



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Ing. Claudia Rodriguez Espino

*Asignatura:* Fundamentos de Programacion

*Grupo:* 3

*No de Práctica(s):* Practica 13


*Integrante(s):* Borja Portela Jose Fabio

*Semestre:* 2018-2

*Fecha de entrega:* 14 de mayo del 2018

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

	<b>Manual de prácticas del Laboratorio de Fundamentos de programación</b>	Código:	MADO-17
		Versión:	02
		Página	204/214
		Sección ISO	8.3
		Fecha de emisión	6 de abril de 2018
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

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

### OBJETIVO:

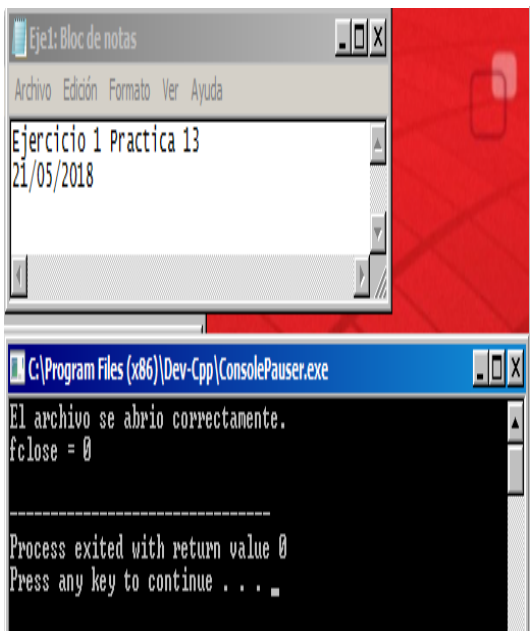
Elaborar programas en C que requieran el uso de archivos de texto plano en la resolución de problemas, entendiendo a los archivos como un elemento de almacenamiento secundario.

### INTRODUCCION:

En el presente reporte se entregan capturas de pantalla pertinentes respectivas a los desarrollos de la práctica y también para los ejercicios de tarea, siempre y cuando sea el caso, dichas imágenes y procedimientos irán detallados por una lista acciones y actividades seguidas en la clase de laboratorio.

### DESARROLLO:

#### 1. Código (abrir cerrar archivo)



```
#include<stdio.h>
/*
Este programa permite abrir un archivo en modo de lectura, de ser posible.
*/
int main() {
    FILE *archivo;
    archivo = fopen("ejel.txt", "r");

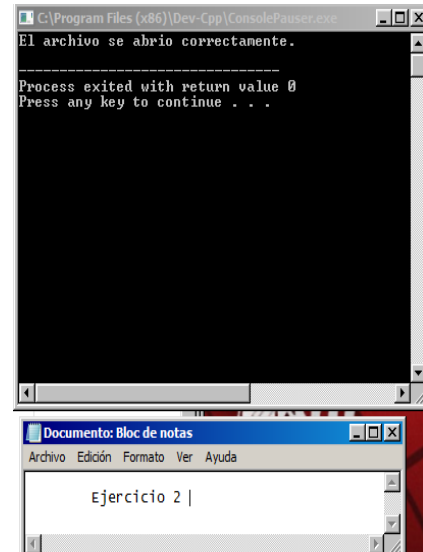
    if (archivo != NULL) {
        printf("El archivo se abrió correctamente.\n");
        int res = fclose(archivo);
        printf("fclose = %d\n", res);
    } else {
        printf("Error al abrir el archivo.\n");
        printf("El archivo no existe o no se tienen permisos de lectura.\n");
    }

    return 0;
}
```

## 2. Código (fgets)

```
#include<stdio.h>
/*
 Este programa permite leer el contenido de un archivo, de ser posible, a
 través de la función fgets.
 */
int main() {
    FILE *archivo;
    char caracteres[50];
    archivo = fopen("Documento.txt", "r");

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

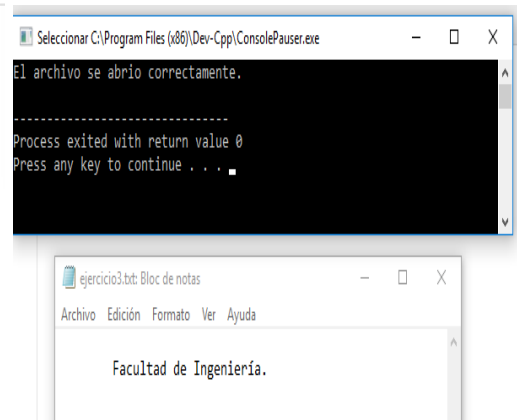


## 3. Código (fputs)

