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

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| Profesor: | García Morales Karina |
| Asignatura: | Fundamentos de Programación |
| Grupo: | 1121 |
| No de Práctica(s): | 13 |
| Integrante(s): | Tinoco Ramos Isaac Ricardo |
|  |  |
|  |  |
|  |  |
| No. de Equipo de cómputo empleado: |  |
| Semestre: | Primero |
| Fecha de entrega: | 20/11/2018 |
| Observaciones: |  |
|  |  |

Calificación\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

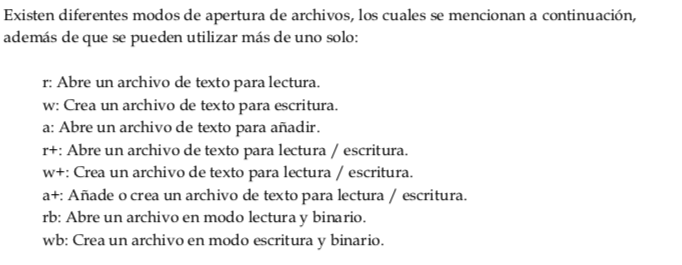
***Lectura y escritura de datos***

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

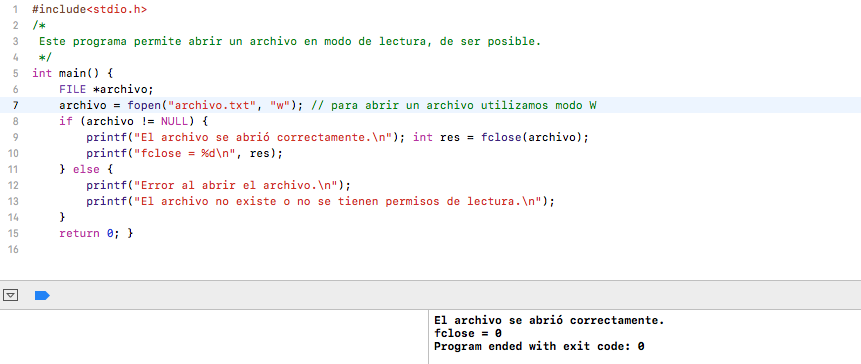
**Objetivo**: Elaborar programas en lenguaje C que requieran el uso de archivos de texto plano en la resolución de problemas, entendiendo a los archivos como un elemento de almacenamiento secundario.

