|  |
| --- |
| Laboratorio de Computación  Salas A y B |
|  |

|  |  |
| --- | --- |
| Profesor: | GARCIA MORALES KARINA ING. |
| Asignatura: | FUNDAMENTOS DE PROGRAMACION |
| Grupo: | 1121 |
| No de Práctica(s): | 13 |
| Integrante(s): | JOSE DANIEL CALLEJAS SANDOVAL |
|  |  |
|  |  |
|  |  |
| No. de Equipo de cómputo empleado: | 25 |
| Semestre: | 1 |
| Fecha de entrega: | 20-11-2018 |
| 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.

**Desarrollo:**

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*);

Argumentos

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

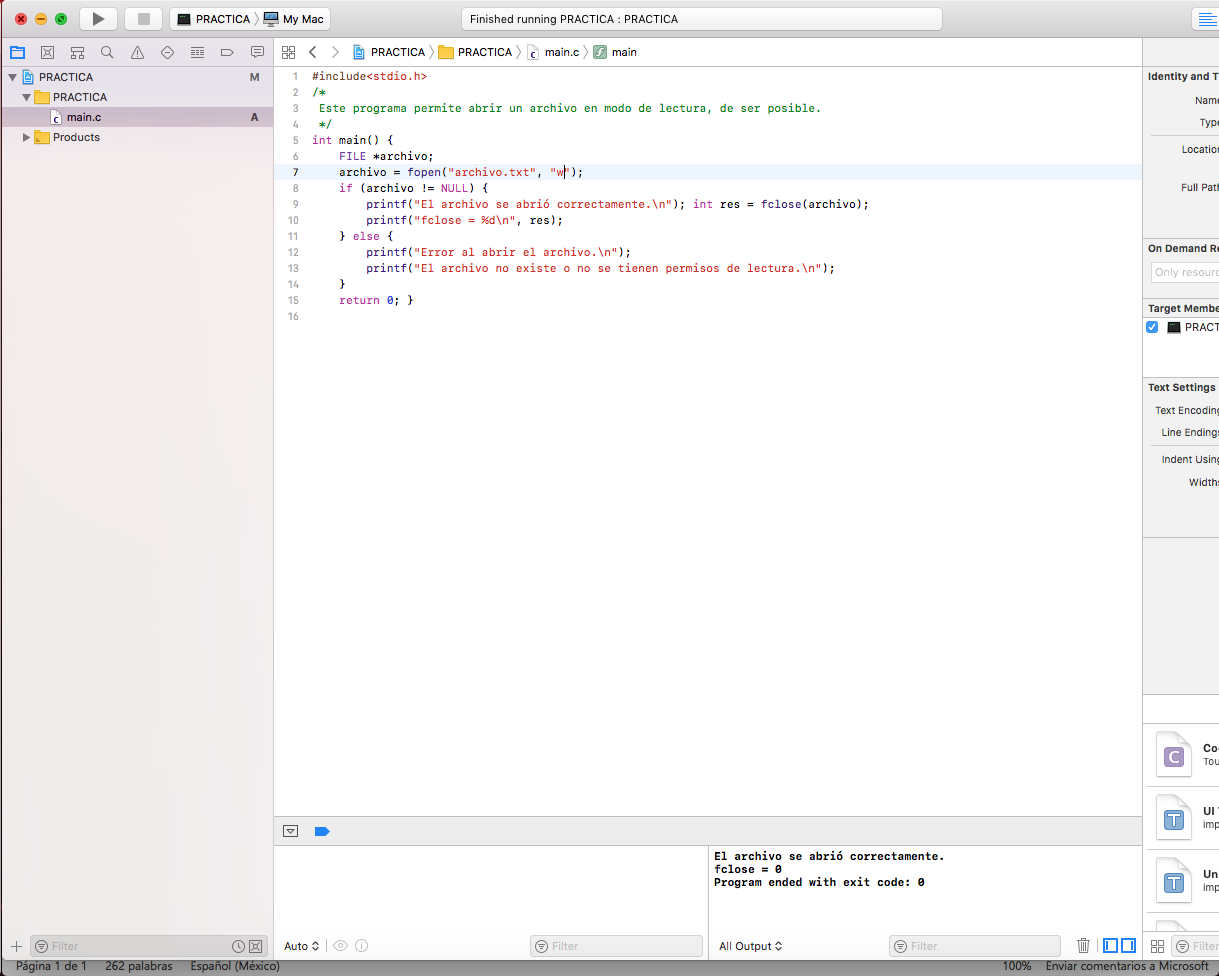
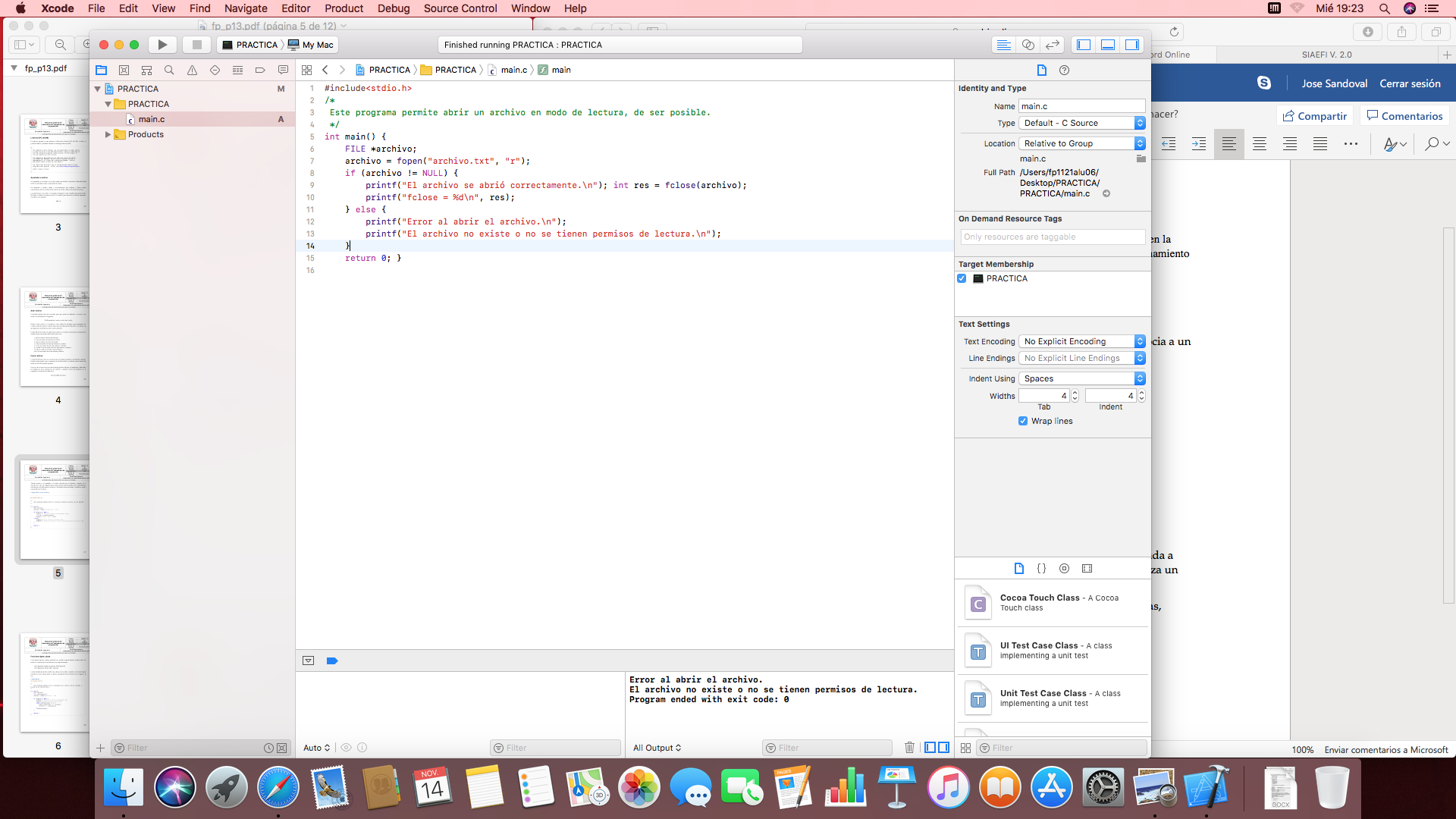
rb: Abre un archivo en modo lectura y binario.  
wb: Crea un archivo en modo escritura y binario.