```
#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[] = "\n\tFacultad de Ingenieria.\n";
    archivo = fopen("ejercicio3.txt", "r+");

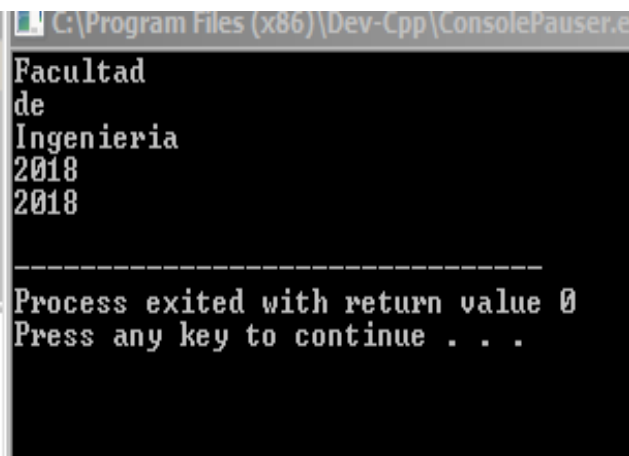
    if (archivo != NULL) {
        printf("El archivo se abrio correctamente.\n");
        fputs (escribir, archivo);
        fclose(archivo);
    } else {
        printf("Error al abrir el archivo.\n");
        printf("El archivo no existe o no se tienen permisos de lectura.\n");
    }