**Introducción:**

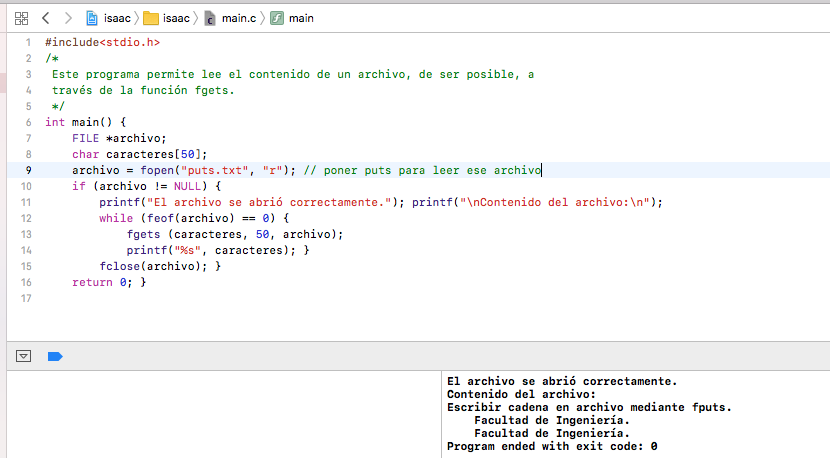
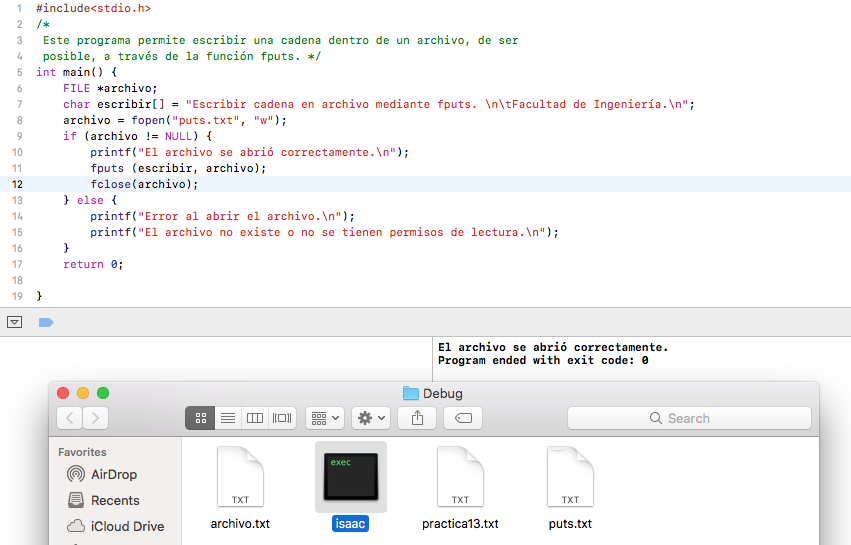
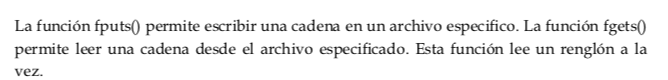
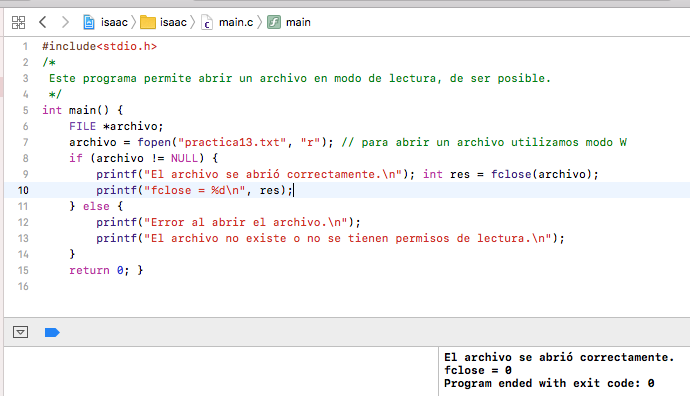
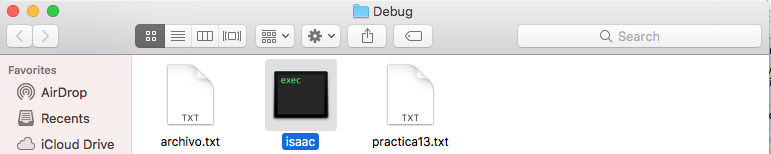
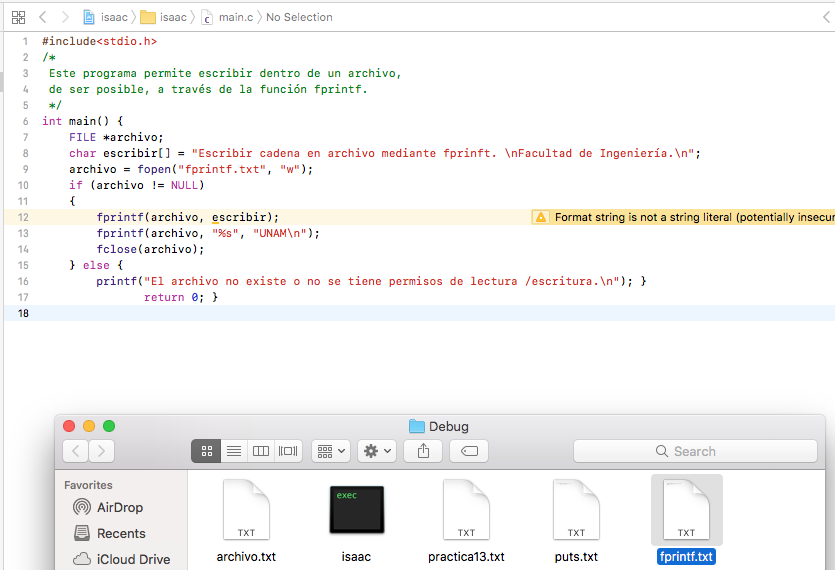
Un archivo es un conjunto de datos estructurados en una colección de entidades elementales o básicas denominadas registros que son del mismo tipo, pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Lenguaje C permite manejar la entrada y la salida de datos desde o hacia un archivo, respectivamente, a través del uso de la biblioteca de funciones de la cabecera stdio.h.

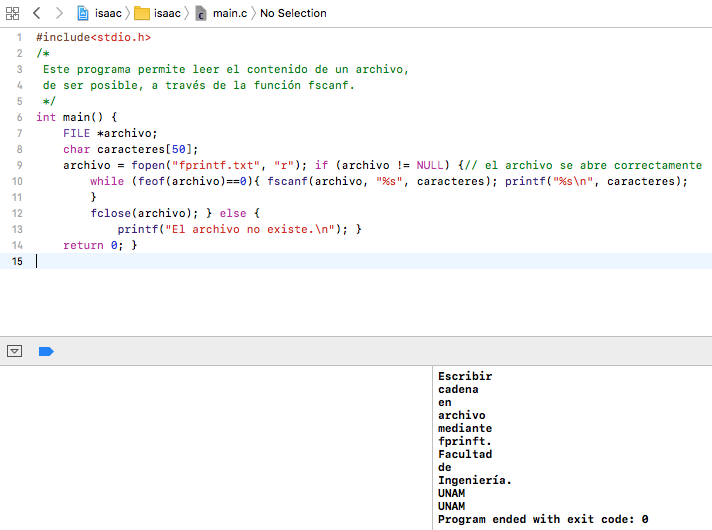
Modos de aperturas

**Desarrollo**

Para poder crear el archivo utilizamos el modo “w”