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.

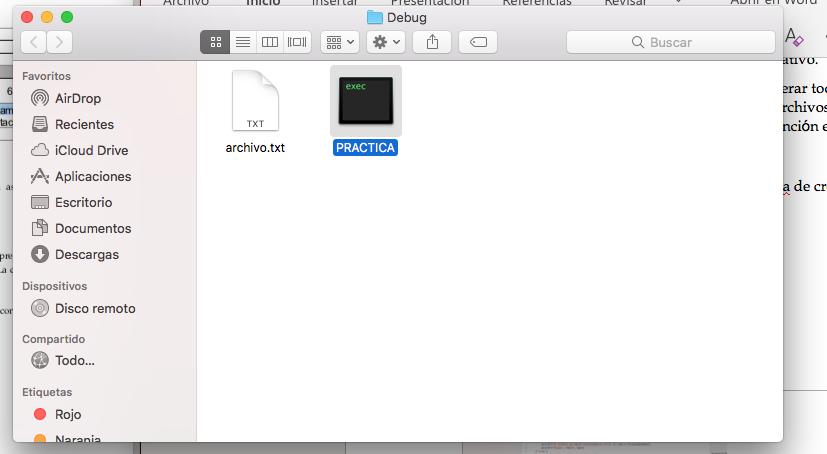
Un error en el cierre de una secuencia puede generar todo tipo de problemas, incluyendo la pérdida de datos, destrucción de archivos y posibles errores intermitentes en el programa. La firma de esta función es:

int fclose(FILE \*apArch);

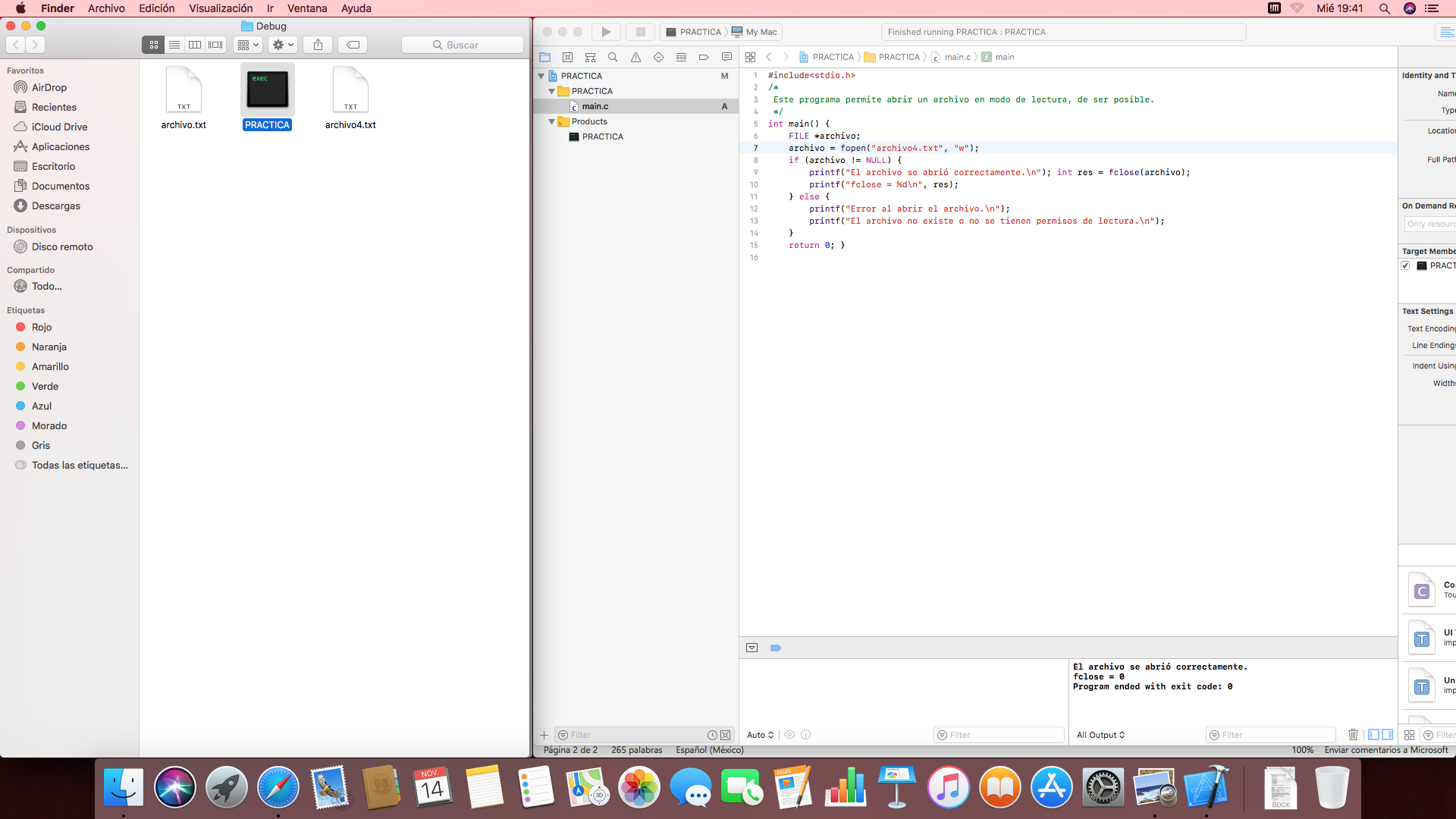
El error principal de este ejercicio fue que se debía de crear un archivo, por lo que modificamos el argumento a “w”.



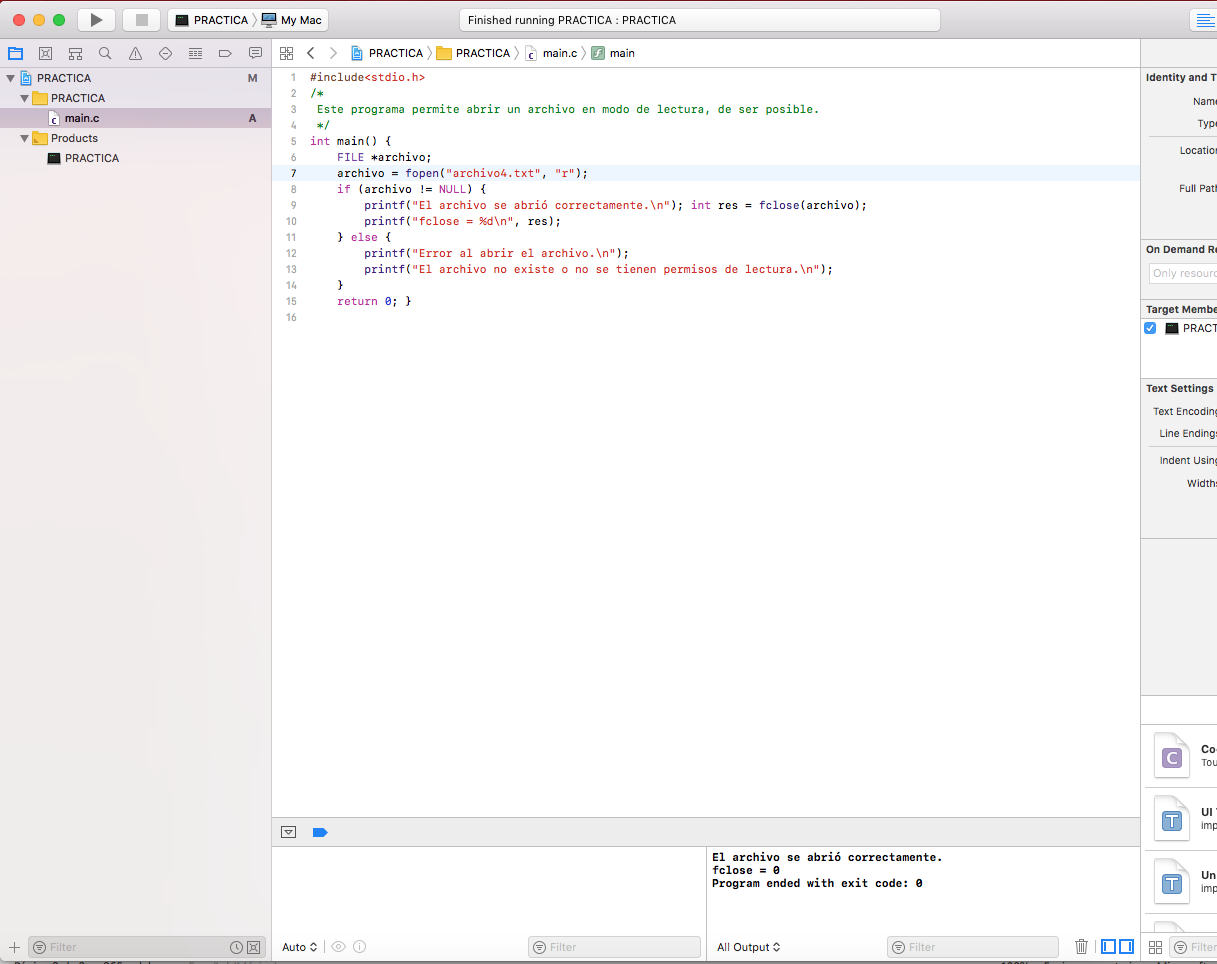
Validamos el archivo.



