



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

Marco Antonio Martínez Quintana

*Profesor:*

Fundamentos de Programación

*Asignatura:*

3

*Grupo:*

13

*No de Práctica(s):*

Sánchez Hernández Marco Antonio

*Integrante(s):*

*No. de Equipo de  
cómputo empleado:*

No aplica

48

*No. de Lista o Brigada:*

2021-1

*Semestre:*

24/Enero/2021

*Fecha de entrega:*

*Observaciones:*

Se utilizó el sistema operativo Kubuntu 20.04 LTS con una configuración de TWM y el procesador de texto NeoVim

**CALIFICACIÓN:** \_\_\_\_\_

# Lectura y escritura de datos

## Introducción

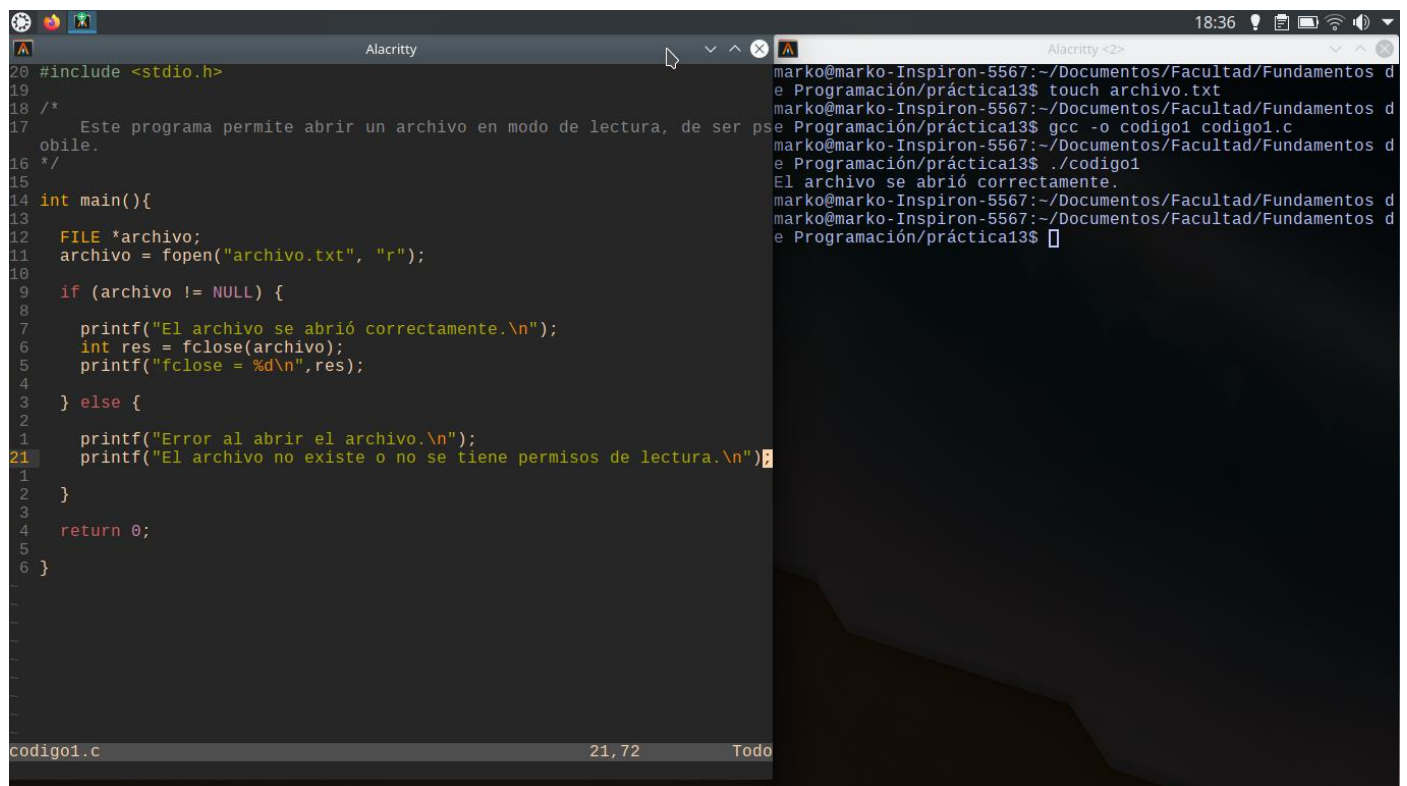
“Un archivo es un conjunto de datos estructurados en una colección de entidades elementales o básicas denominadas registros que son del mismo tipo, pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.” (Solano J., 2020). Los archivos nos permiten almacenar datos, de tal forma que nuestra información ocupe un determinado espacio en el disco duro de nuestra computadora, permitiendo conservar la información después de la ejecución de un programa.

