|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Ing. Rodríguez Espino Claudia |
| *Asignatura:* | Fundamentos de programación |
| *Grupo:* | 1122, Bloque: 139 |
| *No de Práctica(s):* | Décimo tercera |
| *Integrante(s):* | Fonseca Ramírez Gadiel Moisés |
|  |  |
|  |  |
| *Semestre:* | 2018-1 |
| *Fecha de entrega:* | il est vendredi le 17 novembre 2017 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Guía práctica de estudio 13: Lectura y escritura de datos**

**Objetivo:**

Elaborar 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. +

**Actividades realizadas en clase**

En esta clase, tras dar lectura a la práctica, la profesora fue explicando los conceptos del uso de archivos .txt en C con los nuevos comandos que estábamos viendo. Cabe señalar que cuando los archivos .txt se crean, se guardan en la misma carpeta en la que se está corriendo el ejecutable del programa.

Para esta práctica de revisaron, probaron y se hicieron pruebas con los comandos para los programas:

* Abrir un archivo en modo de lectura:

Este programa sólo marca que el arhivo se abrió correctamente, este siempre marcará que abrió correctamente, puesto que el comando ‘w’ crea el archivo en caso de que este no exista.

#include<stdio.h>

int main() {

FILE \*archivo;

archivo = fopen("archivo.txt", "w");

if (archivo != NULL) {

printf("El archivo se abrió correctamente.\n");

int res = fclose(archivo);

printf("fclose = %d\n", res);

} else {

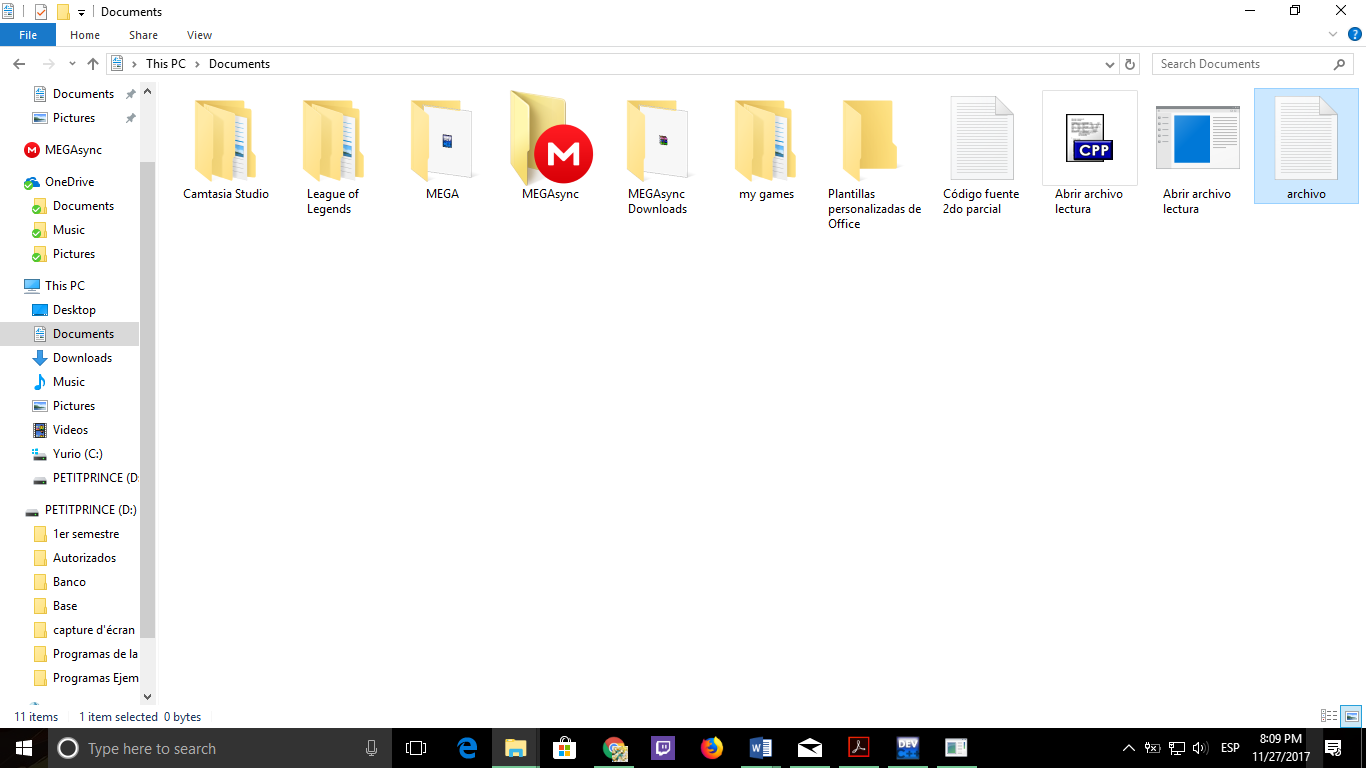
printf("Error al abrir el archivo.\n");

