



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Karina Morales Garcia

Asignatura: Fundamentos de programacion

Grupo: 20

No. de práctica(s): 12

Integrante(s): Avila Pineda Samuel David

No. de lista o brigada: 06

Semestre: 2023-1

Fecha de entrega: 10 de enero del 2023

Observaciones:

CALIFICACIÓN: _____

Objetivo:

El alumno elaborará programas en lenguaje C que requieran el uso de archivos de texto plano en la resolución de problemas, entendiendo a los archivos como un elemento de almacenamiento secundario.

Actividades:

- A través de programas en C, emplear las funciones para crear, leer, escribir y sobrescribir archivos de texto plano.
- Manipular archivos empleando los diferentes tipos de acceso a ellos

¿Qué es un archivo?

Es un conjunto de datos estructurados en una colección de entidades elementales o básicas denominadas registros que son del mismo tipo.

Apuntador a archivo

Es un hilo común que unifica el sistema de Entrada/Salida con un buffer donde se transportan los datos. Este señala la información que contiene y define ciertas características sobre él, incluyendo nombre, estado y la posición del archivo. Cabe recalcar que se manejan en lenguaje C como variables de tipo FILE. La sintaxis es:

FILE *F;

Abrir archivo

La función fopen() abre una secuencia para que pueda ser utilizada y la asocia a un archivo. La sintaxis es:

*FILE fopen(char *nombre_archivo, char *modo);

Donde nombre_archivo es un puntero a una cadena de caracteres que representa un nombre válido. La cadena a la que apunta modo determina como se abre el archivo.

Tipos de modos de apertura de archivos.

r: Abre un archivo de texto para lectura.

w: Crea un archivo de texto para escritura.

a: Abre un archivo de texto para añadir.

r+: Abre un archivo de texto para lectura / escritura.

w+: Crea un archivo de texto para lectura / escritura.

a+: Añade o crea un archivo de texto para lectura / escritura.

rb: Abre un archivo en modo lectura y binario.

wb: Crea un archivo en modo escritura y binario.

Cerrar archivo.

La función fclose() cierra una secuencia que fue abierta mediante una llamada a fopen().

Un error en el cierre de una secuencia puede generar todo tipo de problemas, incluyendo la pérdida de datos, destrucción y posibles errores intermitentes en el programa. La sintaxis es.

int fclose (FILE *apArch);

Donde apArch es el apuntador al archivo devuelto por la llamada a fopen(). Si se devuelve un valor cero significa que la operación de cierre ha tenido éxito.