    return 0;
}
```

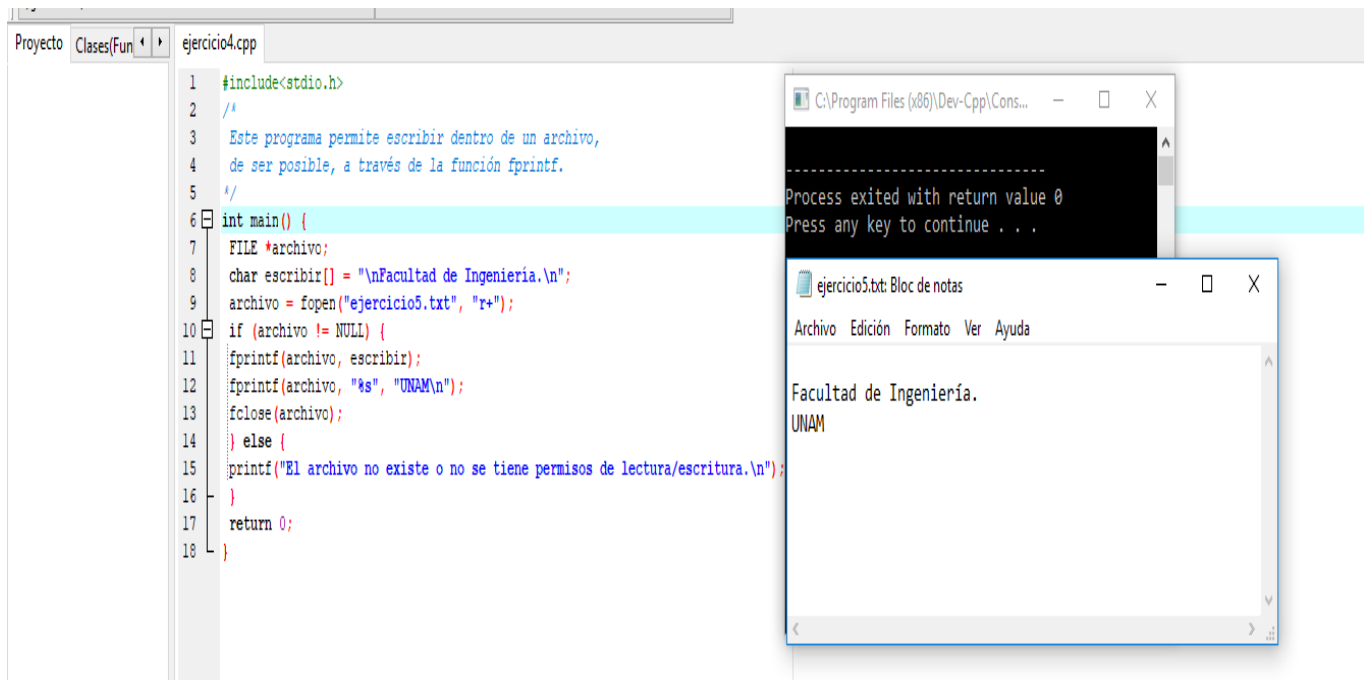


## 4. Código (fscanf)

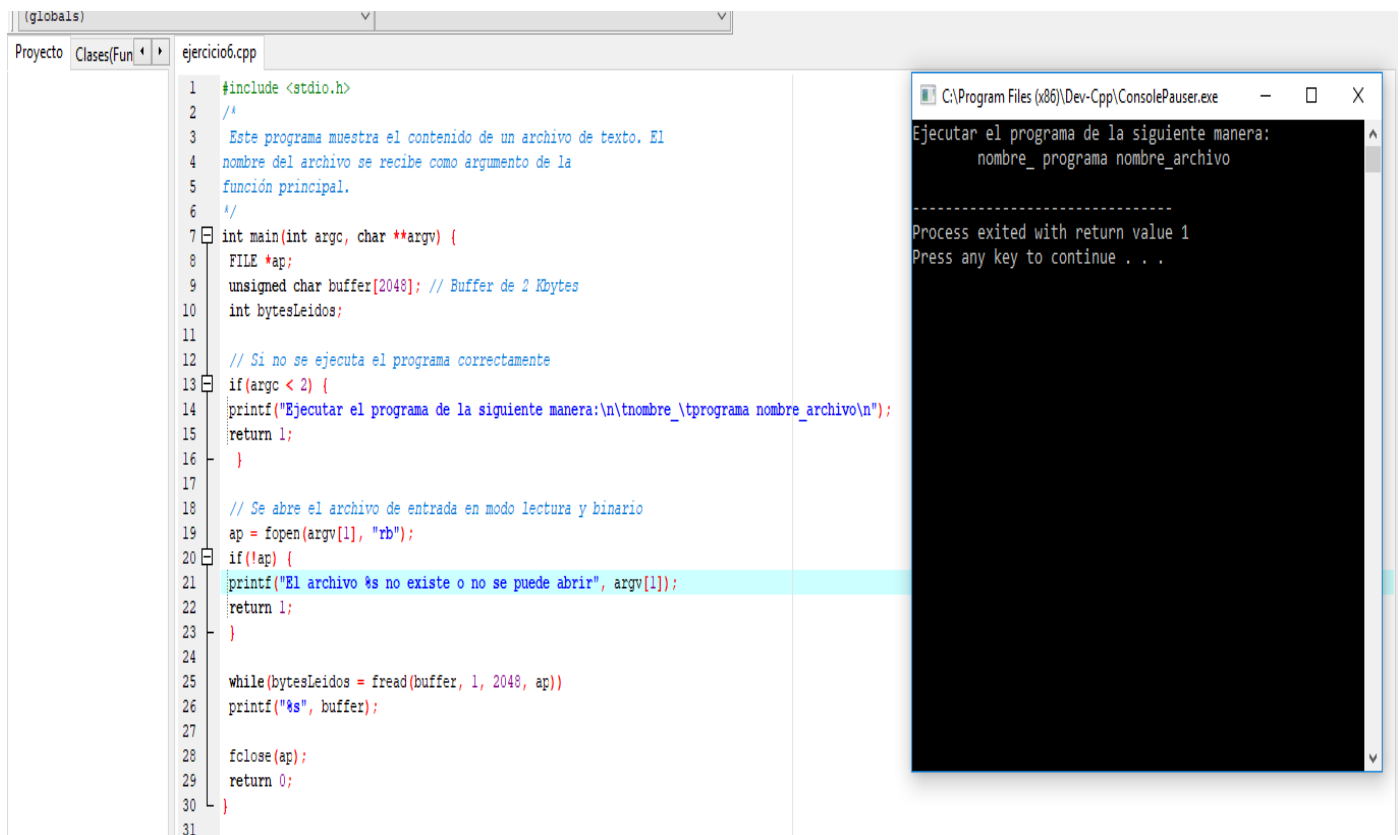
```
#include<stdio.h>
/*
 Este programa permite leer el contenido de un archivo,
 de ser posible, a través de la función fscanf.
 */
int main() {
    FILE *archivo;
    char caracteres[50];
    archivo = fopen("Documento2.txt", "r");
    if (archivo != NULL) {
        while (feof(archivo)==0){
            fscanf(archivo, "%s", caracteres);
            printf("%s\n", caracteres);
        }
        fclose(archivo);
    } else {
        printf("El archivo no existe.\n");
    }
    return 0;
}
```



## 5. Código (fprintf)



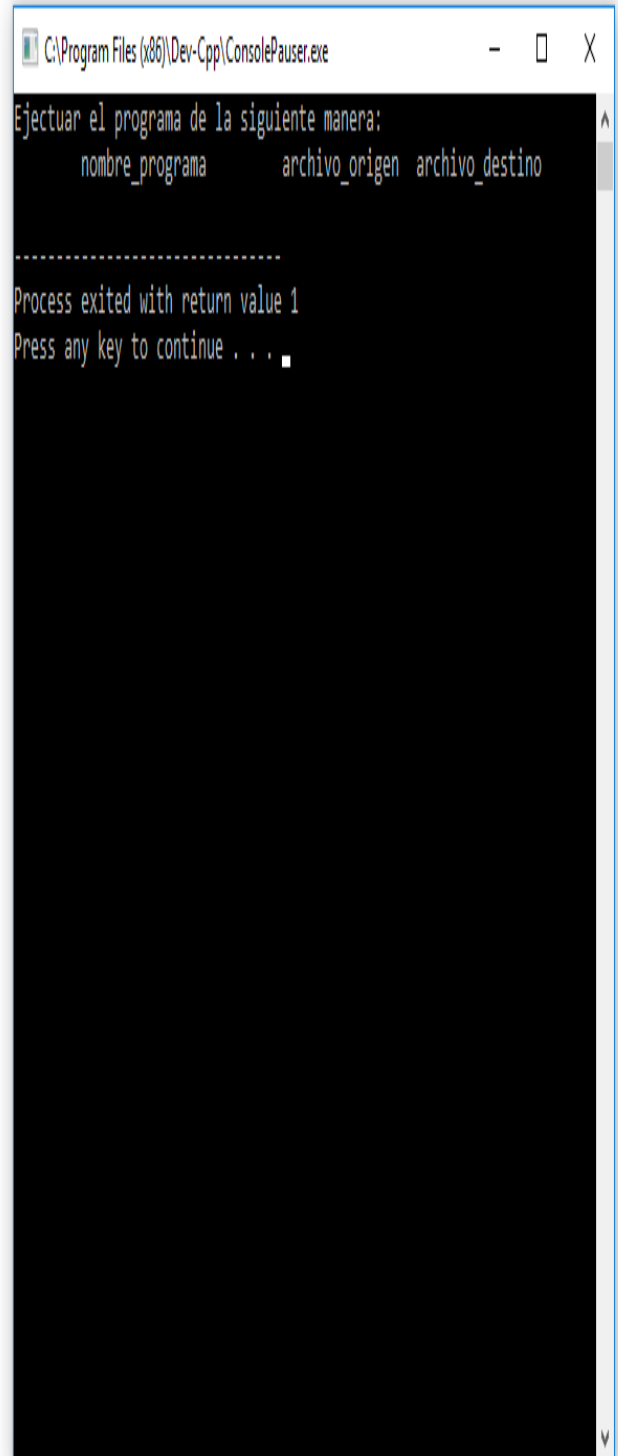
## 6. Código (fread)



## 7. Código (fwrite)

ejercicio6.cpp ejercicio7.cpp