printf("El archivo no existe o no se tienen permisos de lectura.\n");

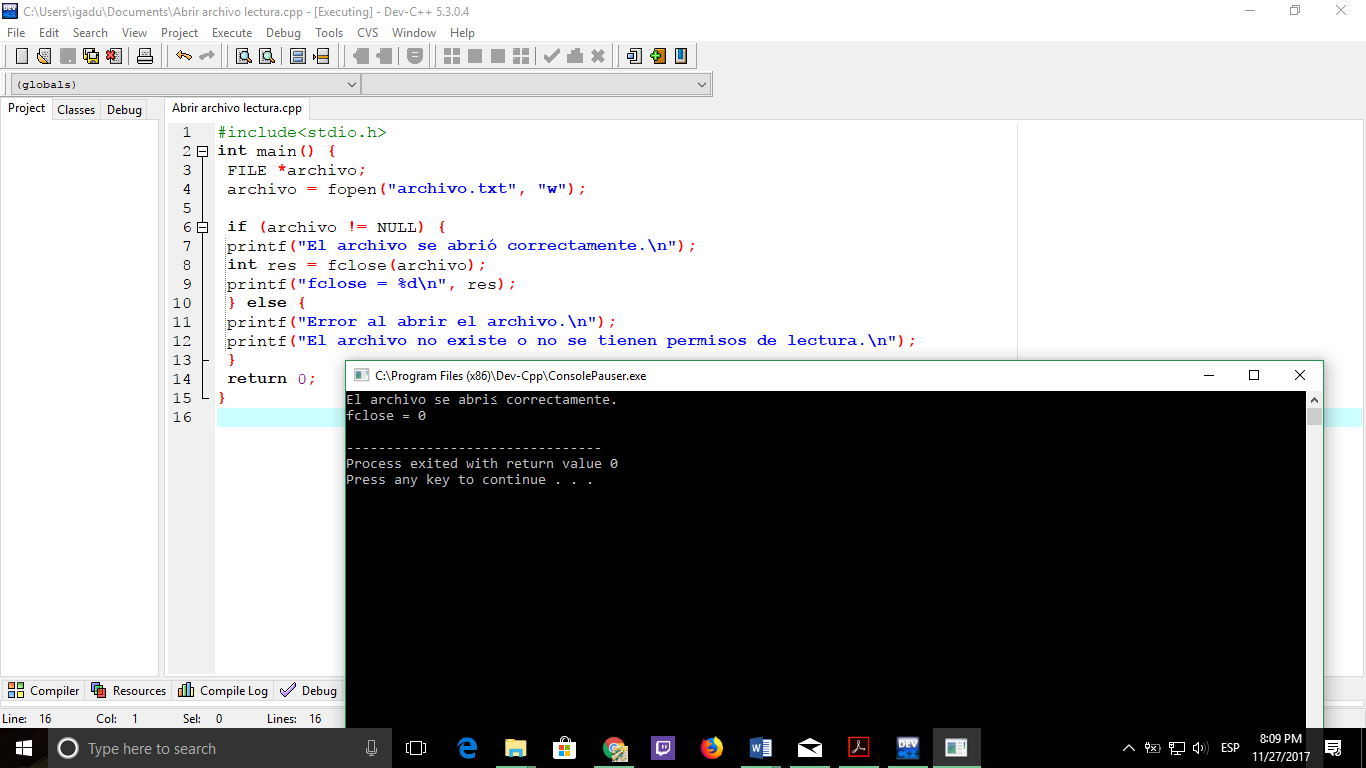
}

return 0;

}



El archivo llamado “Archivo” se crea en la misma carpeta que está el ejecutable del programa.