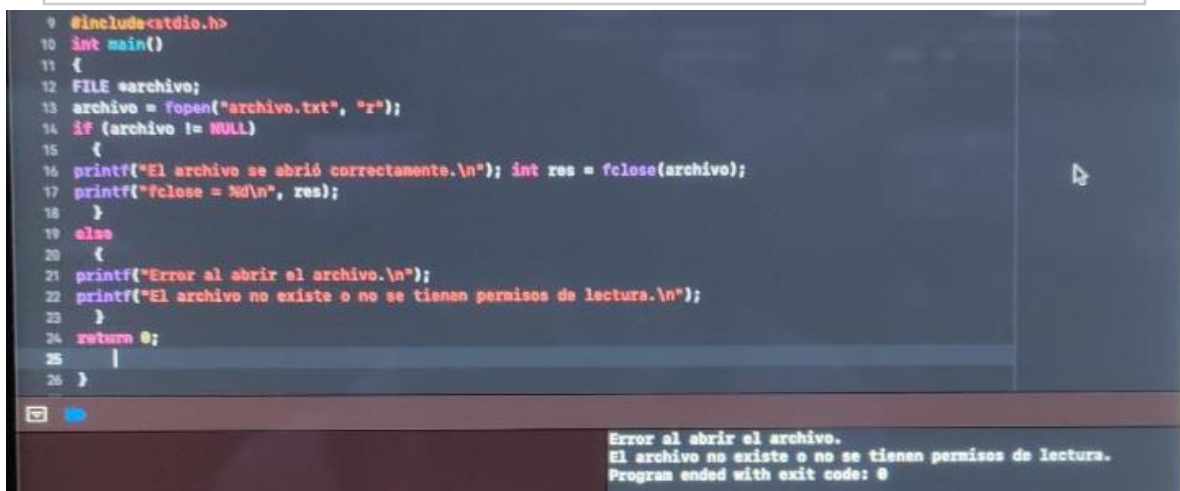
Programa 1.c

```
#include<stdio.h>

int main()
{
    FILE *archivo;
    archivo = fopen("archivo.txt", "r");

    if (archivo != NULL)
    {
        printf("El archivo se abrió correctamente.\n");
        int res = fclose(archivo);
        printf("fclose = %d\n", res);
    }
    else
    {
        printf("Error al abrir el archivo.\n");
        printf("El archivo no existe o no se tienen permisos de lectura.\n");
    }

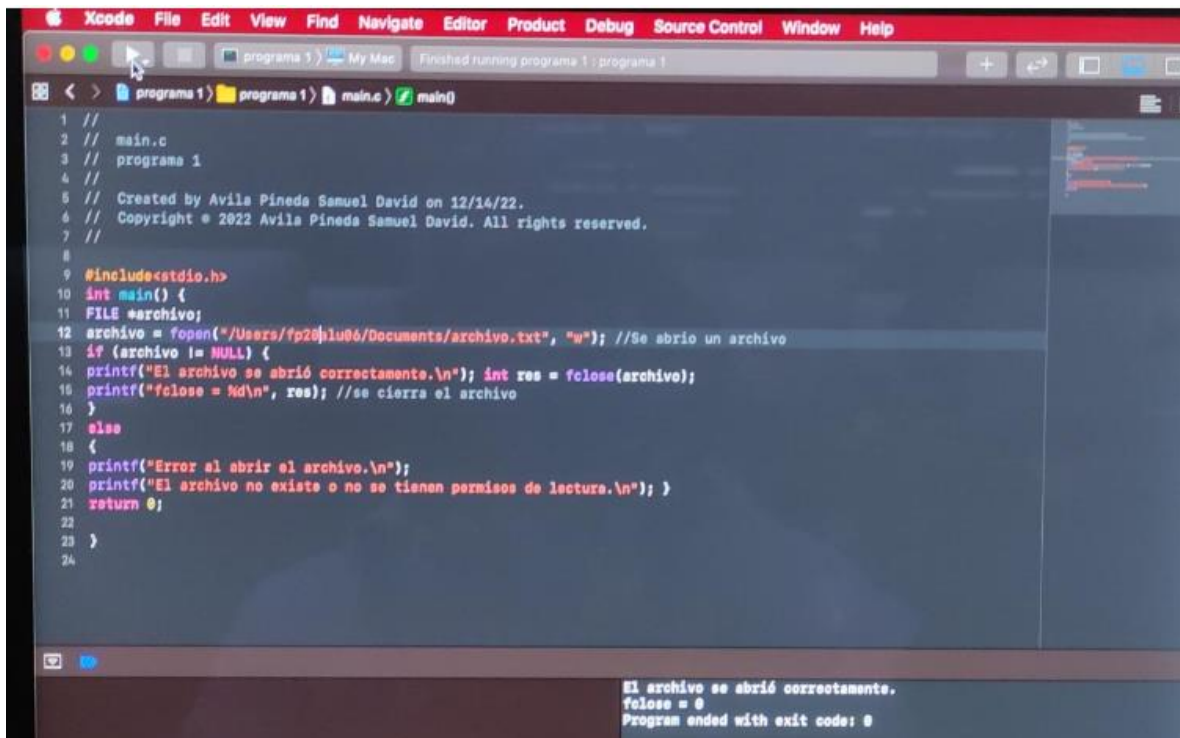
    return 0;
}
```



```
9 #include<stdio.h>
10 int main()
11 {
12     FILE *archivo;
13     archivo = fopen("archivo.txt", "r");
14     if (archivo != NULL)
15     {
16         printf("El archivo se abrió correctamente.\n"); int res = fclose(archivo);
17         printf("fclose = %d\n", res);
18     }
19     else
20     {
21         printf("Error al abrir el archivo.\n");
22         printf("El archivo no existe o no se tienen permisos de lectura.\n");
23     }
24     return 0;
25 }
26
```

Error al abrir el archivo.
El archivo no existe o no se tienen permisos de lectura.
Program ended with exit code: 0

Este fue el primer programa que se ejecutó, sin embargo, nos salió error al abrir el archivo por lo que tuvimos que modificarlo.



