .descrip: "Casco para ciclismo"},
        [8]: { .codART: 9, .precio: 90.0, .descrip: "Pelota de voleibol"},
        [9]: { .codART: 10, .precio: 100.0, .descrip: "Equipo de golf"},
    };

    FILE * pf = fopen ( filename: "Articulos.dat", mode: "wb"); //Declaracion de un archivo binario
    if (pf == NULL) //Si el archivo no se abre correctamente
        printf ("Error al abrir el archivo. \n"); //Se imprime un mensaje de error en pantalla
    fwrite ( ptr: &articulos[0], size: sizeof (Articulo), nitems: 10, stream: pf); //Se escribe el arreglo en el archivo binario
    return 0; //Se retorna 0 si el programa se ejecuta correctamente
};
```

WB: El modo wb abre un archivo binario para escritura. Si el archivo no existe, se crea. Si existe, se sobrescribe. Recibe como argumentos el nombre del archivo y el modo de apertura.

FWRITE: Se utiliza para escribir los datos de los artículos almacenados en el arreglo “articulos” en el archivo binario “Articulos.dat”.

Parametros de FWRITE: (&articulos[0], sizeof(Articulo), 10, pf)

&articulos[0]: Puntero al inicio del bloque de memoria que se va a escribir en el archivo.

sizeof(Articulo): Especifica el tamaño en bytes de cada elemento que se va a escribir en el archivo.

10: Indica el numero de elementos que se va a escribir en el archivo.

pf: Es un puntero al archivo en el que se escribirán los datos.

Este código crea una estructura de datos de tipo Artículo, inicializa un arreglo de 10 elementos de tipo Artículo con datos de ejemplo y luego escribe estos datos en un archivo binario llamado "Articulos.dat".

Ejercicio 2:

```
#include "stdio.h"
#include "stdlib.h"
#include "string.h"

typedef struct //Estructura de tipo Artículo
{
    int codART; //Codigo del artículo
    float precio; //Precio del artículo
    char descrip [50]; //Descripcion del artículo
} Artículo; //Tipo de dato Artículo

int main ()
{
    Artículo articulos [10]= { //Declaracion de un arreglo de tipo Artículo

        [0]: { .codART: 1, .precio: 10.0, .descrip: "Pelota de futbol"}, //Inicializacion de los articulos
        [1]: { .codART: 2, .precio: 20.0, .descrip: "Raqueta de tenis"},
        [2]: { .codART: 3, .precio: 30.0, .descrip: "Pelota de baloncesto"},
        [3]: { .codART: 4, .precio: 40.0, .descrip: "Bicicleta"},
        [4]: { .codART: 5, .precio: 50.0, .descrip: "Guantes de boxeo"},
        [5]: { .codART: 6, .precio: 60.0, .descrip: "Gafas de natacion"},
        [6]: { .codART: 7, .precio: 70.0, .descrip: "Zapatillas de correr"},
        [7]: { .codART: 8, .precio: 80.0, .descrip: "Casco para ciclismo"},
        [8]: { .codART: 9, .precio: 90.0, .descrip: "Pelota de voleibol"},
        [9]: { .codART: 10, .precio: 100.0, .descrip: "Equipo de golf"},

    };

    FILE* pf = fopen ( filename: "Articulos.dat", mode: "rb"); //Declaracion de un archivo binario
    if (pf == NULL) //Si el archivo no se abre correctamente
        printf ("Error al abrir el archivo. \n"); //Se imprime un mensaje de error en pantalla