* 2. Leer el contenido de un archivo .txt usando “fgets”:

#include<stdio.h>

#include <stdlib.h>

int main() {

FILE \*archivo;

char caracteres[100];

archivo = fopen("archivo.txt", "r+");

if (archivo == NULL)

exit(1);

printf("El contenido del archivo de 'archivo' es \n\n");

while (feof(archivo) == 0) {

fgets (caracteres, 100, archivo);

printf("%s", caracteres);

}

puts("\n");

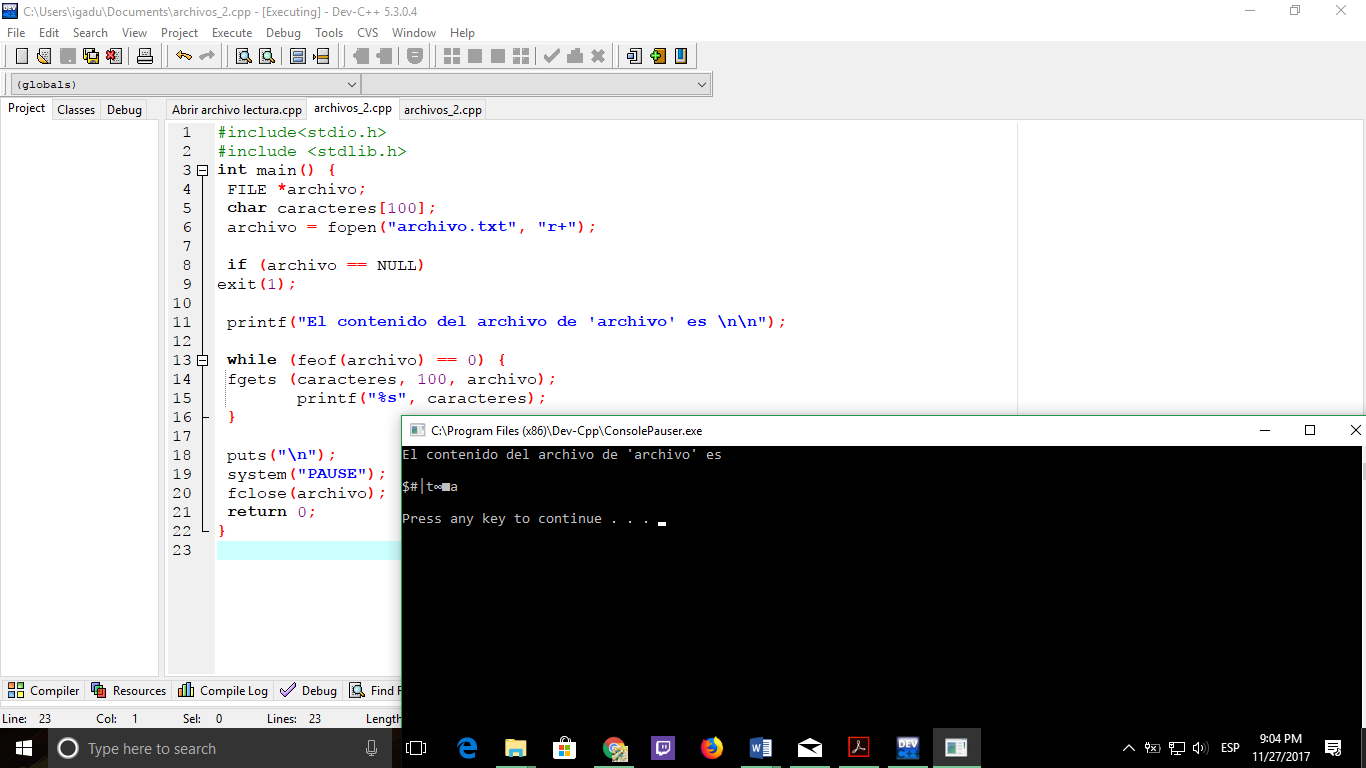
system("PAUSE");

fclose(archivo);

return 0;

}

Aunque el archivo está en blancpo verdaderamente, este presenta basura, la cuál es visible desde la terminal de C, ya que no fue limpiado.



* 3. Escribir oraciones (cadenas de texto) en un archivo:

#include<stdio.h>

/\*

Este programa permite escribir una cadena dentro de un archivo, de ser

posible, a través de la función fputs.