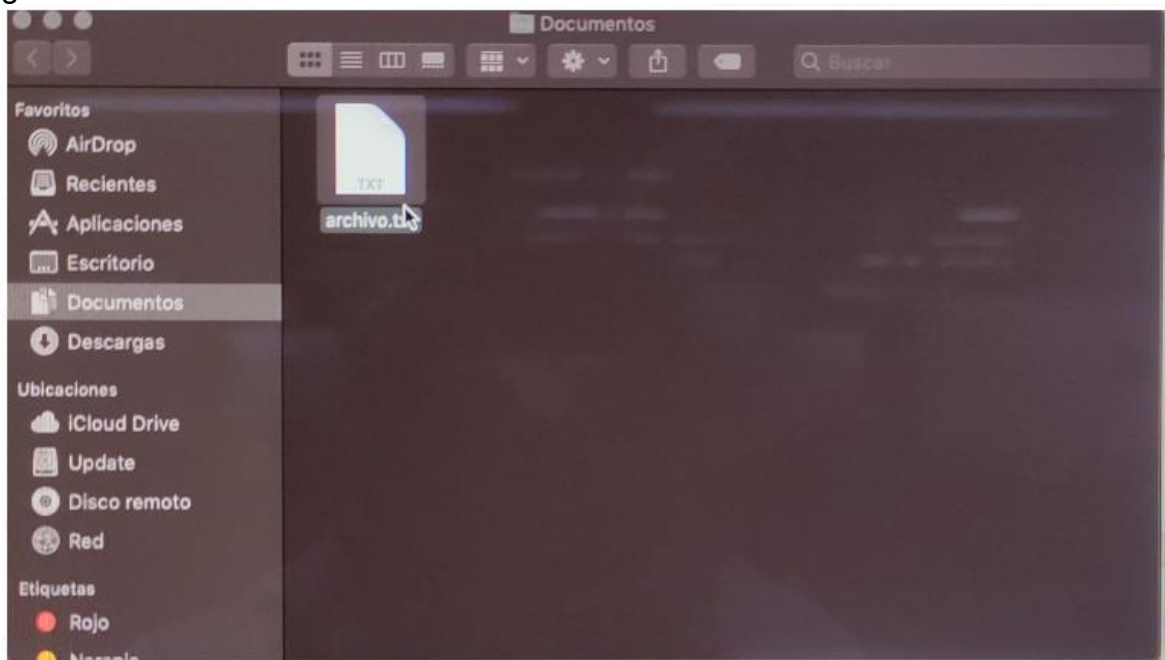
The screenshot shows the Xcode IDE with a C program named 'main.c' in a project called 'programa 1'. The code is as follows:

```
1 //
2 // main.c
3 // programa 1
4 //
5 // Created by Avila Pineda Samuel David on 12/14/22.
6 // Copyright © 2022 Avila Pineda Samuel David. All rights reserved.
7 //
8
9 #include<stdio.h>
10 int main() {
11     FILE *archivo;
12     archivo = fopen("/Users/fp201u06/Documents/archivo.txt", "w"); //Se abrio un archivo
13     if (archivo != NULL) {
14         printf("El archivo se abrió correctamente.\n"); int res = fclose(archivo);
15         printf("fclose = %d\n", res); //se cierra el archivo
16     }
17     else
18     {
19         printf("Error al abrir el archivo.\n");
20         printf("El archivo no existe o no se tienen permisos de lectura.\n"); }
21     return 0;
22 }
23
24
```

The output window at the bottom shows the following messages:

```
El archivo se abrió correctamente.
fclose = 0
Program ended with exit code: 0
```

En la imagen que se muestra arriba se comentó la función open y el close y se guardó el archivo.



Funciones *fgets* y *fputs*

Estas dos pueden leer y escribir, respectivamente, cadenas sobre los archivos. Las firmas de estas funciones son:

```
char *fgets(char *buffer, int tamaño, FILE *apArch);
char *fputs(char *buffer, FILE *apArch);
```

La función `fputs()` permite escribir una cadena en un archivo específico y la función

fgets() permite leer una cadena.

Programa 2.c

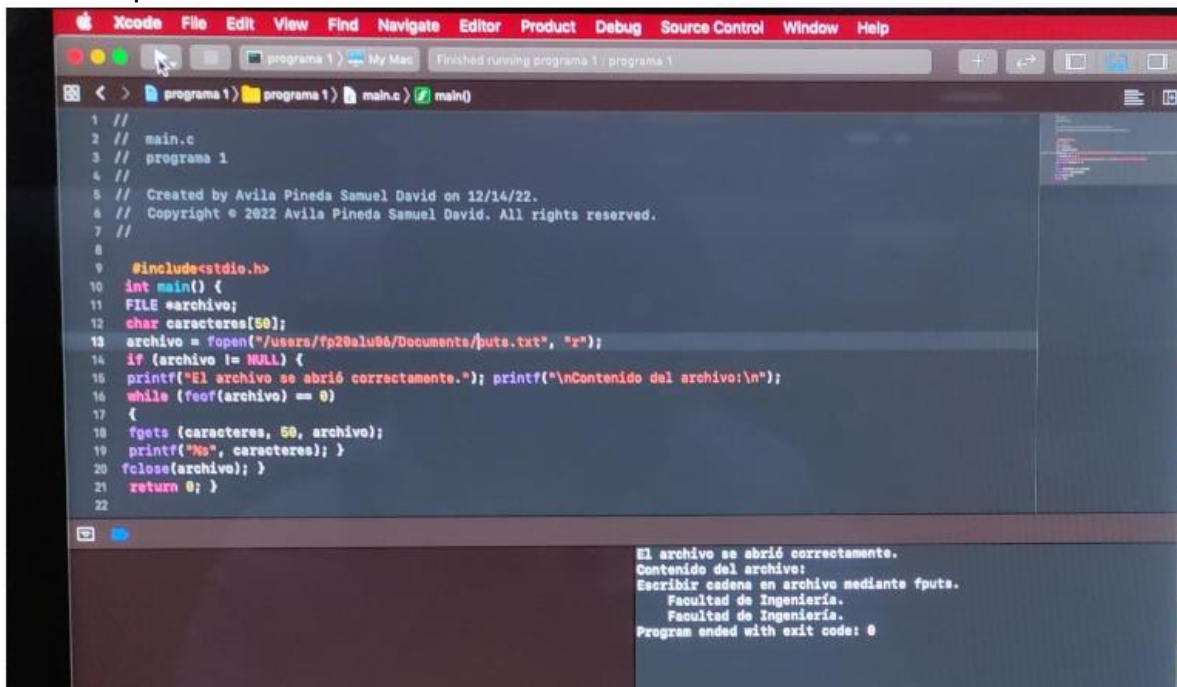
```
#include<stdio.h>

int main()
{
    FILE *archivo;
    char caracteres[50];
    archivo = fopen("gets.txt", "r");

    if (archivo != NULL)
    {
        printf("El archivo se abrió correctamente.");
        printf("\nContenido del archivo:\n");
        while (feof(archivo) == 0)
        {
            fgets (caracteres, 50, archivo);
            printf("%s", caracteres);
        }
        fclose(archivo);
    }

    return 0;
}
```