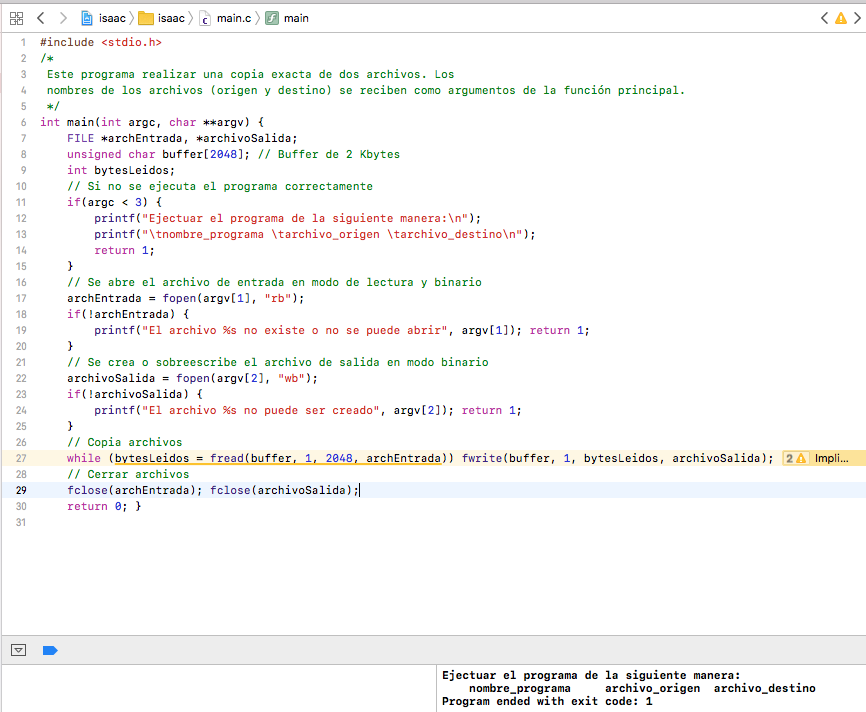
Creamos el archivo “archivo.txt”

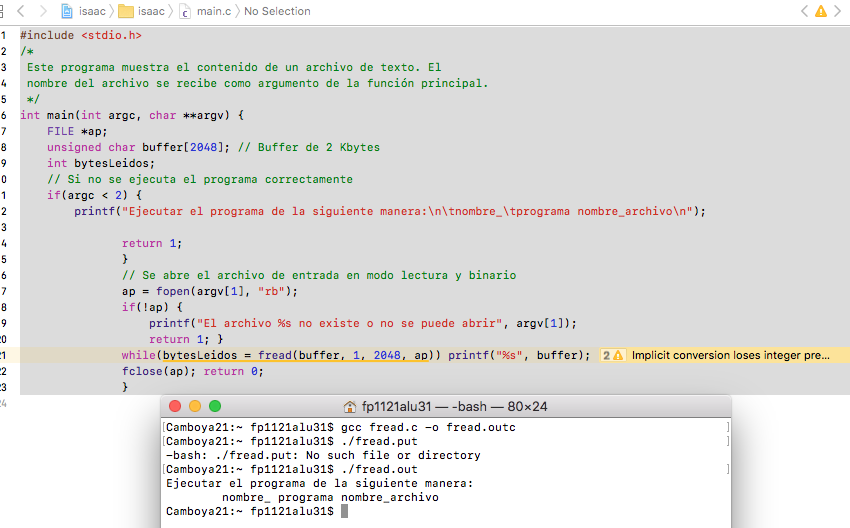
La función feof nos permite saber si queda algún carácter más que leer en el archivo. En lugar de leer cadenas de caracteres, como en el ejemplo de hoy, tendrás que leer caracteres con el formato C y devuelve 0 si no se ha alcanzado el final del archivo que se pasa como argumento. Esta función resulta muy útil en combinación con un bucle para leer archivos de forma flexible

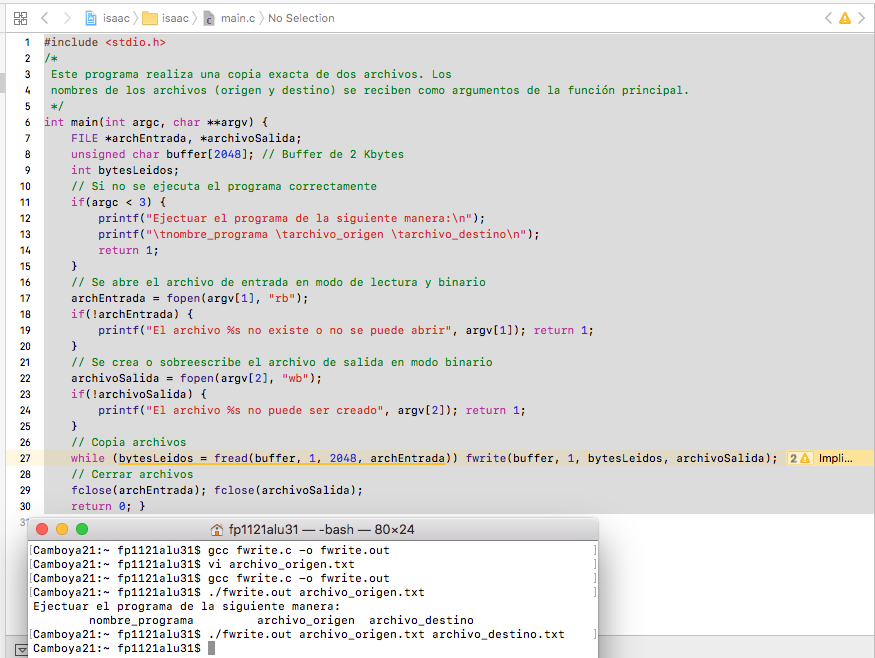


¿Qué hace cada función?