    Artículo aux; //Declaracion de una variable auxiliar de tipo Artículo
    fread( ptr: &aux, size: sizeof(Artículo), nitems: 1, stream: pf); //Se lee el primer artículo del archivo binario
    printf("-----EL ARTICULO SOLICITADO ES ----- \n"); //Se imprime un mensaje en pantalla
    printf("CODIGO_ARTICULO : %d\n", aux.codART); //Se imprime el código del artículo en pantalla
    printf("PRECIO_ARTICULO : %.2f\n", aux.precio); //Se imprime el precio del artículo en pantalla
    printf("DESCRIPCION_ARTICULO : %-20s \n \n", aux.descrip); //Se imprime la descripción del artículo en pantalla

    return 0;
};
```

RB: El modo rb sirve para abrir un archivo binario en modo lectura. Recibe como parametros el nombre del archivo y el modo de apertura.

FREAD: Se utiliza para leer datos desde un archivo binario.

Parametros de FREAD: (void *ptr, size_t size, size_t count, FILE *stream)

Pttr: Es un punter el area de memoria donde se almacenaran los datos leidos desde el archivo.

Size: Indica el tamaño en bytes de cada elemento que se va a leer desde el archivo.

Count: Representa la cantidad de elementos que se desean leer desde el archive.

Stream: Es un punter al archivo desde el cual se leeran los datos.

El código abre un archivo binario llamado "Articulos.dat", lee un artículo de dicho archivo y muestra sus datos en pantalla.

Ejercicio 3:

```
#include "stdio.h"
#include "stdlib.h"
#include "string.h"

typedef struct //Estructura de tipo Artículo
{
    int codART; //Codigo del artículo
    float precio; //Precio del artículo
    char descrip [50]; //Descripcion del artículo
} Artículo; //Tipo de dato Artículo

int main ()
{
    Artículo articulos [10]= { //Declaracion de un arreglo de tipo Artículo

        [0]: { .codART: 1, .precio: 10.0, .descrip: "Pelota de futbol"}, //Iniciailizacion de los articulos
        [1]: { .codART: 2, .precio: 20.0, .descrip: "Raqueta de tenis"},
        [2]: { .codART: 3, .precio: 30.0, .descrip: "Pelota de baloncesto"},
        [3]: { .codART: 4, .precio: 40.0, .descrip: "Bicicleta"},
        [4]: { .codART: 5, .precio: 50.0, .descrip: "Guantes de boxeo"},
        [5]: { .codART: 6, .precio: 60.0, .descrip: "Gafas de natacion"},
        [6]: { .codART: 7, .precio: 70.0, .descrip: "Zapatillas de correr"},
        [7]: { .codART: 8, .precio: 80.0, .descrip: "Casco para ciclismo"},
        [8]: { .codART: 9, .precio: 90.0, .descrip: "Pelota de voleibol"},
        [9]: { .codART: 10, .precio: 100.0, .descrip: "Equipo de golf"},

    };

    FILE* pf = fopen ( filename: "Articulos.dat", mode: "rb"); //Declaracion de un archivo binario
    if (pf == NULL) //Si el archivo no se abre correctamente
        printf ("Error al abrir el archivo. \n"); //Se imprime un mensaje de error en pantalla

    Artículo aux; //Declaracion de una variable auxiliar de tipo Artículo
    fseek(pf, sizeof(Artículo), SEEK_SET); //Se posiciona el puntero en el segundo artículo del archivo binario
    fread( ptr: &aux, size: sizeof(Artículo), nitems: 1, stream: pf); //Se lee el segundo artículo del archivo binario
    printf("-----EL ARTICULO SOLICITADO ES ----- \n"); //Se imprime un mensaje en pantalla
    printf("CODIGO_ARTICULO : %d\n", aux.codART); //Se imprime el código del artículo en pantalla
    printf("PRECIO_ARTICULO : %.2f\n", aux.precio); //Se imprime el precio del artículo en pantalla
    printf("DESCRIPCION_ARTICULO : %-20s \n \n", aux.descrip); //Se imprime la descripción del artículo en pantalla

    return 0;
};
```

FSEEK: Sirve para mover el puntero de posición en un archivo, permitiendo navegar y acceder a datos en ubicaciones específicas dentro del archivo.

Parametros de FSEEK: int fseek(FILE *stream, long offset, int whence);

Stream: Es un puntero al archivo en el que desea mover el puntero de posición.

Offset: Indica la cantidad de bytes que se movera al puntero de posición.

Whence: Es un indicador que especifica la referencia desde la cual se realizara el desplazamiento. **SEEK_SET (0):** Establece el puntero al principio del archivo y luego se desplaza desde allí según el valor de offset.

El código sirve para abrir un archivo binario que contiene datos de artículos, mover el puntero de lectura al segundo artículo y luego mostrar la información de ese segundo artículo en la pantalla.

Ejercicio 4:

```
#include "stdio.h"
#include "stdlib.h"
#include "string.h"

typedef struct //Estructura de tipo Articulo
{
    int codART; //Codigo del articulo
    float precio; //Precio del articulo
    char descrip [50]; //Descripcion del articulo
} Articulo; //Tipo de dato Articulo

int main ()
{
    Articulo articulos [10]= { //Declaracion de un arreglo de tipo Articulo

        [0]: { .codART: 1, .precio: 10.0, .descrip: "Pelota de futbol"}, //Inicializacion de los articulos
        [1]: { .codART: 2, .precio: 20.0, .descrip: "Raqueta de tenis"},
        [2]: { .codART: 3, .precio: 30.0, .descrip: "Pelota de baloncesto"},
        [3]: { .codART: 4, .precio: 40.0, .descrip: "Bicicleta"},
        [4]: { .codART: 5, .precio: 50.0, .descrip: "Guantes de boxeo"},
        [5]: { .codART: 6, .precio: 60.0, .descrip: "Gafas de natacion"},
        [6]: { .codART: 7, .precio: 70.0, .descrip: "Zapatillas de correr"},
        [7]: { .codART: 8, .precio: 80.0, .descrip: "Casco para ciclismo"},
        [8]: { .codART: 9, .precio: 90.0, .descrip: "Pelota de voleibol"},
        [9]: { .codART: 10, .precio: 100.0, .descrip: "Equipo de golf"},

    };

    FILE* pf = fopen ( filename: "Articulos.dat", mode: "rb"); //Declaracion de un archivo binario
    if (pf == NULL) //Si el archivo no se abre correctamente
        printf ("Error al abrir el archivo. \n"); //Se imprime un mensaje de error en pantalla

    Articulo aux; //Declaracion de una variable auxiliar de tipo Articulo
    fseek(pf, sizeof(Articulo), SEEK_END); //Se posiciona el puntero en el segundo articulo del archivo binario (desde el final)
    fread( ptr: &aux, size: sizeof(Articulo), nitens: 1, stream: pf); //Se lee el segundo articulo del archivo binario
    printf("-----EL ARTICULO SOLICITADO ES ----- \n"); //Se imprime un mensaje en pantalla
    printf("CODIGO ARTICULO : %d\n", aux.codART); //Se imprime el codigo del articulo en pantalla
    printf("PRECIO ARTICULO : %.2f\n", aux.precio); //Se imprime el precio del articulo en pantalla
    printf("DESCRIPCION ARTICULO : %-20s \n\n", aux.descrip); //Se imprime la descripcion del articulo en pantalla

    return 0;
};
```

FSEEK: Sirve para mover el puntero de posición en un archivo, permitiendo navegar y acceder a datos en ubicaciones específicas dentro del archivo.

Parametros de FSEEK: int fseek(FILE *stream, long offset, int whence);

Stream: Es un puntero al archivo en el que desea mover el puntero de posición.

Offset: Indica la cantidad de bytes que se movera al puntero de posición.

Whence: Es un indicador que especifica la referencia desde la cual se realizara el desplazamiento.
SEEK_END (2): Establece el puntero al final del archivo y luego se desplaza hacia atrás según el valor de offset.

El código abre un archivo binario llamado "Articulos.dat", se posiciona al final del archivo y lee el último artículo almacenado en el archivo, mostrando su información en la pantalla.

Ejercicio 5:

```
#include "stdio.h"
#include "stdlib.h"
#include "string.h"

typedef struct //Estructura de tipo Artículo
{
    int codART; //Codigo del artículo
    float precio; //Precio del artículo
    char descrip [50]; //Descripcion del artículo
} Artículo; //Tipo de dato Artículo

int main ()
{
    Artículo articulos [10]= {
        [0]: { .codART: 1, .precio: 10.0, .descrip: "Pelota de futbol"},
        [1]: { .codART: 2, .precio: 20.0, .descrip: "Raqueta de tenis"},
        [2]: { .codART: 3, .precio: 30.0, .descrip: "Pelota de baloncesto"},
        [3]: { .codART: 4, .precio: 40.0, .descrip: "Bicicleta"},
        [4]: { .codART: 5, .precio: 50.0, .descrip: "Guantes de boxeo"},
        [5]: { .codART: 6, .precio: 60.0, .descrip: "Gafas de natacion"},
        [6]: { .codART: 7, .precio: 70.0, .descrip: "Zapatillas de correr"},
        [7]: { .codART: 8, .precio: 80.0, .descrip: "Casco para ciclismo"},
        [8]: { .codART: 9, .precio: 90.0, .descrip: "Pelota de voleibol"},
        [9]: { .codART: 10, .precio: 100.0, .descrip: "Equipo de golf"},
    };

    FILE * pf = fopen ( filename: "Articulos.dat", mode: "rb"); //Declaracion de un archivo binario
    if (pf == NULL) //Si el archivo no se abre correctamente
        printf ("Error al abrir el archivo. \n"); //Se imprime un mensaje de error en pantalla

    Artículo aux; //Declaracion de una variable auxiliar de tipo Artículo
    fseek(pf, 0L, SEEK_END); //Se posiciona el puntero en el segundo artículo del archivo binario
    int cant_registros = ftell(pf) / sizeof(Artículo); //Se calcula la cantidad de registros del archivo binario
    rewind(pf); //Se posiciona el puntero al inicio del archivo binario
    fread( ptr: &aux, size: sizeof(Artículo), nitems: 1, stream: pf); //Se lee el segundo artículo del archivo binario
    printf("-----CANTIDAD DE REGISTROS EN EL FICHERO ----- \n"); //Se imprime un mensaje en pantalla
    printf("----- %d ----- \n", cant_registros); //Se imprime la cantidad de registros en pantalla
    printf("-----EL ARTICULO SOLICITADO ES ----- \n"); //Se imprime un mensaje en pantalla
    printf("CODIGO_ARTICULO : %d \n", aux.codART); //Se imprime el código del artículo en pantalla
    printf("PRECIO_ARTICULO : %.2f \n", aux.precio); //Se imprime el precio del artículo en pantalla
    printf("DESCRIPCION_ARTICULO : %-20s \n \n", aux.descrip); //Se imprime la descripción del artículo en pantalla
    return 0;
}
```

REWIND: Se utiliza para reposiciona el puntero de una archivo al principio del archivo.

Parametros: void rewind (FILE *stream).

Stream: Un punter a un objeto de tipo (FILE) que representa el archivo al que se desea reposicionar el puntero.

El código abre un archivo binario llamado "Articulos.dat", calcula la cantidad de registros en el archivo, y muestra la información del segundo artículo almacenado en el archivo, junto con la cantidad total de registros.

Ejercicio 6:

```
typedef struct //Estructura de tipo Artículo
{
    int codART; //Codigo del artículo
    float precio; //Precio del artículo
    char descrip [50]; //Descripción del artículo
} Artículo; //Tipo de dato Artículo

void mostrarRegistros (void* dato)
{
    Artículo *aux =(Artículo*) dato; //Declaración de una variable auxiliar de tipo Artículo
    printf("-----\n"); //Se imprime un mensaje en pantalla
    printf("CODIGO_ARTICULO : %d\n", aux->codART); //Se imprime el código del artículo en pantalla
    printf("PRECIO_ARTICULO : %.2f\n", aux->precio); //Se imprime el precio del artículo en pantalla
    printf("DESCRIPCION_ARTICULO : %-20s \n \n", aux->descrip); //Se imprime la descripción del artículo en pantalla
}

int main ()
{
    Artículo articulos [10]= { //Declaración de un arreglo de tipo Artículo