Primero se elaboró el programa 3 ya que no nos arrojaba lo que queríamos, además tuvimos que hacer unas modificaciones.



```
1 //
2 // main.c
3 // programa 1
4 //
5 // Created by Avila Pineda Samuel David on 12/14/22.
6 // Copyright © 2022 Avila Pineda Samuel David. All rights reserved.
7 //
8
9 #include<stdio.h>
10 int main() {
11     FILE *archivo;
12     char caracteres[50];
13     archivo = fopen("/users/fp20alu06/Documents/puts.txt", "r");
14     if (archivo != NULL) {
15         printf("El archivo se abrió correctamente."); printf("\nContenido del archivo:\n");
16         while (feof(archivo) == 0)
17         {
18             fgets (caracteres, 50, archivo);
19             printf("%s", caracteres); }
20         fclose(archivo); }
21     return 0; }
22
```

El archivo se abrió correctamente.
Contenido del archivo:
Escribir cadena en archivo mediante fputs.
Facultad de Ingeniería.
Facultad de Ingeniería.
Program ended with exit code: 0

En lugar de poner gets, pusimos puts ya que este estaba guardado y por eso nos imprimio lo que se muestra en la pared de arriba, además como se puede observar se agrego */users/fp20alu06/Documents/

Programa 3.c

```

#include<stdio.h>

int main()
{
    FILE *archivo;
    char escribir[]="Escribir cadena en archivo mediante fputs.\n\tFacultad de
Ingeniería.\n";
    archivo = fopen("puts.txt", "r+");

    if (archivo != NULL)
    {
        printf("El archivo se abrió correctamente.\n");
        fputs (escribir, archivo);
        fclose(archivo);
    }
    else
    {
        printf("Error al abrir el archivo.\n");
        printf("El archivo no existe o no se tienen permisos de lectura.\n");
    }

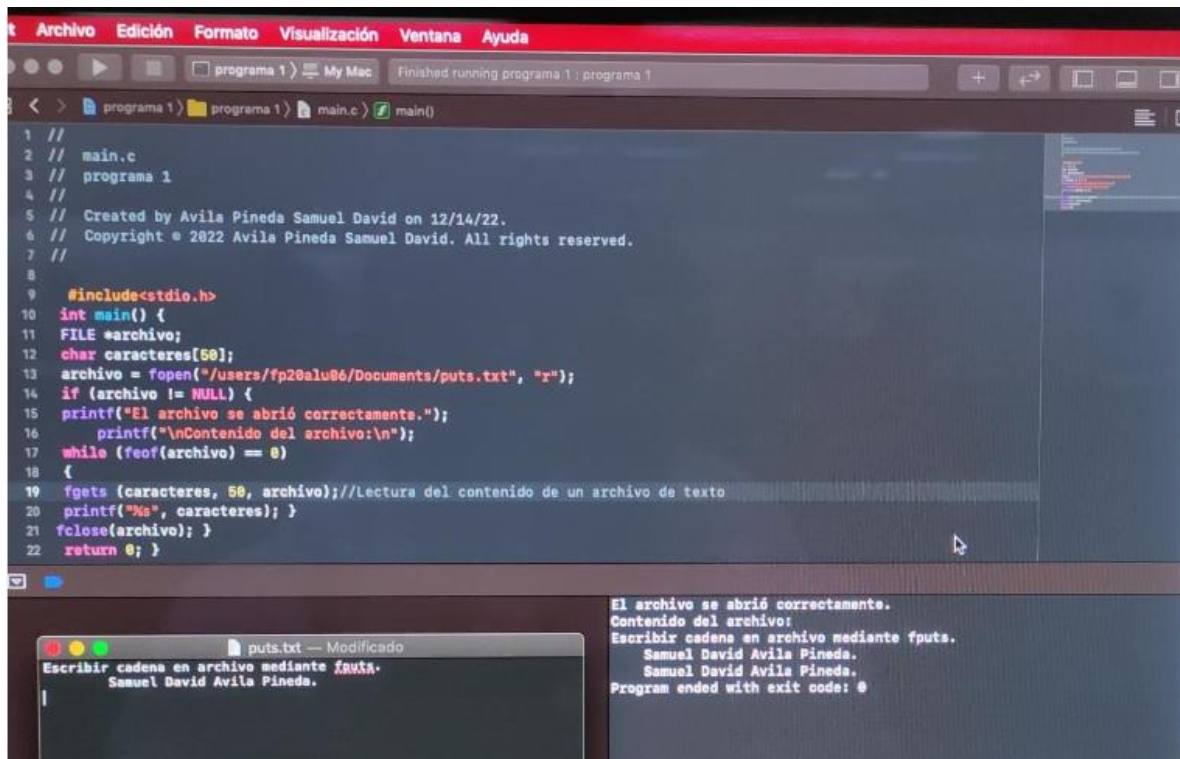
    return 0;
}