```
1 #include <stdio.h>
2 /*
3  Este programa realizar una copia exacta de dos archivos. Los
4  nombres de los archivos (origen y destino) se reciben como
5  argumentos de la función principal.
6  */
7 int main(int argc, char **argv) {
8     FILE *archEntrada, *archivoSalida;
9     unsigned char buffer[2048]; // Buffer de 2 Kbytes
10    int bytesLeidos;
11
12    // Si no se ejecuta el programa correctamente
13    if(argc < 3) {
14        printf("Ejectuar el programa de la siguiente manera:\n");
15        printf("\tnombre_programa \tarchivo_origen \tarchivo_destino\n");
16        return 1;
17    }
18    // Se abre el archivo de entrada en modo de lectura y binario
19    archEntrada = fopen(argv[1], "rb");
20    if(!archEntrada) {
21        printf("El archivo %s no existe o no se puede abrir", argv[1]);
22        return 1;
23    }
24    // Se crea o sobrescribe el archivo de salida en modo binario
25    archivoSalida = fopen(argv[2], "wb");
26    if(!archivoSalida) {
27        printf("El archivo %s no puede ser creado", argv[2]);
28        return 1;
29    }
30
31    // Copia archivos
32    while (bytesLeidos = fread(buffer, 1, 2048, archEntrada))
33        fwrite(buffer, 1, bytesLeidos, archivoSalida);
34
35    // Cerrar archivos
36    fclose(archEntrada);
37    fclose(archivoSalida);
38    return 0;
```



```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Ejectuar el programa de la siguiente manera:
    nombre_programa      archivo_origen  archivo_destino

-----
Process exited with return value 1
Press any key to continue . . .
```

### **CONCLUSIONES:**

La presente practica fue de gran utilidad, ya que nos enseñó comandos esenciales para acceder a algunos archivos desde dev c, con funciones como fopen, lo cual puede ser sumamente útil al momento de correr ejecutables, mas cuando van de la mano y se puede utilizar su código en nuestra función principal main.