### 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.

## Ejercicios

### Ejercicio 1: abrir y cerrar un archivo



```
#include <stdio.h>
/*
 Este programa permite abrir un archivo en modo de lectura, de ser posible.
 */
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 tiene permisos de lectura.\n");
    }
    return 0;
}
```

```
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos de Programación/práctica13$ touch archivo.txt
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos de Programación/práctica13$ gcc -o codigo1 codigo1.c
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos de Programación/práctica13$ ./codigo1
El archivo se abrió correctamente.
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos de Programación/práctica13$
```

Antes de la ejecución de este programa fue necesario la creación de un archivo con el nombre “archivo.txt” para lo cual se utilizó el comando touch en la consola de Linux (en este caso Alacritty).

### Ejercicio 2: función fgets

The screenshot shows three Alacritty terminal windows. The leftmost window displays the source code for `codigo2.c`, which uses `fgetc` to read the contents of `gets.txt`. The middle window shows the compilation and execution of `codigo2.c`, with output indicating successful file opening and reading of the text "Prueba". The rightmost window shows the file `gets.txt` with its contents.

```
#include <stdio.h>
/*
 * Este programa permite leer el contenido de un archivo, de ser posible
 * a través de la función fgetc.
 */
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) {
            fgetc (caracteres,50,archivo);
            printf("%s",caracteres);
        }
        fclose(archivo);
    }
    return 0;
}
```

```
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos de Programación/práctica13$ gcc -o codigo2 codigo2.c
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos de Programación/práctica13$ ./codigo2
El archivo se abrió correctamente.
Contenido del archivo:
Prueba
Prueba
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos de Programación/práctica13$
```

gets.txt 1L, 7C escritos

Para este ejercicio fue creado un archivo con el nombre “fgetc.txt” que contiene la cadena de texto “Prueba” antes de la ejecución del código.

### Ejercicio 3: función fputs

The screenshot shows three Alacritty terminal windows. The leftmost window displays the source code for `codigo3.c`, which uses `fputs` to write the contents of `escribir` to `puts.txt`. The middle window shows the compilation and execution of `codigo3.c`, with output indicating successful file opening and writing. The rightmost window shows the file `puts.txt` with its contents.

```
#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[] = "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;
}
```

```
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos de Programación/práctica13$ gcc -o codigo3 codigo3.c
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos de Programación/práctica13$ ./codigo3
El archivo se abrió correctamente.
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos de Programación/práctica13$
```

puts.txt 1L, 1C escritos

Para este ejercicio se creó un archivo con el nombre “puts.txt” para que al momento de ejecutar el programa este pudiera escribir sobre este la cadena de texto.

### Ejercicio 4: función fscanf

```
1 #include <stdio.h>
2 /*
3  Este programa permite leer el contenido de un archivo,
4  de ser posible, a través de la función fscanf.
5 */
6
7 int main(){
8
9     FILE *archivo;
10    char caracteres[50];
11    archivo = fopen("fscanf.txt","r");
12    if (archivo != NULL){
13
14        while (feof(archivo)==0){
15
16            fscanf(archivo, "%s", caracteres);
17            printf("%s\n",caracteres);
18
19        }
20
21        fclose(archivo);
22    } else {
23
24        printf("El archivo no existe.\n");
25
26    }
27
28    return 0;
29 }
30
31 }
```

```
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$ gcc -o codigo4 codigo4.c
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$ ./codigo4
prueba
prueba
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$
```

```
1 prueba
```

codigo4.c 1,18 Todo  
"codigo4.c" [Nuevo] 32L, 462C escritos

fscanf.txt 1,6 Todo  
"fscanf.txt" 1L, 7C escritos

Para este ejercicio fue necesaria la creación de un archivo de texto de nombre “fscanf.txt” que contuviera alguna cadena de texto para que al ejecutar el programa este la mostrara.