```

The screenshot shows a Mac OS X desktop environment. At the top, there is a window titled 'programa 1' containing the C code from the previous block. Below it, a terminal window shows the output of the program: 'El archivo se abrió correctamente.' followed by 'Program ended with exit code: 0'. In the bottom left corner, a file explorer window shows the contents of the 'puts.txt' file, which contains the text: 'Escribir cadena en archivo mediante fputs.' and 'Facultad de Ingeniería.'

Extra

Como actividad extra de los dos programas anteriores la profesora nos dijo que ahora nos imprimiera nuestro nombre.



```
1 //
2 // main.c
3 // programa 1
4 //
5 // Created by Avila Pineda Samuel David on 12/14/22.
6 // Copyright © 2022 Avila Pineda Samuel David. All rights reserved.
7 //
8
9 #include<stdio.h>
10 int main() {
11     FILE *archivo;
12     char caracteres[50];
13     archivo = fopen("/Users/fp20alu06/Documents/puts.txt", "r");
14     if (archivo != NULL) {
15         printf("El archivo se abrió correctamente.");
16         printf("\nContenido del archivo:\n");
17         while (feof(archivo) == 0)
18         {
19             fgets(caracteres, 50, archivo); //Lectura del contenido de un archivo de texto
20             printf("%s", caracteres); }
21         fclose(archivo); }
22     return 0; }
```

El archivo se abrió correctamente.
Contenido del archivo:
Escribir cadena en archivo mediante fputs.
Samuel David Avila Pineda.
Samuel David Avila Pineda.
Program ended with exit code: 0

Funciones fscanf y fprintf

Se comportan similarmente a printf() scanf(), con la diferencia que operan un archivo. Sus estructuras son:

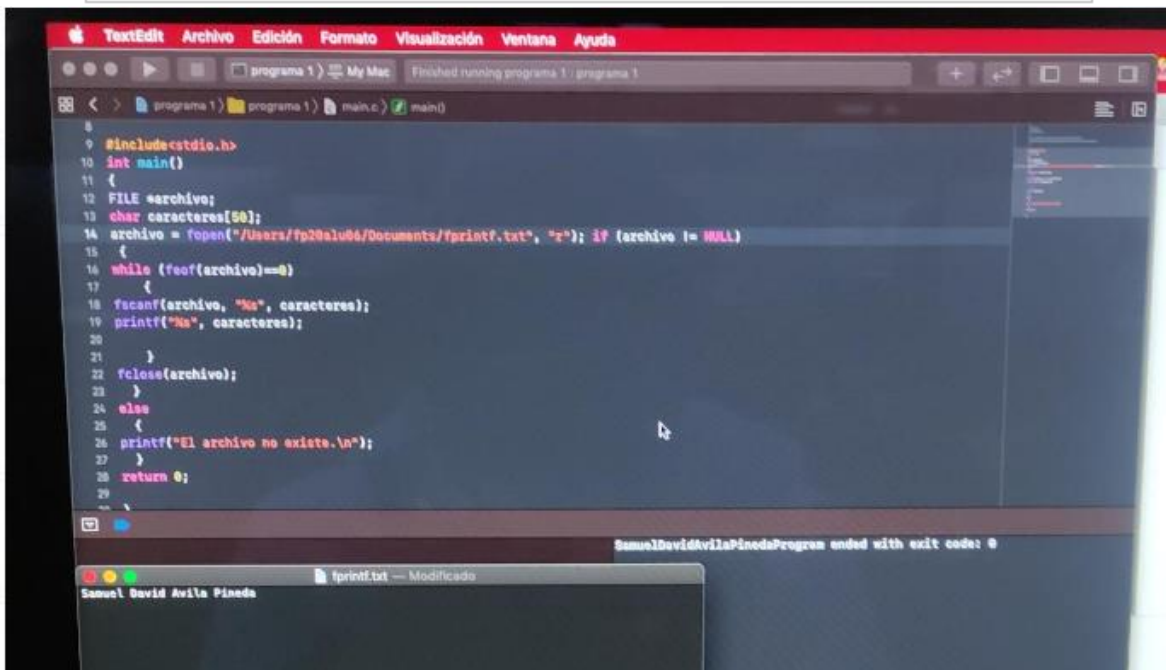
```
int fprintf(FILE *apArch, char *formato, ...);
int fscanf(FILE *apArch, char *formato, ...);
```