        [0]: { .codART: 1, .precio: 18.8, .descrip: "Pelota de futbol"},
        [1]: { .codART: 2, .precio: 28.8, .descrip: "Raqueta de tenis"},
        [2]: { .codART: 3, .precio: 38.8, .descrip: "Pelota de baloncesto"},
        [3]: { .codART: 4, .precio: 48.8, .descrip: "Bicicleta"},
        [4]: { .codART: 5, .precio: 58.8, .descrip: "Guantes de boxeo"},
        [5]: { .codART: 6, .precio: 68.8, .descrip: "Gafas de natacion"},
        [6]: { .codART: 7, .precio: 78.8, .descrip: "Zapatillas de correr"},
        [7]: { .codART: 8, .precio: 88.8, .descrip: "Casco para ciclismo"},
        [8]: { .codART: 9, .precio: 98.8, .descrip: "Pelota de voleibol"},
        [9]: { .codART: 10, .precio: 108.8, .descrip: "Equipo de golf"},

    };

    FILE* pf = fopen ( filename: "Articulos.dat", mode: "rb"); //Declaración de un archivo binario
    if (pf == NULL) //Si el archivo no se abre correctamente
        printf ("Error al abrir el archivo. \n"); //Se imprime un mensaje de error en pantalla

    fseek(pf, 0, SEEK_END); //Se posiciona el puntero en el segundo artículo del archivo binario
    int cant_registros = ftell(pf) / sizeof(Artículo); //Se calcula la cantidad de registros del archivo binario
    rewind(pf); //Se posiciona el puntero al inicio del archivo binario
    void* vec = malloc( size: sizeof(Artículo) * cant_registros); //Se reserva memoria para un vector de tipo Artículo
    fread( ptr: vec, size: sizeof(Artículo), ritems: cant_registros, stream: pf); //Se lee el archivo binario y se almacena en el vector

    for(int i=0; i < cant_registros; i++, vec+=sizeof(Artículo)) //Se recorre el vector de tipo Artículo
        mostrarRegistros( dato: vec); //Se llama a la función mostrarRegistros
    return 0; // Se retorna 0
};
```

El código abre un archivo binario llamado "Articulos.dat", lee todos los registros del archivo y luego imprime la información de los artículos almacenados en el archivo utilizando la función mostrarRegistros.

Ejercicio 7:

```
#include "stdio.h"
#include "stdlib.h"
#include "string.h"

typedef struct //Estructura de tipo Artículo
{
    int codART; //Codigo del artículo
    float precio; //Precio del artículo
    char descrip [50]; //Descripción del artículo
} Artículo; //Tipo de dato Artículo

void mostrarArtículo (void* dato); //Prototipo de la función mostrarArtículo
void mostrarArchivo (const char* nombre); //Prototipo de la función mostrarArchivo
int generarArchivo(const char* nombre); //Prototipo de la función generarArchivo

void actualizarPrecios (const char* nombre, int cantRegistros){ //Función actualizarArchivo
    FILE *arch = fopen( filename: nombre, mode: "r+b" ); //Declaración de un archivo binario en modo lectura y escritura
    if(arch == NULL){ //Si el archivo no se abre correctamente
        printf("Error al abrir el archivo. %s \n", nombre); //Se imprime un mensaje de error en pantalla
        return; //Se retorna
    }

    Artículo aux; //Declaración de una variable auxiliar de tipo Artículo
    unsigned tam = sizeof(Artículo); //Declaración de una variable de tipo unsigned
    for (int i = 0; i < cantRegistros; i++){ //Ciclo for para recorrer el archivo
        fflush(arch); //Se limpia el buffer del archivo
        fread( ptr: &aux, size: tam, nitens: 1, stream: arch); //Se lee el artículo del archivo
        aux.precio *= 1.10; //Se actualiza el precio del artículo
        fseek(arch, -tam, SEEK_CUR); //Se posiciona el puntero en el artículo leído
        fwrite( ptr: &aux, size: tam, nitens: 1, stream: arch); //Se escribe el artículo en el archivo
    }
    fclose(arch); //Se cierra el archivo
}