## Ejercicio 5: función fprintf

```
10 #include <stdio.h>
11
12 /*
13  Este programa permite escribir dentro de un archivo,
14  de ser posible, a través de la función fprintf.
15 */
16
17 int main(){
18
19     FILE *archivo;
20     char escribir[] = "Escribir cadena en archivo mediante fprintf. \nFacul
21     tad de Ingeniería.\n";
22     archivo = fopen("fprintf.txt","r+");
23
24     if (archivo !=NULL){
25
26         fprintf(archivo, escribir);
27         fprintf(archivo, "%s", "UNAM\n");
28         fclose(archivo);
29
30     } else {
31
32         printf("El archivo no existe o no se tiene permiso de lectura/escrit
33         ura.\n");
34
35     }
36
37     return 0;
38 }
39 }
```

```
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$ gcc -o codigo5 codigo5.c
codigo5.c: In function 'main':
codigo5.c:16:22: warning: format not a string literal and no
format arguments [-Wformat-security]
16 |     fprintf(archivo, escribir);
   |                  ~~~~~~
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$ ls
archivo.txt  codigo2  codigo3.c  codigo5  fscanf.txt
codigo1     codigo2.c  codigo4    codigo5.c  gets.txt
codigo1.c   codigo3    codigo4.c  fprintf.txt  puts.txt
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$ ./codigo5
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$
```

```
1 Escribir cadena en archivo mediante fprintf.
2 Facultad de Ingeniería.
3 UNAM
```

codigo5.c 11,95-94 Todo  
"codigo5.c" [Nuevo] 11L, 462C escritos

fprintf.txt 1,1 Todo  
"fprintf.txt" 3L, 37C escritos

Para este ejercicio fue necesaria la creación de un archivo de texto que tuviese por nombre “fprintf.txt” antes de la ejecución del código.

## Ejercicio 6: función fread

The screenshot shows three Alacritty terminal windows. The top-left window displays the source code of a C program named `codigo6.c`. The program uses `fread` to read a file specified as a command-line argument. The top-right window shows the terminal output after compiling and running the program without arguments, resulting in an error message about the missing file. The bottom window shows the program being run with `prueba.txt` as an argument, which successfully displays the file's content.

```
1 #include <stdio.h>
2 /*
3  Este programa muestra el contenido de un archivo de texto. El nombre
4  del archivo se recibe como argumento de la función principal.
5 */
6 int main(int argc, char **argv){
7     FILE *ap;
8     unsigned char buffer[2048]; // Buffer de 2 kbytes
9     int bytesLeidos;
10    //Si no se ejecuta el programa correctamente
11    if (argc < 2){
12        printf("Ejecutar el programa de la siguiente manera: \n\nnombre\tpr
13        ograma nombre_archivo\n");
14    }
15    return 1;
16 }
17 //Se abre el archivo de entrada en modo lectura y binario
18 ap = fopen(argv[1], "rb");
19 if(!ap){
20     printf("El archivo %s no existe o no se puede abrir", argv[1]);
21     return 1;
22 }
23 while(bytesLeidos = fread(buffer, 1, 2048, ap))
24     printf("%s", buffer);
25 }
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

codigo6.c 1,13 Comienzo  
"codigo6.c" 40L, 760C escritos

```
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$ gcc -o codigo6 codigo6.c
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$ ./codigo6
Ejecutar el programa de la siguiente manera:
nombre_programa nombre_archivo
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$ ./codigo6 prueba.txt
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$ ./codigo6 prueba.txt
prueba
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$
```

prueba.txt 1,6 Todo  
"prueba.txt" 1L, 7C escritos

```
35 /*
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

codigo6.c 40,1 Final  
"codigo6.c" 40L, 760C escritos

```
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$ gcc -o codigo6 codigo6.c
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$ ./codigo6
Ejecutar el programa de la siguiente manera:
nombre_programa nombre_archivo
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$ ./codigo6 prueba.txt
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$ ./codigo6 prueba.txt
prueba
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$
```

prueba.txt 1,6 Todo  
"prueba.txt" 1L, 7C escritos

En este programa se creó primero el archivo de nombre “prueba.txt” al cual se le escribió prueba, al momento de ejecutar el código y no proporcionar el nombre del archivo como parámetro de la función main, el código no mostrará el contenido, sino hasta que reciba como parámetro el nombre del archivo.