Donde apArch es un apuntador al archivo devuelto por una llamada a la función fopen().

Programa 4.c

```
#include<stdio.h>

int main()
{
    FILE *archivo;
    char caracteres[50];
    archivo = fopen("fscanf.txt", "r");
    if (archivo != NULL)
    {
        while (feof(archivo)==0)
        {
            fscanf(archivo, "%s", caracteres);
            printf("%s\n", caracteres);
        }
        fclose(archivo);
    }
    else
    {
        printf("El archivo no existe.\n");
    }
    return 0;
}
```

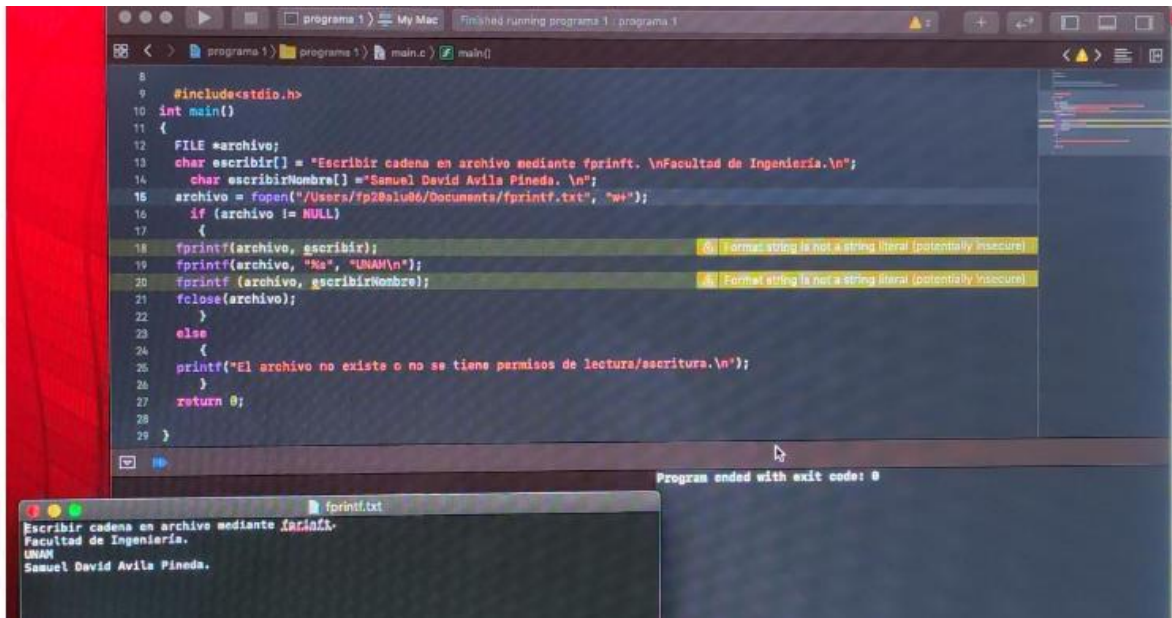


Tuve que volver a modificar todo, porque no me salía y eso que todo estaba bien.

Programa 5.c

```
#include<stdio.h>

int main()
{
    FILE *archivo;
    char escribir[] = "Escribir cadena en archivo mediante fprintf. \nFacultad de
Ingeniería.\n";
    archivo = fopen("fprintf.txt", "r+");
    if (archivo != NULL)
    {
        fprintf(archivo, escribir);
        fprintf(archivo, "%s", "UNAM\n");
        fclose(archivo);
    }
    else
    {
        printf("El archivo no existe o no se tiene permisos de lectura/escritura.\n");
    }
    return 0;
}
```



Funciones fread y fwrite

Son funciones que permiten trabajar con elementos de longitud conocida. fread permite leer uno o varios elementos de la misma longitud a partir de una dirección de memoria determinada

Su sintaxis es:

```
int fread(void *ap, size_t tam, size_t nelem, FILE *archivo)
```

fwrite permite escribir hacia un archivo uno o varios elementos de la misma longitud, almacenados a partir de una dirección de memoria determinada.