* La función fopen() abre una secuencia para que pueda ser utilizada y la asocia a un archivo. Su estructura es la siguiente: \*FILE fopen(char \*nombre\_archivo, char \*modo);
* La función fclose() cierra una secuencia que fue abierta mediante una llamada a fopen(). Escribe la información que se encuentre en el buffer al disco y realiza un cierre formal del archivo a nivel del sistema operativo.
* Las funciones fgets() y fputs() pueden leer y escribir, respectivamente, cadenas sobre los archivos. Las firmas de estas funciones son, respectivamente: char \*fgets(char \*buffer, int tamaño, FILE \*apArch); char \*fputs(char \*buffer, FILE \*apArch); Las funciones fprintf() y fscanf() se comportan exactamente como printf() (imprimir) y scanf() (leer), excepto que operan sobre archivo. Sus estructuras son: int fprintf(FILE \*apArch, char \*formato, ...); int fscanf(FILE \*apArch, char \*formato, ...);
* 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 (apuntador).

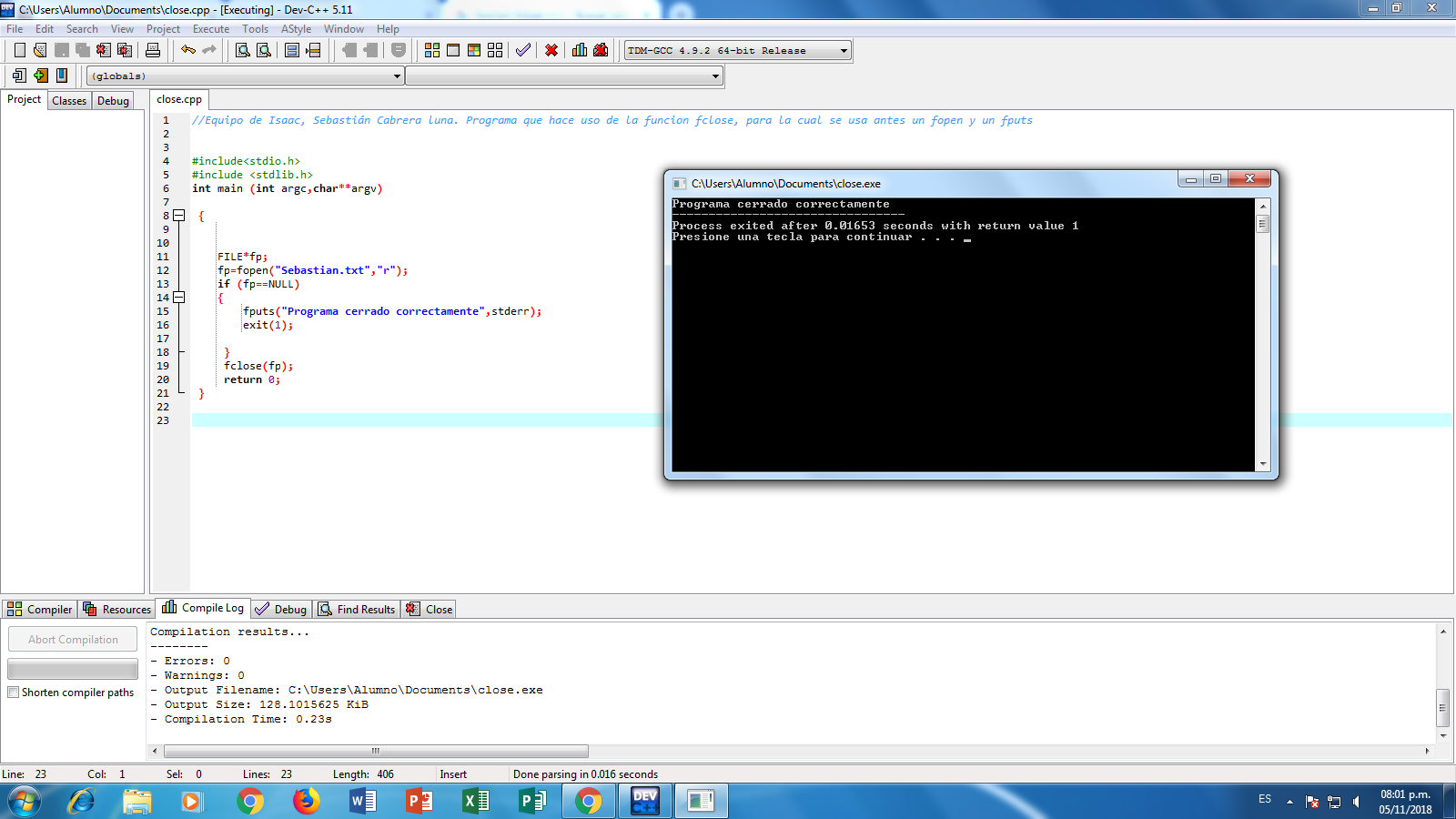


Compilamos y ejecutamos para corroborar que el programa funcione. 