int main() //Función principal
{
    int cantRegistros; //Declaración de una variable de tipo entero
    cantRegistros = generarArchivo( nombre: "Articulos2.dat"); //Se llama a la función generarArchivo
    mostrarArchivo( nombre: "Articulos2.dat"); //Se llama a la función mostrarArchivo
    actualizarPrecios( nombre: "Articulos2.dat", cantRegistros); //Se llama a la función actualizarArchivo
    printf("Artículos con precios actualizados: \n"); //Se imprime un mensaje en pantalla
    mostrarArchivo( nombre: "Articulos2.dat"); //Se llama a la función mostrarArchivo
    return 0; //Se retorna 0
}

int generarArchivo(const char* nombre){
    Artículo articulos [10]= { //Declaración de un arreglo de tipo Artículo
        [0]: { .codART: 1, .precio: 10.0, .descrip: "Pelota de futbol"}, //Iniciación de los artículos
        [1]: { .codART: 2, .precio: 20.0, .descrip: "Raqueta de tenis"}, //Iniciación de los artículos
        [2]: { .codART: 3, .precio: 30.0, .descrip: "Pelota de baloncesto"}, //Iniciación de los artículos
        [3]: { .codART: 4, .precio: 40.0, .descrip: "Bicicleta"}, //Iniciación de los artículos
        [4]: { .codART: 5, .precio: 50.0, .descrip: "Guantes de boxeo"}, //Iniciación de los artículos
        [5]: { .codART: 6, .precio: 60.0, .descrip: "Gafas de natación"}, //Iniciación de los artículos
        [6]: { .codART: 7, .precio: 70.0, .descrip: "Zapatillas de correr"}, //Iniciación de los artículos
        [7]: { .codART: 8, .precio: 80.0, .descrip: "Casco para ciclismo"}, //Iniciación de los artículos
        [8]: { .codART: 9, .precio: 90.0, .descrip: "Pelota de voleibol"}, //Iniciación de los artículos
        [9]: { .codART: 10, .precio: 100.0, .descrip: "Equipo de golf"}, //Iniciación de los artículos
    };

    int ce = sizeof(articulos) / sizeof(articulos[0]); //Declaración de una variable de tipo entero
    printf("ce: %d\n", ce); //Se imprime la cantidad de elementos del arreglo en pantalla
}
```



```

FILE* pf = fopen( filename: nombre, mode: "wb"); //Declaracion de un archivo binario en modo escritura
if (!pf) //Si el archivo no se abre correctamente
{
    printf("Fallo al escribir. %s \n", nombre); //Se imprime un mensaje de error en pantalla
    return 0; //Se retorna 0
}
fwrite( ptr: articulos, size: sizeof(Articulo), nitems: ce, stream: pf); //Se escribe el arreglo en el archivo binario
fclose(pf); //Se cierra el archivo
return ce; //Se retorna la cantidad de elementos del arreglo
};

void mostrarArchivo (const char* nombre){ //Funcion mostrarArchivo
FILE* pf = fopen( filename: nombre, mode: "rb"); //Declaracion de un archivo binario en modo lectura
if (!pf) //Si el archivo no se abre correctamente
{
    printf("Fallo al escribir el archivo. %s \n", nombre); //Se imprime un mensaje de error en pantalla
    return; //Se retorna
}
Articulo aux; //Declaracion de una variable auxiliar de tipo Articulo
printf("Cod\t Precio Descripcion "); //Se imprime un mensaje en pantalla
while (fread( ptr: &aux, size: sizeof(Articulo), nitems: 1, stream: pf)) //Ciclo while para recorrer el archivo
    mostrarArticulo( dato: &aux); //Se llama a la funcion mostrarArticulo
fclose(pf); //Se cierra el archivo
}

void mostrarArticulo (void* dato){ //Funcion mostrarArticulo
Articulo* aux = (Articulo*) dato; //Declaracion de una variable auxiliar de tipo Articulo
printf("%d\t %.2f %-20s \n", aux->codART, aux->precio, aux->descrip); //Se imprime el articulo en pantalla
}

```

R+B: se utiliza para abrir un archivo en modo lectura y escritura binaria.

El código crea un archivo binario, muestra su contenido, actualiza los precios de los artículos en el archivo y muestra el contenido actualizado. Este código es un ejemplo de manipulación de archivos binarios en C.