Su sintaxis es:

```
int fwrite(void *ap, size_t tam, size_t nelem, FILE *archivo)
```

Programa 6.c

C:\Users\KarinaGarciaMorales\OneDrive - Humansite, S.A de C.V\Documentos\GMK\DIAS_EQUIPO\KarinaSerie_III\Práctica12\ejercicio06.cpp - Dev-C++ 5.3.0.4

File Edit Search View Project Execute Debug Tools CVS Window Help

```
(globals)
ejercicio06.cpp
1 #include <stdio.h>
2 int main(int argc, char **argv)
3 {
4     FILE *ap;
5     unsigned char buffer[2048]; // Buffer de 2 Kbytes
6     int bytesLeidos;
7     // Si no se ejecuta el programa correctamente
8     if(argc < 2)
9     {
10         printf("Ejecutar el programa de la siguiente manera: \n\tnombre_\tprograma nombre_archivo\n");
11         return 1;
12     }
13     // Se abre el archivo de entrada en modo lectura y binario
14     ap = fopen(argv[1], "rb");
15     if(!ap)
16     {
17         printf("El archivo %s no existe o no se puede abrir", argv[1]);
18         return 1;
19     }
20     while(bytesLeidos = fread(buffer, 1, 2048, ap))
21     {
22         printf("%s", buffer);
23     }
24     fclose(ap);
25     return 0;
26 }
27
```

Compiler Resources Compile Log Debug Find Results

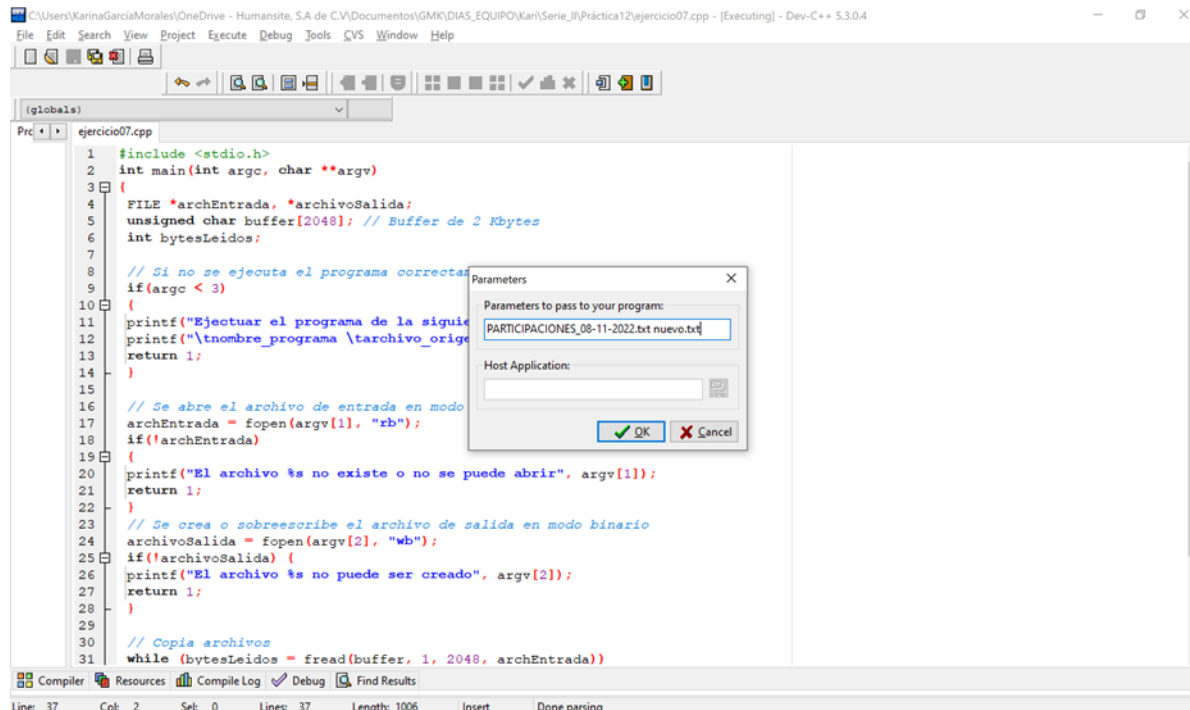
line: 19 Col: 3 Sel: 0 Lines: 27 Length: 623 Insert Done parsing

Después se coloca el nombre del archivo a leer y posteriormente se compila y ejecuta