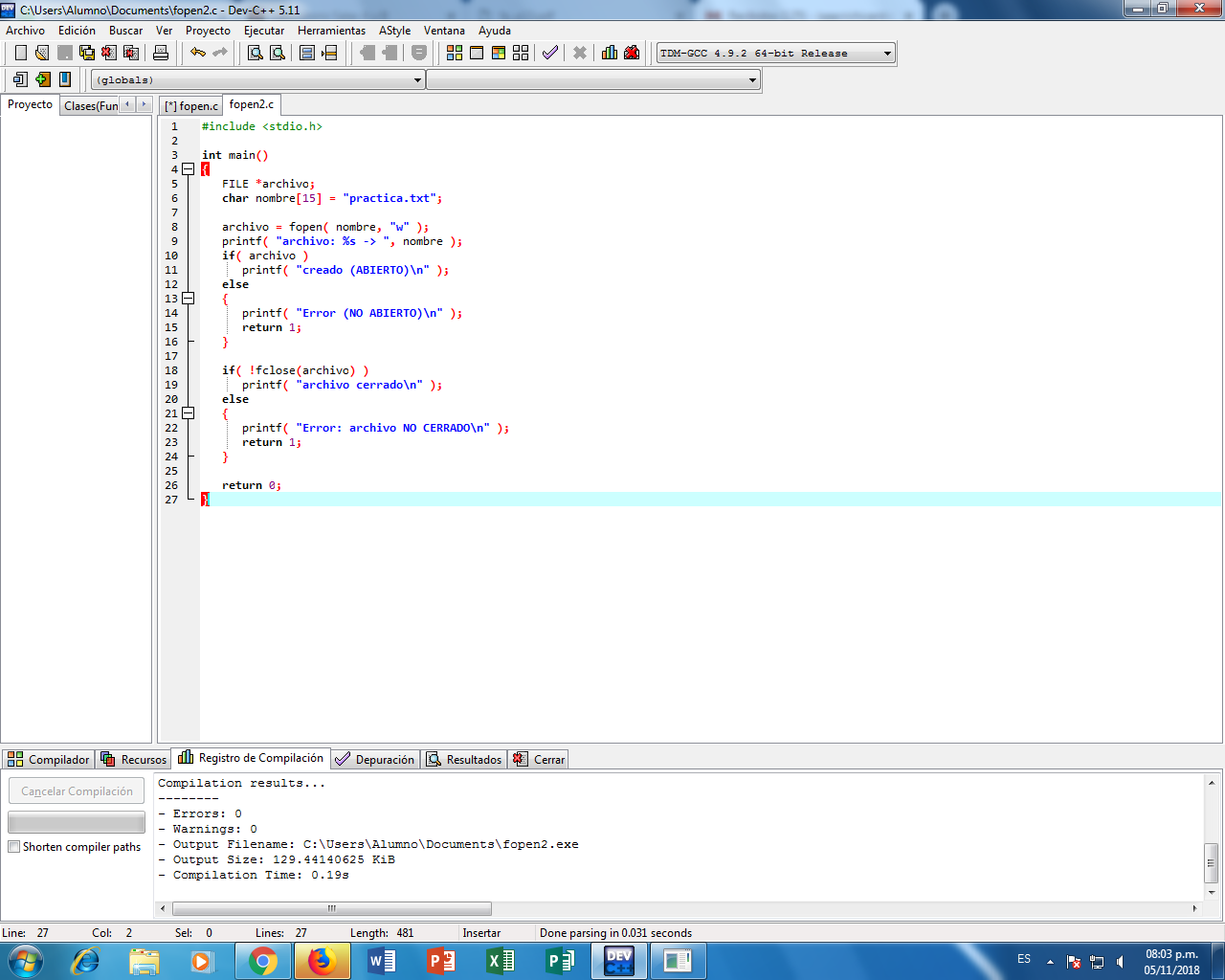


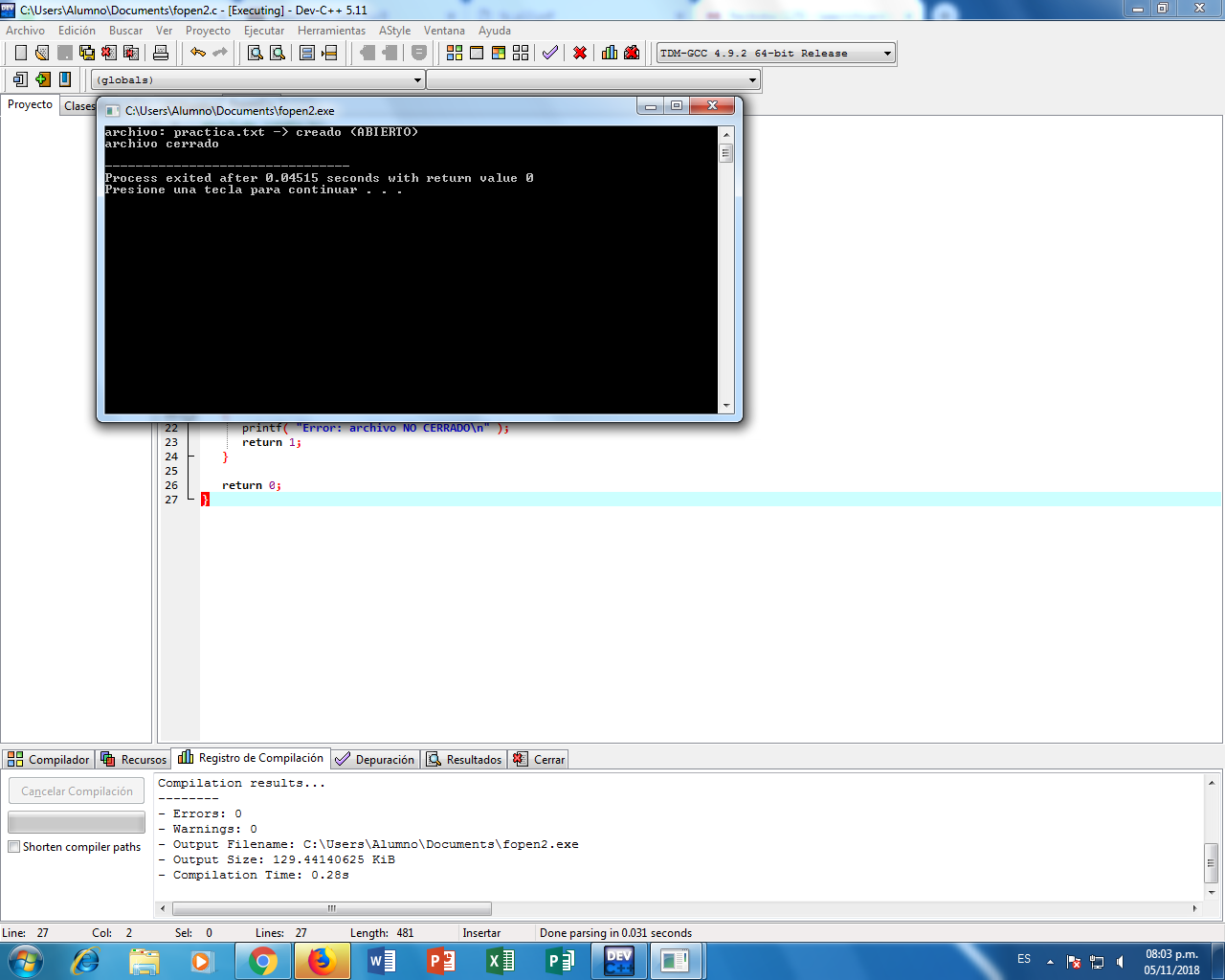
**Ejercicios:**

**Función fclose**

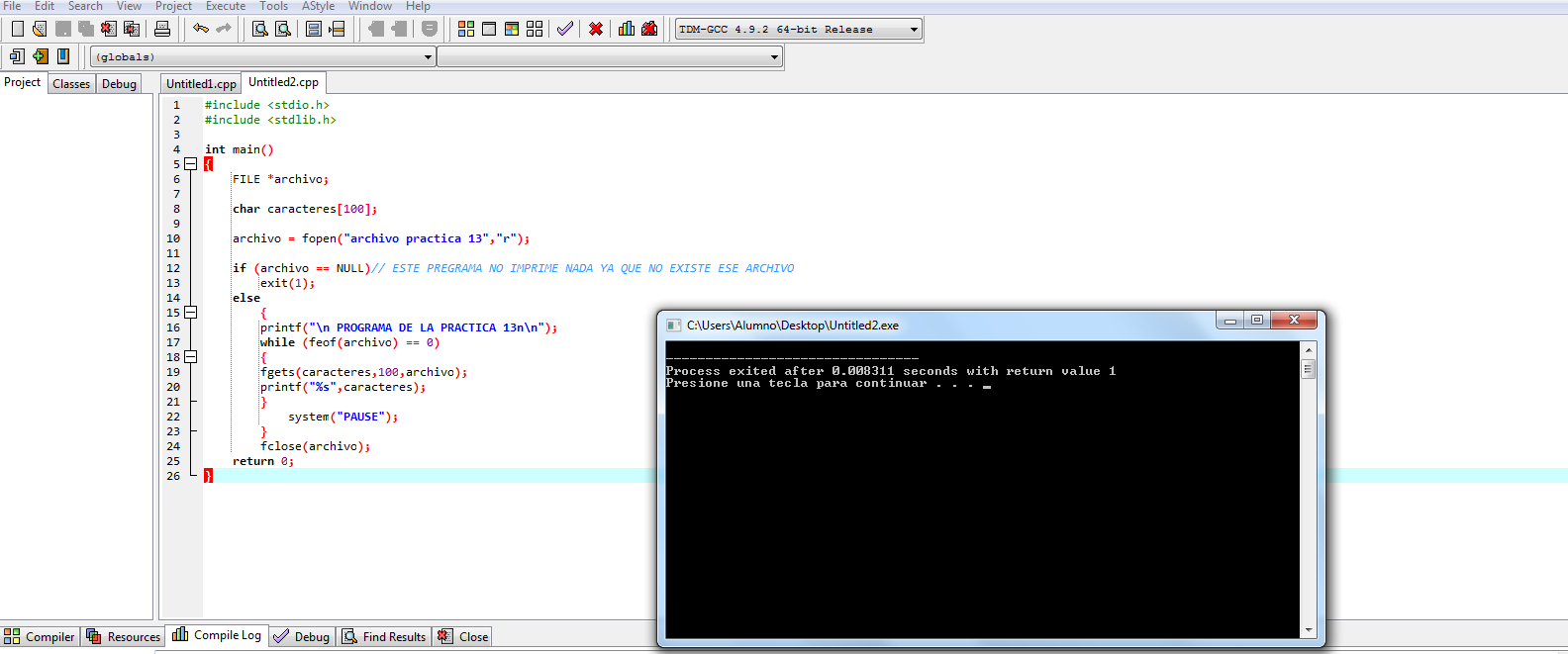


**Función fopen**

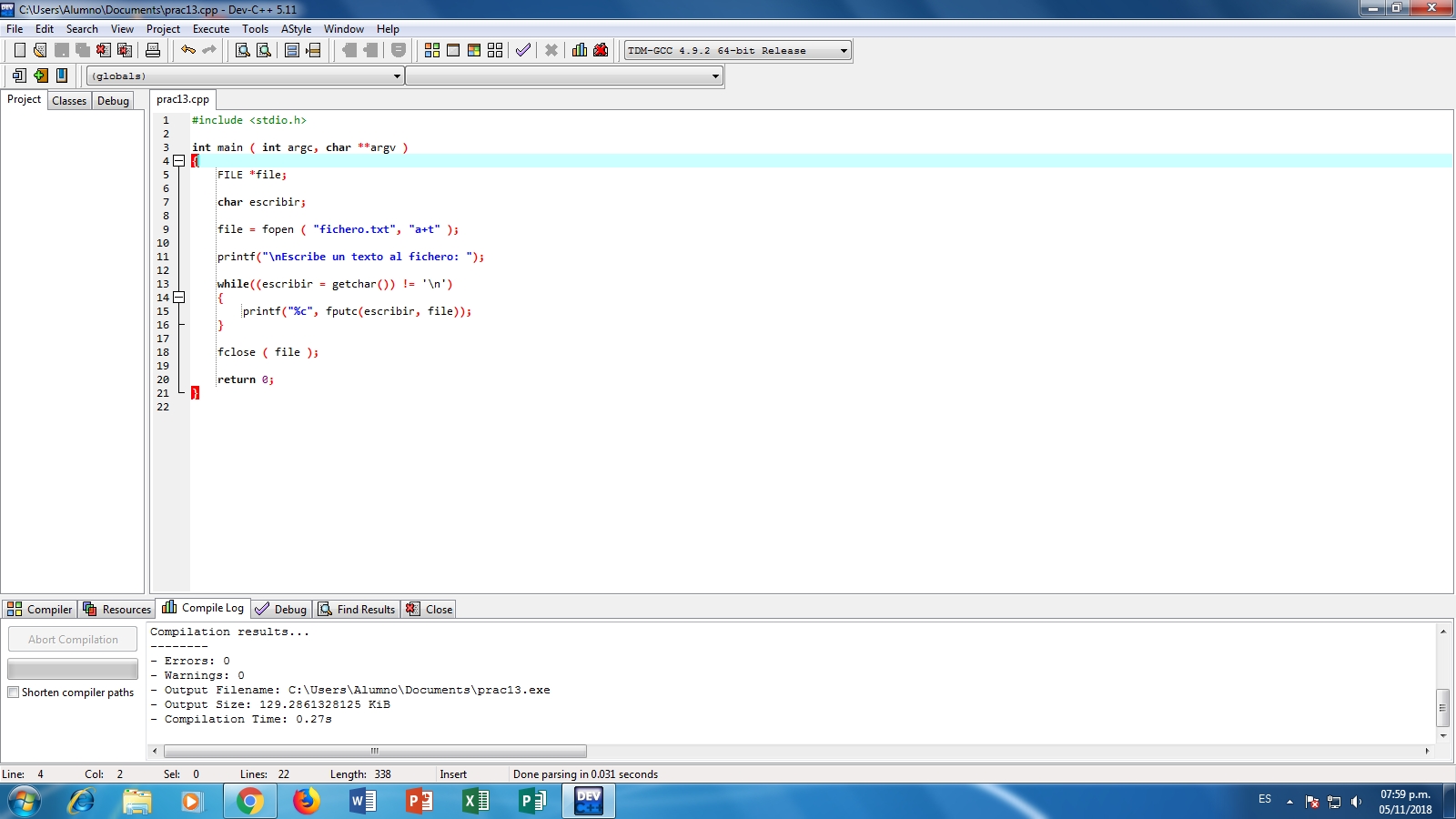




**Función fgets**



**Función fputs**



**Función fread**

//fread permite leer uno o varios elementos de la misma longitud a partir de una dirección //de memoria determinada (apuntador). //El valor de retorno es el número de elementos (bytes) leídos

#include <stdio.h>

/\*

Este programa muestra el contenido de un archivo de texto. El nombre del archivo se recibe como argumento de la función principal.

\*/

int main(int argc, char \*\*argv) {

FILE \*ap; unsigned char buffer[2048]; // Buffer de 2 Kbytes int bytesLeidos;