\*/

int main() {

FILE \*archivo;

char escribir[] = "Yo te amo Elizabeth \n Pero no se vale esto que me has hecho";

archivo = fopen("archivo.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");

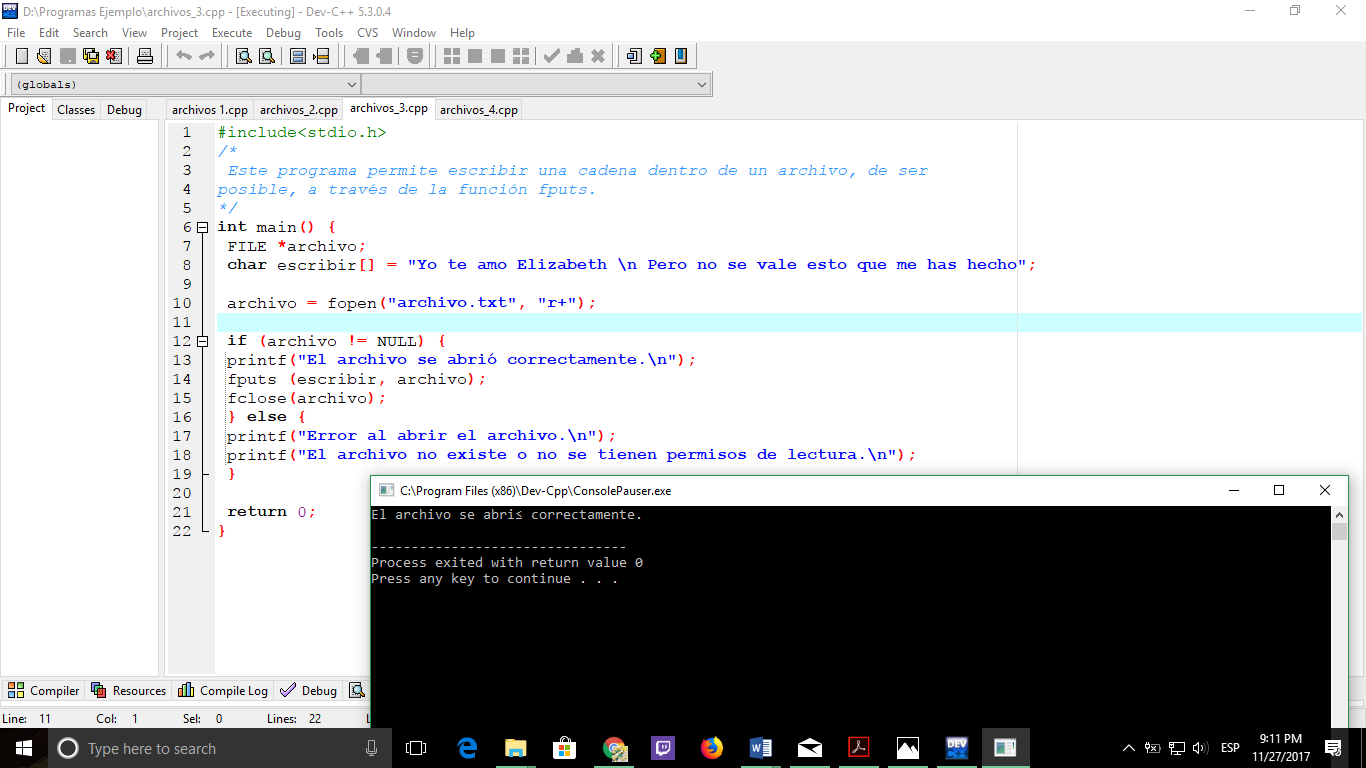
printf("El archivo no existe o no se tienen permisos de lectura.\n");

}

return 0;

}

Cabe señalar que si se vuelve a escribir en el archivo usando este programa, se borra el contenido anterior del archivo.



* 4. Leer el contenido de un archivo, pero usando “fscanf”

#include<stdio.h>

/\*

Este programa permite leer el contenido de un archivo,

de ser posible, a través de la función fscanf.