Creamos un nuevo archivo y observamos que ya fueron creados.



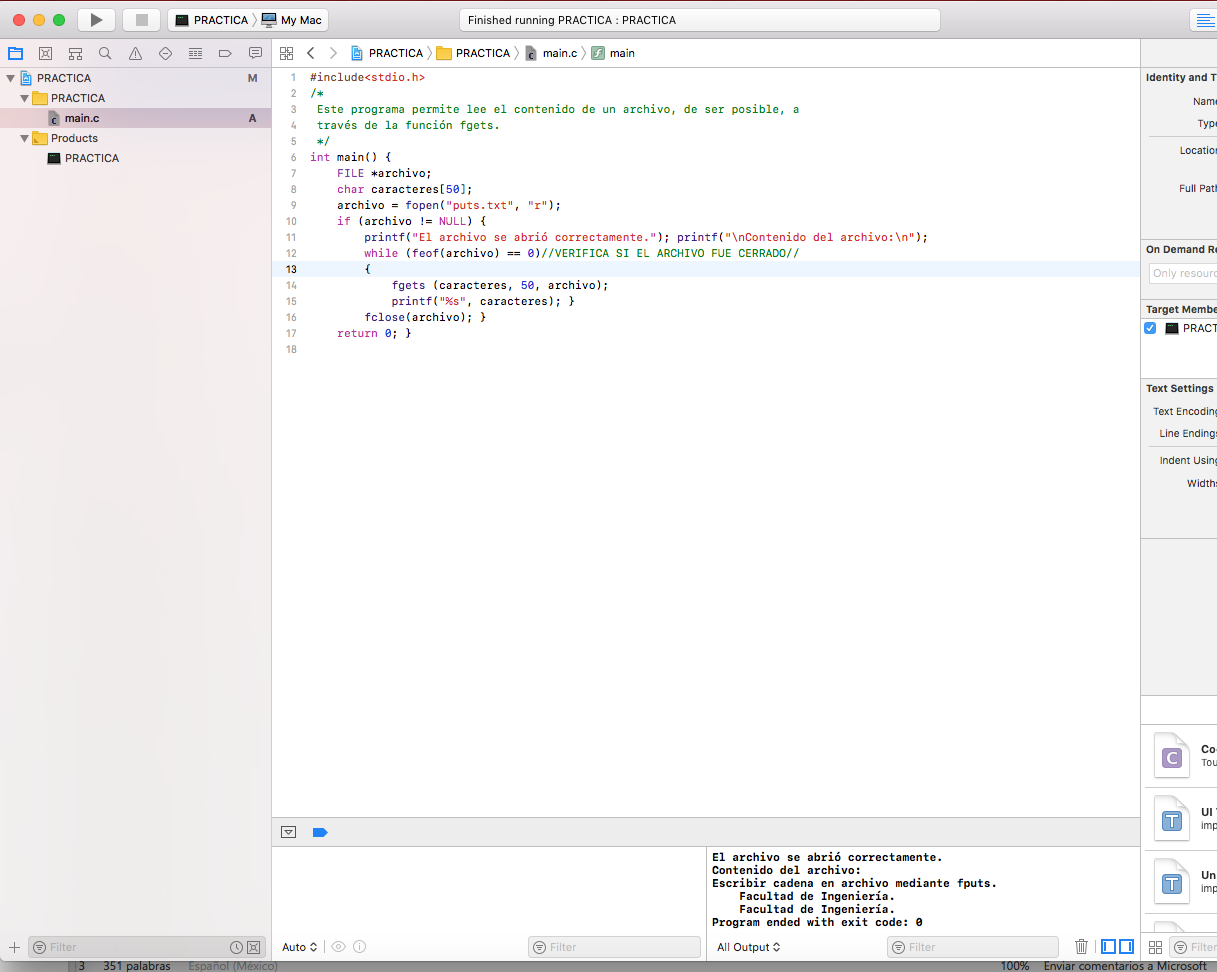
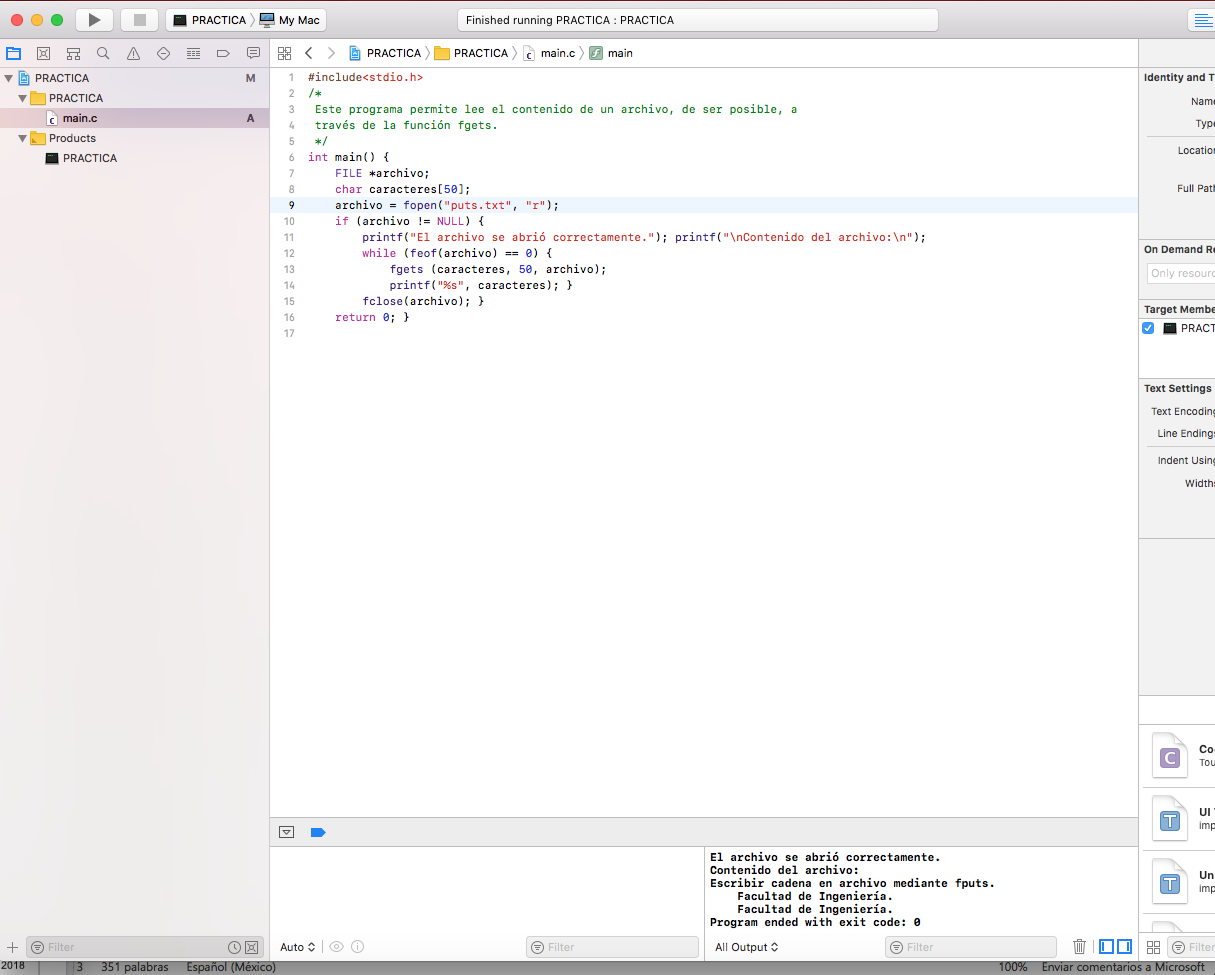
Se modifica el modo y debido a que el archivo existe, entonces valida