// Si no se ejecuta el programa correctamente

if(argc > 2) {

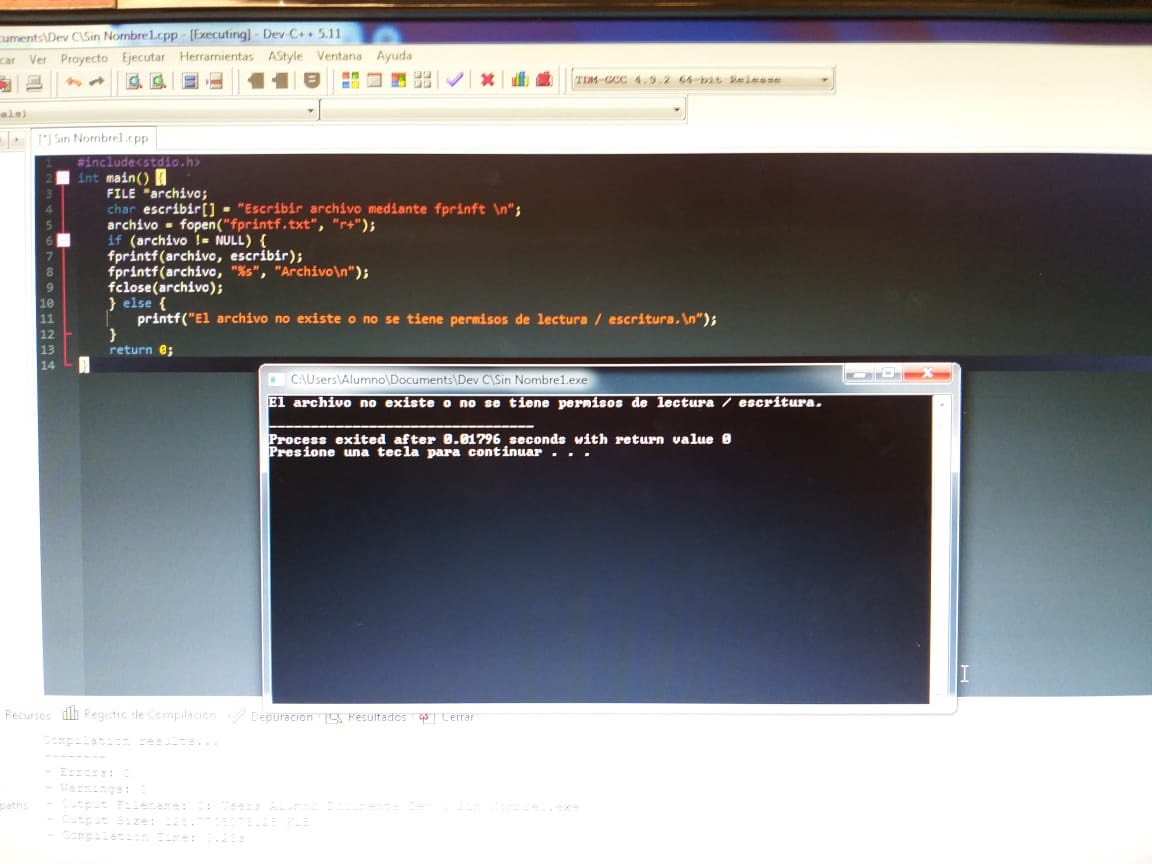
printf(bytesLeidos = fread(buffer, 1, 2048, ap));

return 1; }

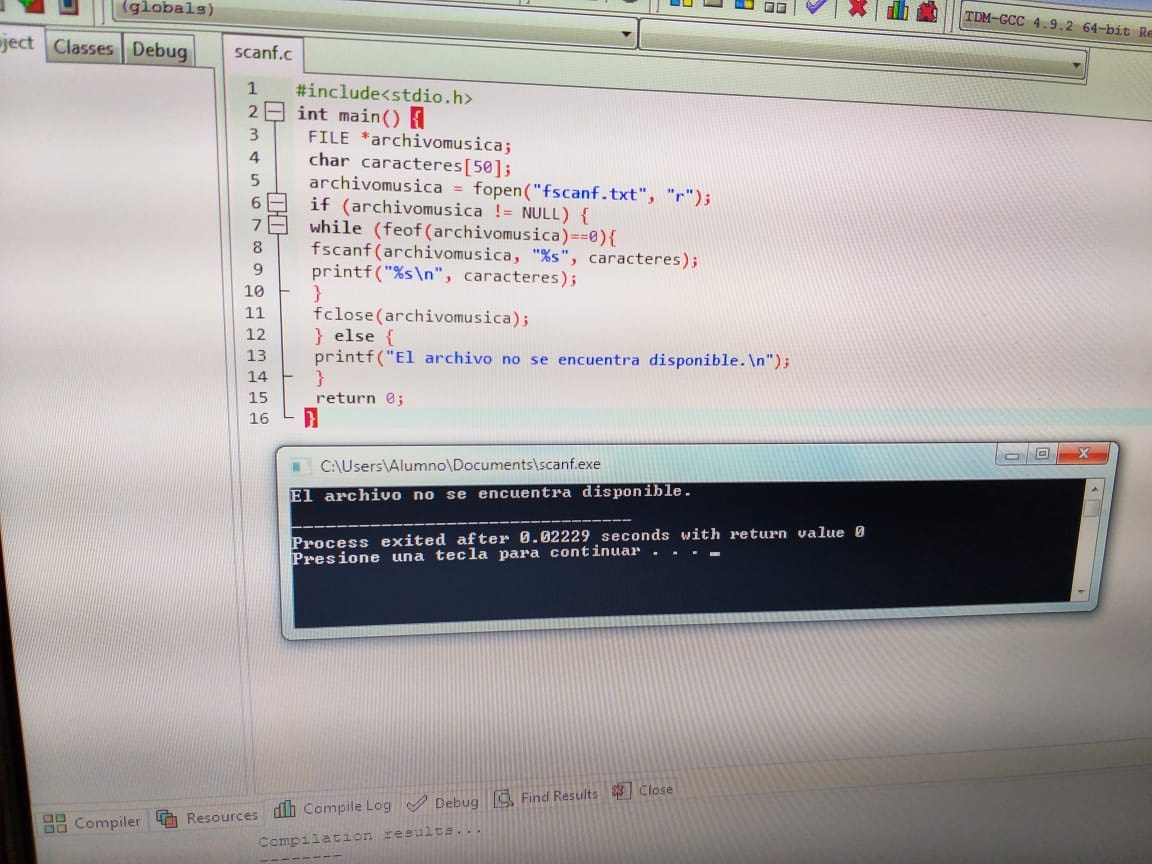
fclose(ap);

return 0; }

**Función fprintf**



**Funcion fscanf**



**Conclusion:**

Elaboramos 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, haciendo uso de todo lo que hemos visto en el año y en especial las funciones que vimos en los últimos temas. Usando datos estructurados en una colección de entidades elementales o básicas denominadas registros que son del mismo tipo, pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso, en lenguaje C que nos permite manejar la entrada y la salida de datos desde o hacia un archivo, respectivamente, a través del uso de la biblioteca de funciones.