\*/

int main() {

FILE \*archivo;

char caracteres[50];

archivo = fopen("archivo.txt", "r");

if (archivo != NULL) {

while (feof(archivo)==0){

fscanf(archivo, "%s", caracteres);

printf("%s ", caracteres);

}

fclose(archivo);

} else {

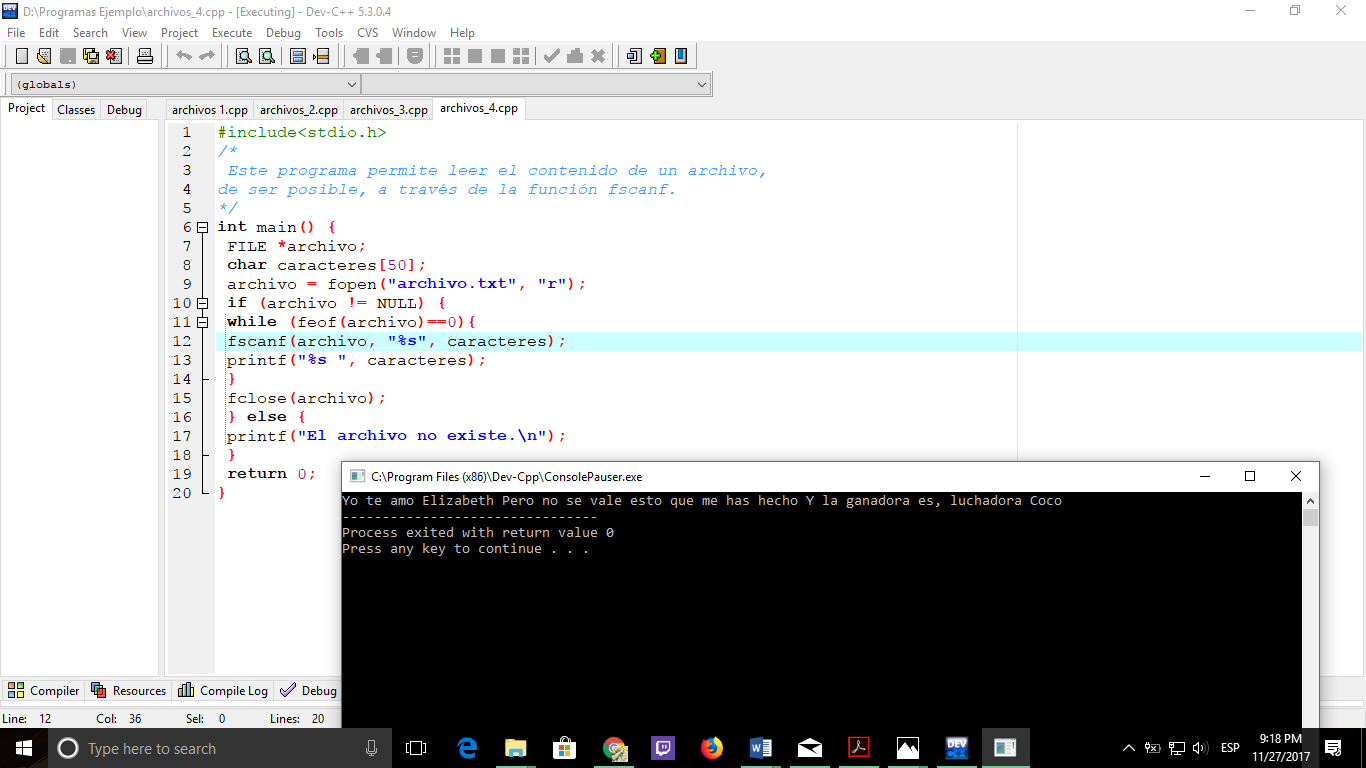
printf("El archivo no existe.\n");

}

return 0;

}

Cabe señalar que al usar”fscanf”, el texto del archivo se escribe de forma continua en una sola línea; mientras que al usar el “fgets” el contenido del archivo se escribe en la pantalla de la misma forma en la que está en el archivo, manteniendo la sangría, renglones, párrafos y espacios.



Conclusión:

Esta práctica nos permitió aprender acerca del como funcionan los comandos para manejo de archivos en C, lo cual es bastante útil para poder manejar la información como lo puede ser en el caso de un expediente médico, en el cuál se guardan los daros de los pacientes dentro de un archivo, y al ser consultados, son mostrados en pantalla mediante una terminal.