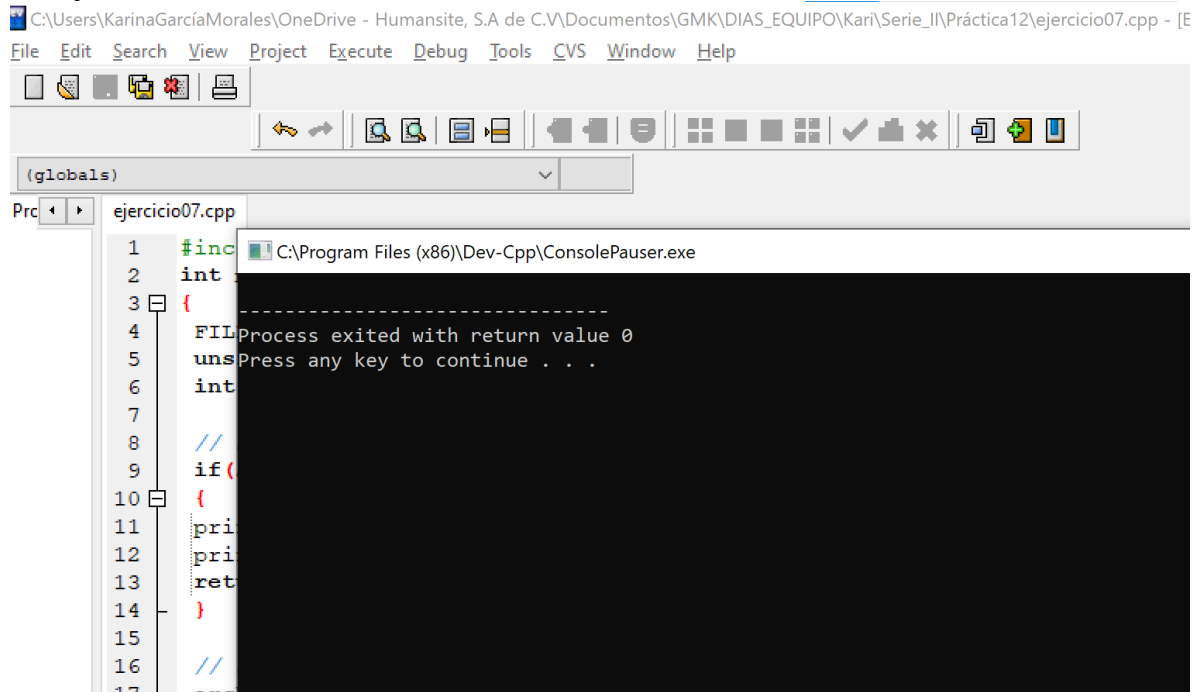
```
C:\Users\KarinaGarciaMorales\OneDrive - Humansite, S.A de C.V\Documentos\GMK\DIAS_EQUIPO\KarinaSerie_III\Práctica12\ejercicio06.cpp - Dev-C++ 5.3.0.4
File Edit Search View Project Execute Debug Tools CVS Window Help
(globals)
ejercicio06.cpp
1 #include <stdio.h>
2 int main(int argc, char **argv)
3 {
4     FILE *ap;
5     unsigned char buffer[2048]; // Buffer de 2 Kbytes
6     int bytesLeidos;
7     // Si no se ejecuta el programa correctamente
8     if(argc < 2)
9     {
10         printf("Ejecutar el programa de la siguiente manera: \n\tnombre_\tprograma nombre_archivo\n");
11         return 1;
12     }
13     // Se abre el archivo de entrada en modo lectura y binario
14     ap = fopen(argv[1], "rb");
15     if(!ap)
16     {
17         printf("El archivo %s no existe o no se puede abrir", argv[1]);
18         return 1;
19     }
20     while(bytesLeidos = fread(buffer, 1, 2048, ap))
21     {
22         printf("%s", buffer);
23     }
24     fclose(ap);
25     return 0;
26 }
27
```

C:\Program Files (x86)\Dev-Cpp\Compiler\Compiler.exe
Ejemplo de archivo para practica
Este archivo se visualiza en terminal
Karina
Morales
K
.....
Process exited with return value 0
Press any key to continue . . .

Programa 7.c



Se ejecuta



Conclusiones

Esta práctica, fue una de las que más se complicó, entendí lo teórico, como lo de las funciones fopen y fclose, una es para abrir archivos y la otra para cerrar los archivos abiertos, sin embargo, a la hora de llevarlo a cabo se me fue complicando, pero eso no me impidió que no lo hiciera. Se utilizó algunos modos de apertura de un archivo, vimos que la función fprintf es similar a la printf y el fscanf es similar al scanf.

Referencias

1. El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Edu
2. <http://lcp02.fi-b.unam.mx/>