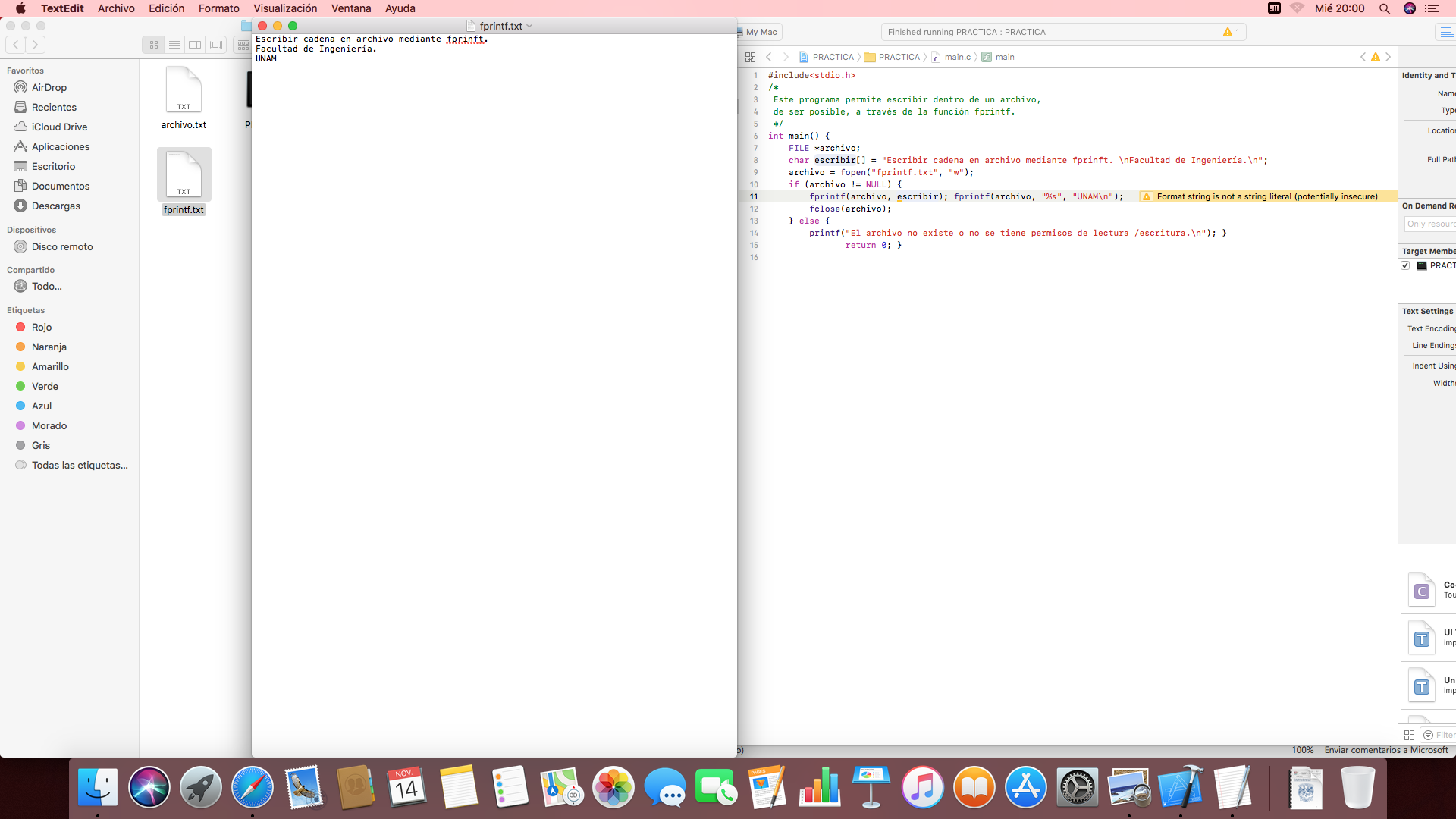
Las funciones fgets() y fputs() pueden leer y escribir, respectivamente, cadenas sobre los archivos. Las firmas de estas funciones son, respectivamente:

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

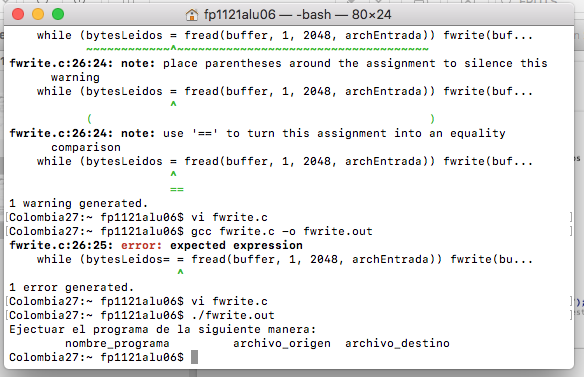
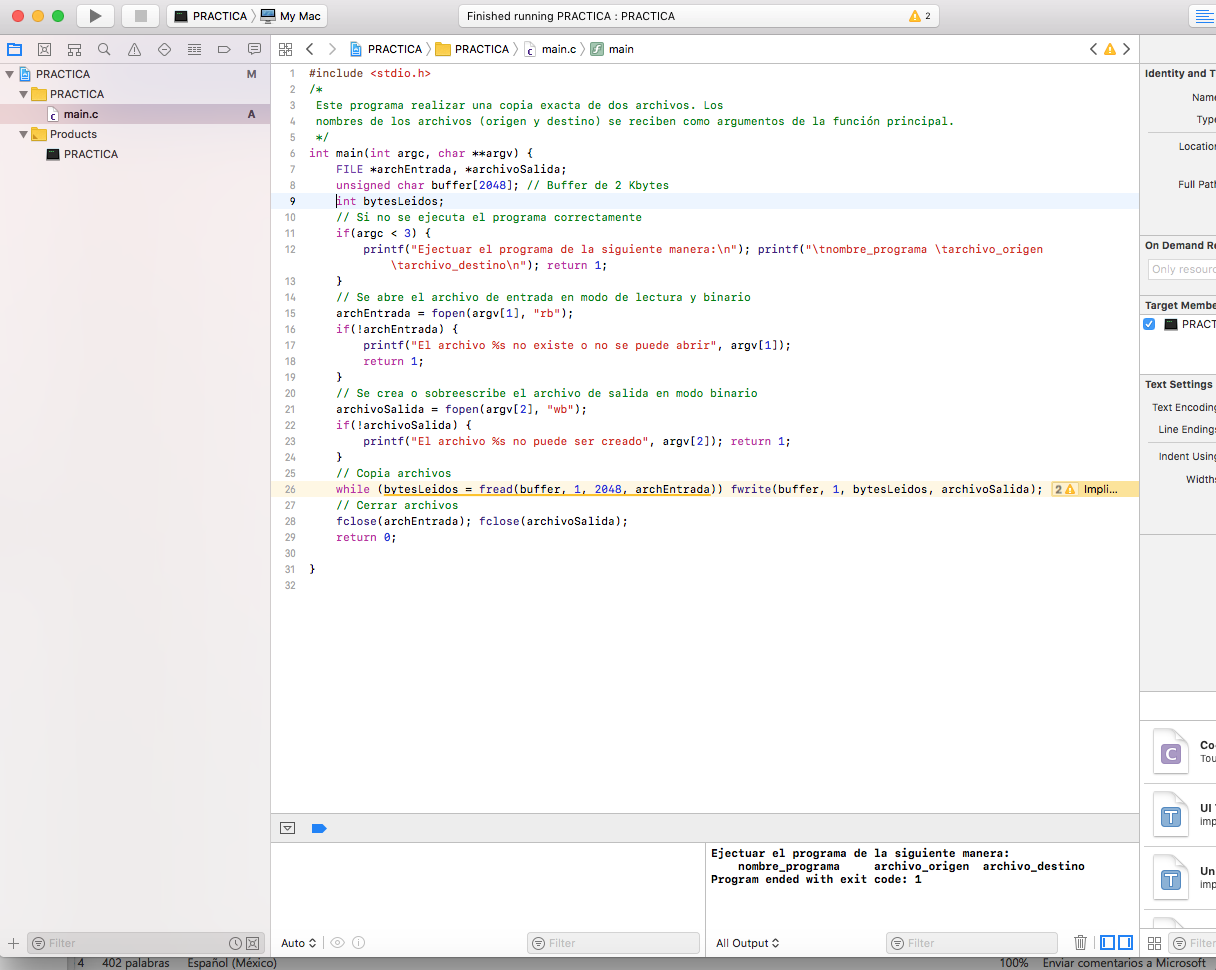
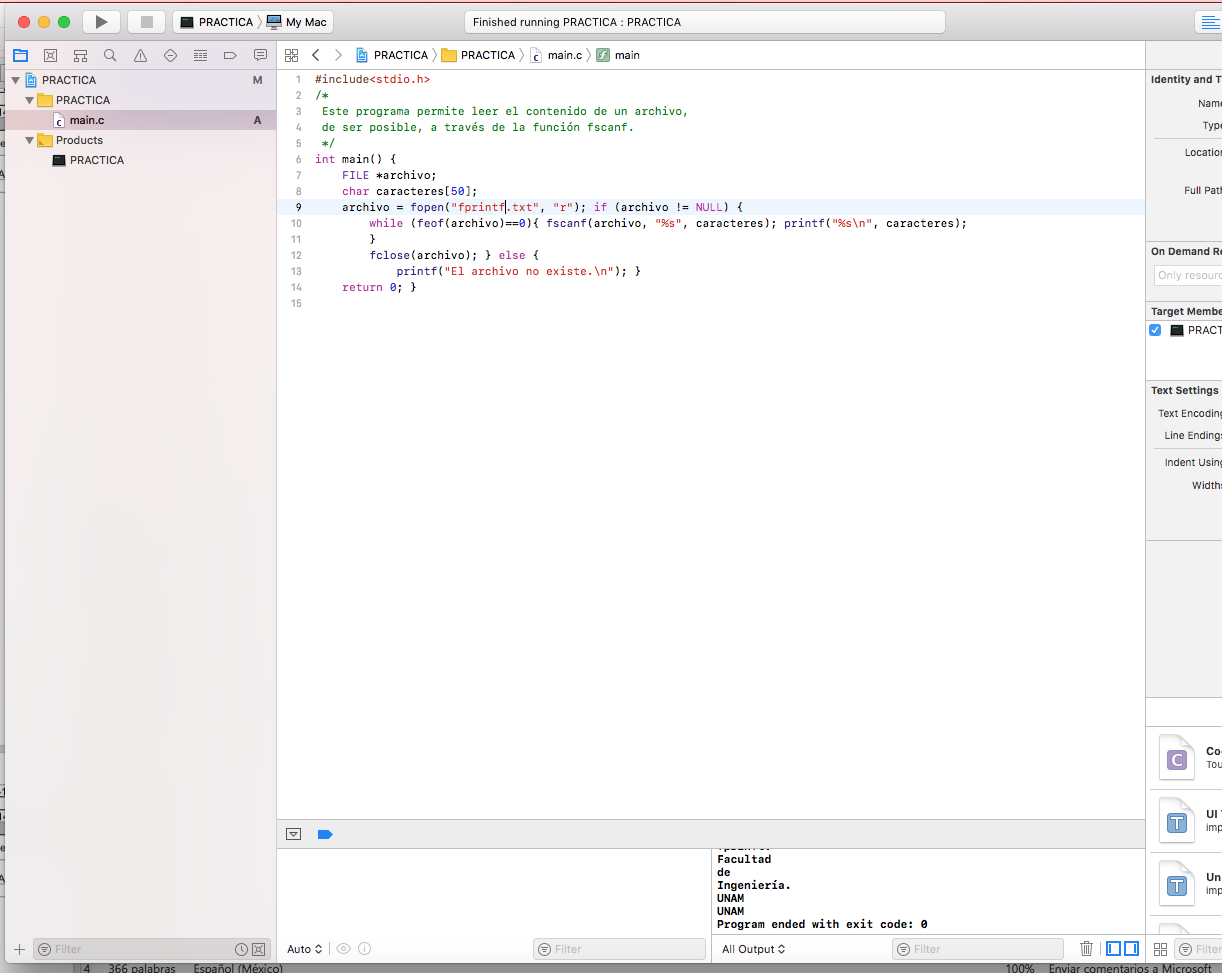
La función fputs() permite escribir una cadena en un archivo especifico. La función fgets() permite leer una cadena desde el archivo especificado. Esta función lee un renglón a la vez.



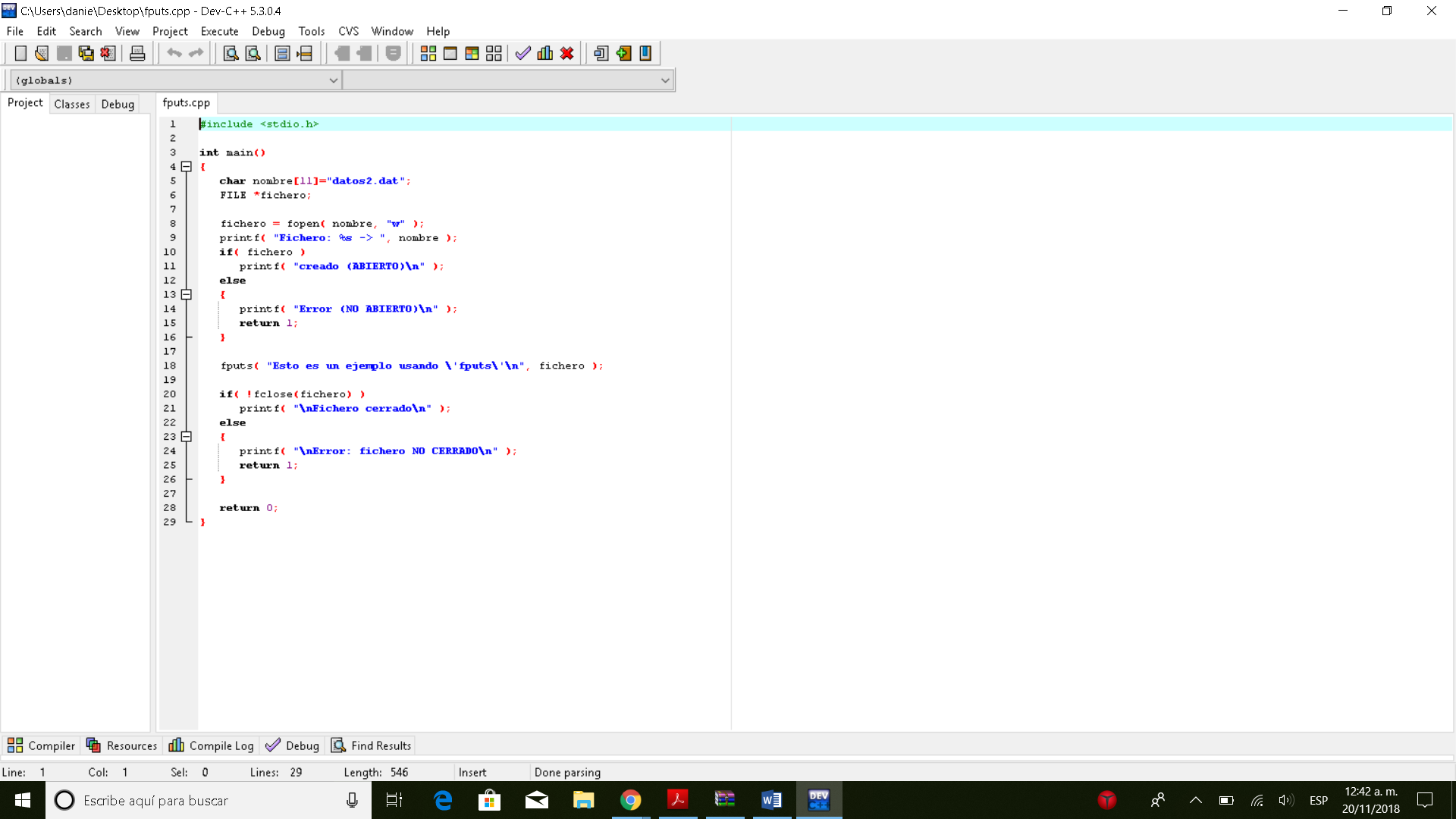
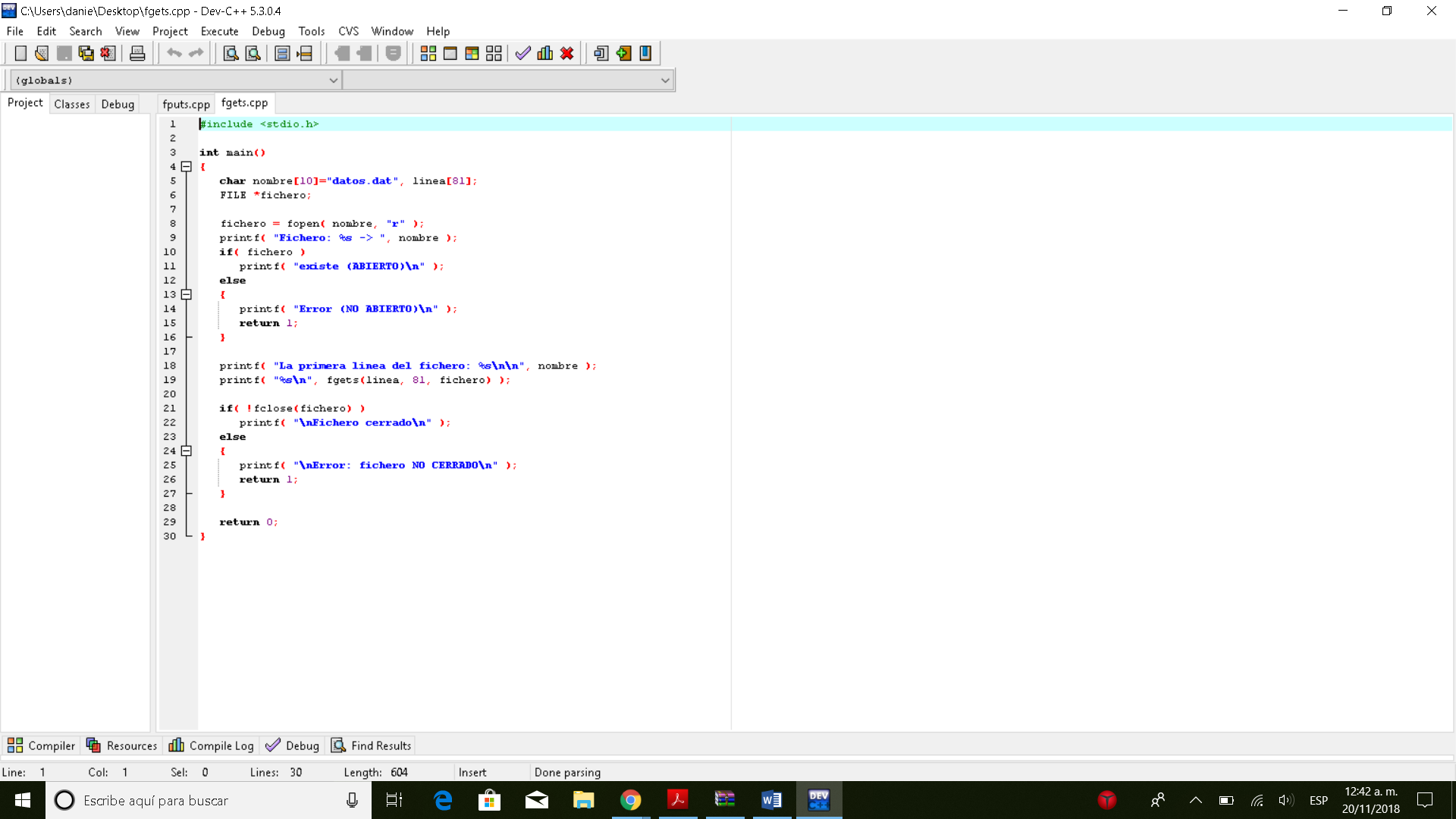
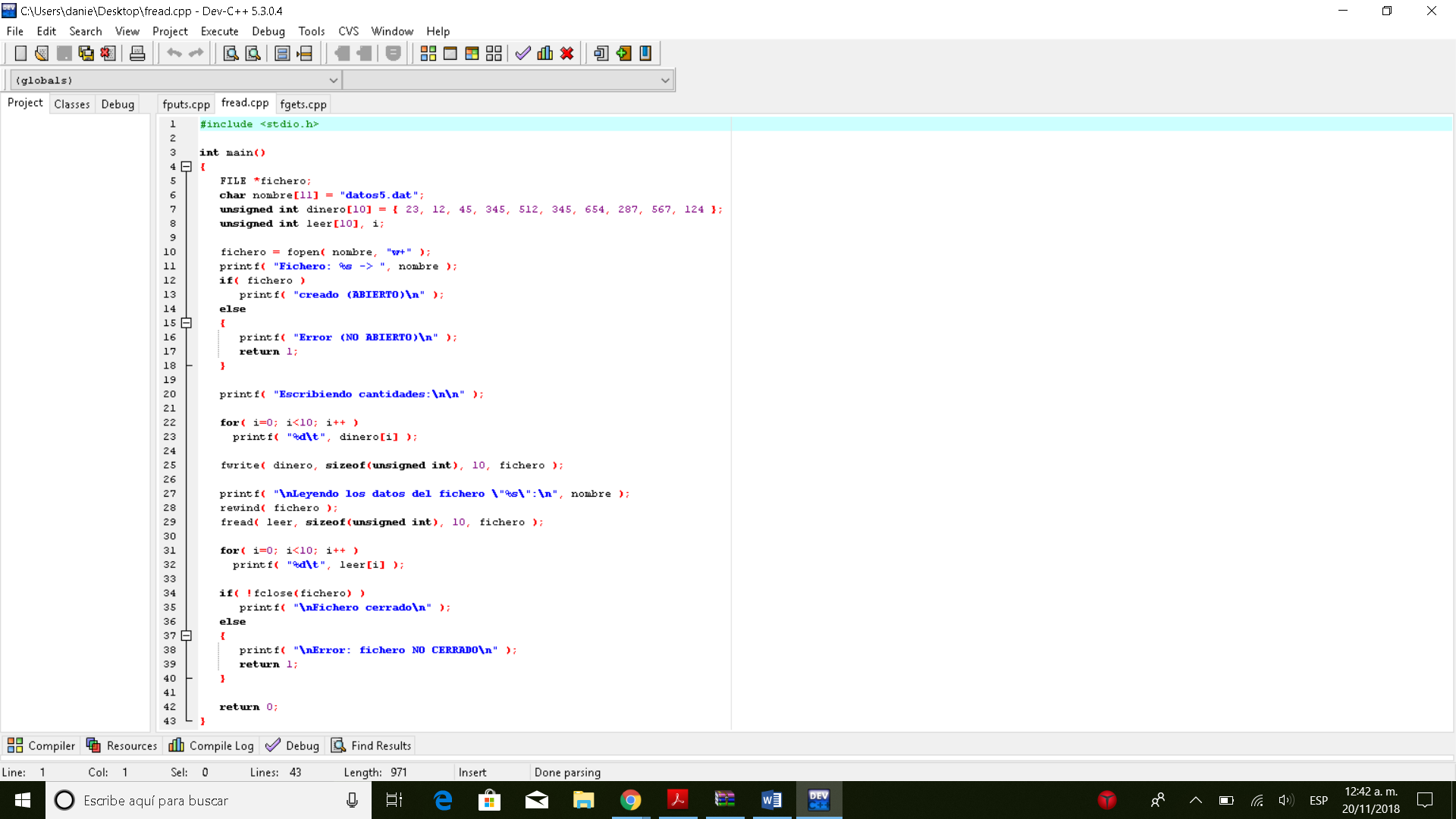
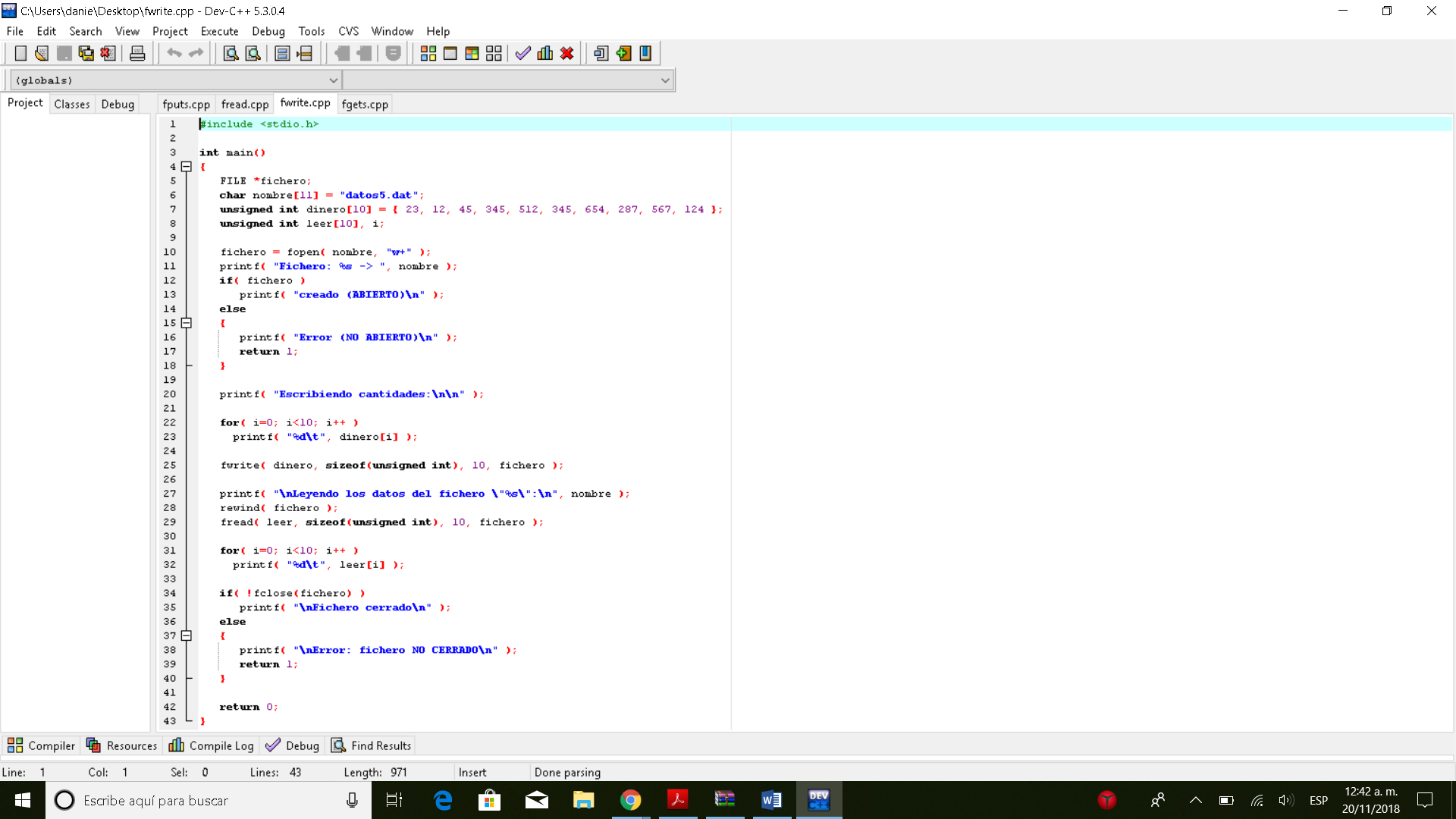
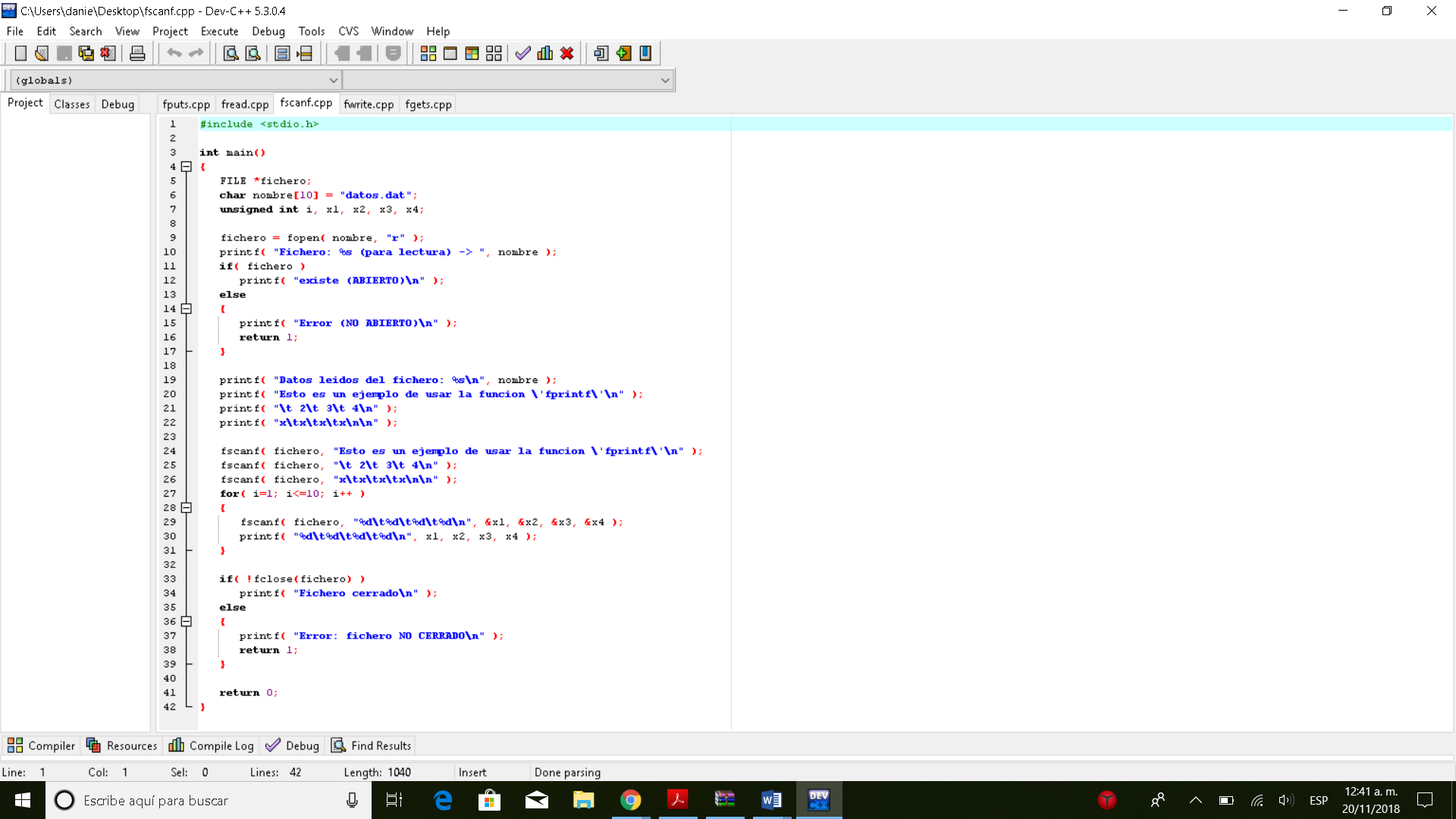
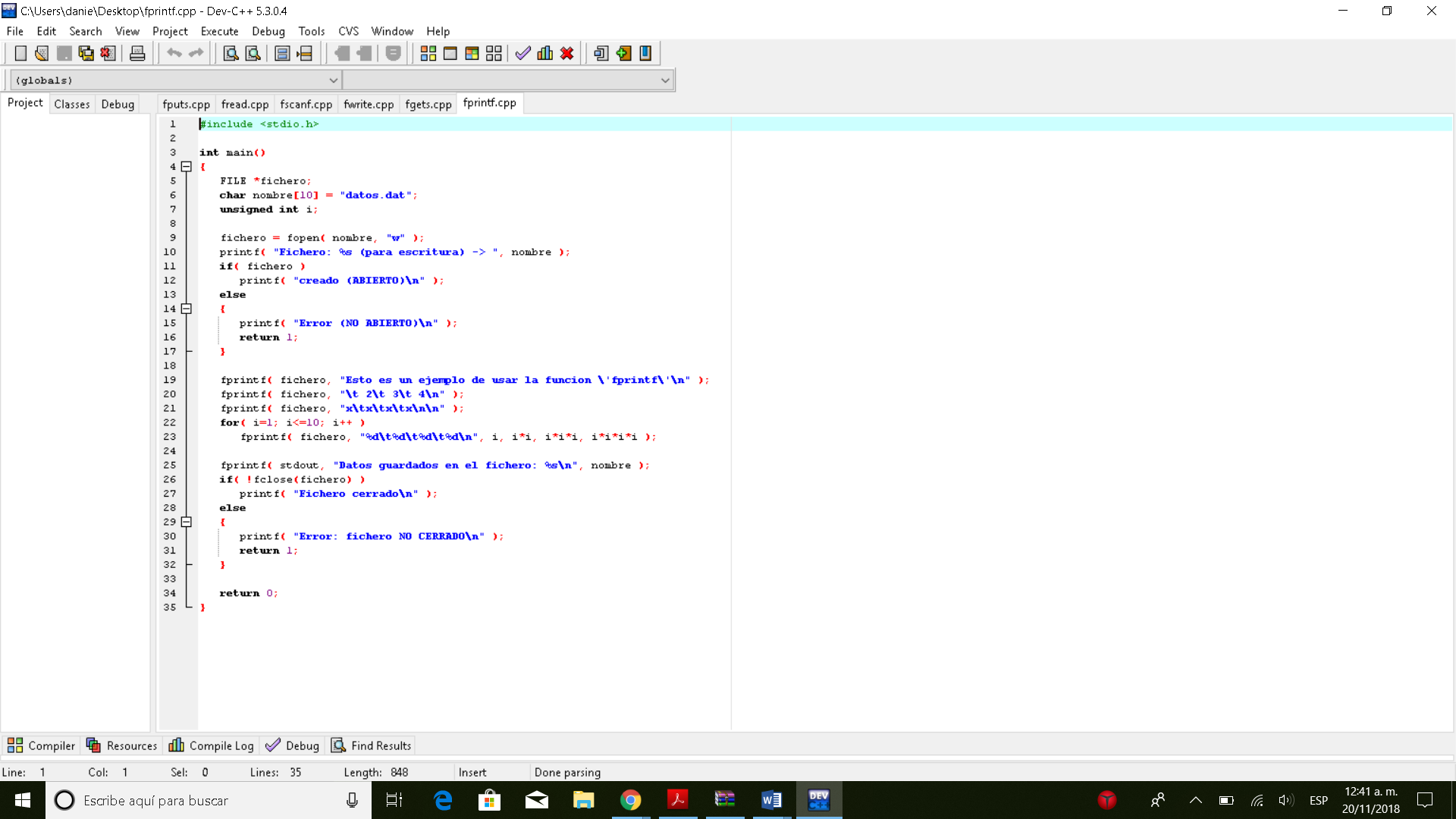
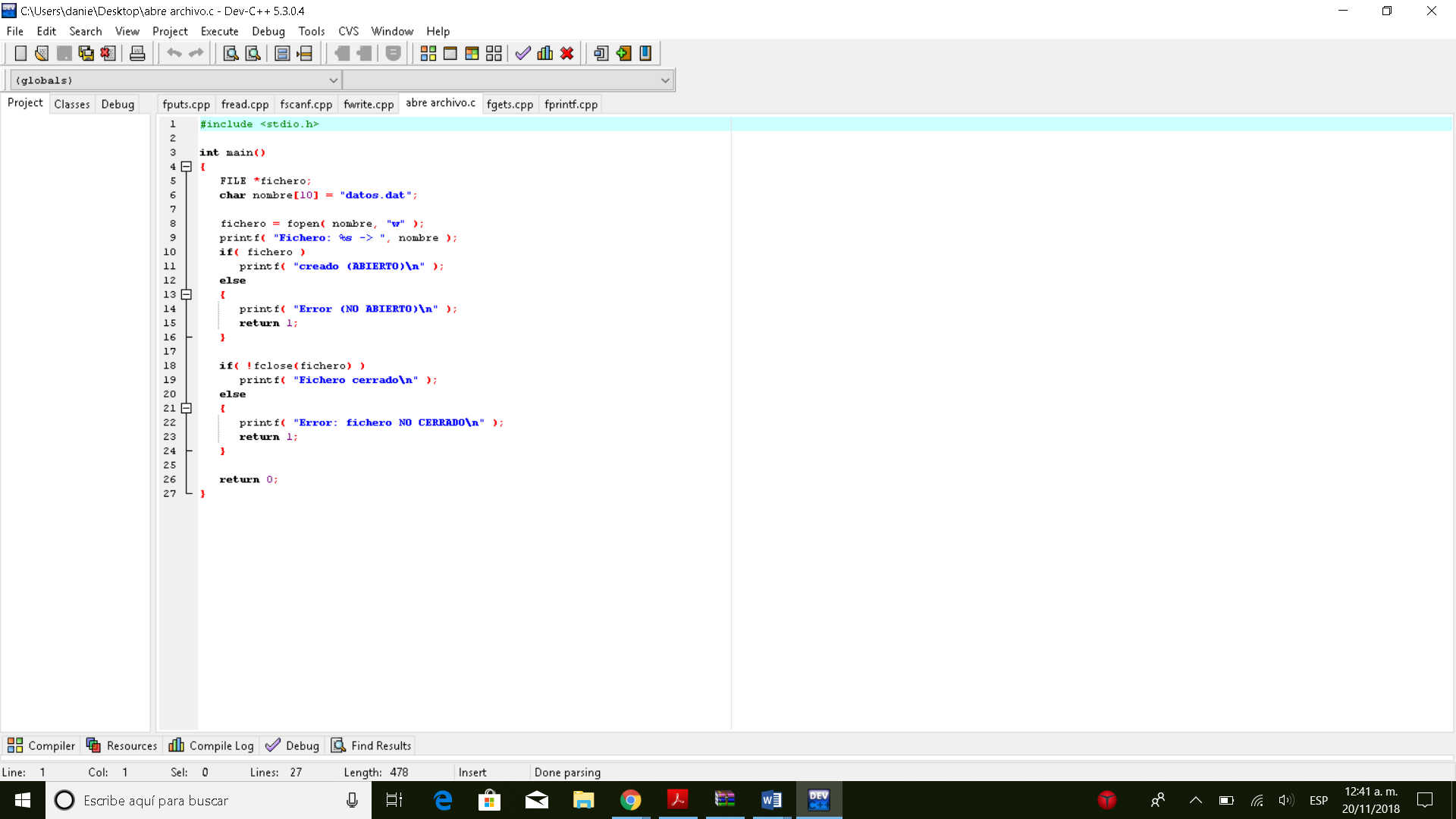
Utilizamos la función fprintf y verificamos en los archivos que el mensaje sea el correcto. Nosotros insertamos información. No podemos ver lo que escribimos en terminal.



Ahora utilizamos la función fscanf y podemos visualizar la función fprintf. Nosotros leemos la información del texto y podemos ver lo que escribimos en terminal.



**Tarea:**



**Conclusiones:** Los ejercicios realizados durante la práctica fueron los suficientes como para poder comprender la función de cada función.

Es mas complejo trabajar desde la terminal que desde XCode debido a que gráficamente no te provee de tanta información o es fácil de editar.

Los ejercicios propuestos me permitieron ver mas de cerca la estructura general.

**Bibliografía**

El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.