## Ejercicio 7: función fwrite



```
#include <stdio.h>
/*
 * Este programa realiza una copia exacta de dos archivos. Los nombres
 * de los archivos (origen y destino) se reciben como argumentos de la funcio
 * n principal.
 */
int main(int argc, char **argv){
    FILE *archEntrada, *archSalida;
    unsigned char buffer[2048]; //Buffer de 2 kbytes
    int bytesLeidos;
    //Si no se ejecuta el programa correctamente
    if (argc < 3){
        printf("Ejecutar el nombre de la siguiente manera:\n");
        printf("\tnombre_programa \tarchivo_origen \tarchivo_destino\n");
        return 1;
    }
    //Se abre el archivo de entrada en modo de lectura y binario
    archEntrada = fopen(argv[1], "rb");
    if (!archEntrada){
        printf("El archivo %s no existe o no se puede abrir", argv[1]);
        return 1;
    }
    //Se crea o sobrescribe el archivo de salida en modo binario
    archSalida = fopen(argv[2], "wb");
    while (bytesLeidos = fread(buffer, 1, 2048, archEntrada))
        fwrite(buffer, 1, bytesLeidos, archSalida);
    // Cerrar archivos
    fclose(archEntrada);
    fclose(archSalida);
    return 0;
}
```

codigo7.c 9,26 Comienzo

marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos de Programaci3n/pr3ctica13\$ gcc -o codigo7 codigo7.c

codigo7.c: In function 'main':

codigo7.c:16:12: warning: comparison between pointer and integer

16 | if (argv < 3){

marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos de Programaci3n/pr3ctica13\$ ./codigo7 prueba.txt destino.txt

marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos de Programaci3n/pr3ctica13\$

prueba.txt 1,6 Todo

destino.txt 1,1 Todo

marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos de Programaci3n/pr3ctica13\$ gcc -o codigo7 codigo7.c

codigo7.c: In function 'main':

codigo7.c:16:12: warning: comparison between pointer and integer

16 | if (argv < 3){

marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos de Programaci3n/pr3ctica13\$ ./codigo7 prueba.txt destino.txt

marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos de Programaci3n/pr3ctica13\$

prueba.txt 1,6 Todo

destino.txt 1,1 Todo

codigo7.c 54,1 Final

"codigo7.c" 54L, 1162C escritos

Al igual que en el ejercicio anterior, requiere que se indique el nombre del archivo que desea copiarse en otro como un parámetro de la función main, cabe resaltar que se utilizó el mismo archivo que en el ejercicio anterior, y, a diferencia del ejercicio anterior, se debe proporcionar un segundo nombre de archivo que es donde se copiará, como el modo de escritura para el archivo de destino es w, no es necesario crearlo, ya que, en caso de no existir, este lo creará de ser posible.

## Ejercicio 8: suma de los primeros n números

```
15 #include <stdio.h>
14
13 int main(){
12     //Declarar variables
11     int n,res;
10
9     //Apuntador al archivo
8     FILE *a;
7     a=fopen("resultadosGauss.txt","w");
6
5     //Mensaje de bienvenida
4     printf("\n\n\t\t\tSuma de los primeros n números\n\n");
3
2
1     //Solicitar el número de elementos a sumar
16 printf("¿Cuántos números deseas sumar? ")
1 scanf("%d",&n);
2
3     //Sumar los n números
4     res=0;
5     for(int i=1;i<=n;i++)
6     {
7         fprintf(a,"%d + %d =",res,i);
8         res=res+i;
9         fprintf(a,"%d\n",res);
10    }
11
12    //Mostrar el resultado
13    printf("La suma de los primeros %d números es: %d \n",n,res);
14
15    fclose(a);
16
17    return 0;
18
19
20 }

gaussArchivos.c 16,47-44 Todo
"gaussArchivos.c" 36L, 608C escritos
```

```
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$ gcc -o gaussArchivos gaussArchivos
.c
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$ ./gaussArchivos

Suma de los primeros n números

¿Cuántos números deseas sumar? 100
La suma de los primeros 100 números es: 5050
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$
```

```
1 0 + 1 =1
1 1 + 2 =3
2 3 + 3 =6
3 6 + 4 =10
4 10 + 5 =15
5 15 + 6 =21
6 21 + 7 =28
7 28 + 8 =36
8 36 + 9 =45
9 45 + 10 =55
10 55 + 11 =66
11 66 + 12 =78
12 78 + 13 =91
13 91 + 14 =105
14 105 + 15 =120
15 120 + 16 =136
16 136 + 17 =153

resultadosGauss.txt 1,1 Comienzo
```

En esta primera captura se utiliza el modo de escritura w, el cual crea el archivo y sobrescribe los datos.

```
9 #include <stdio.h>
8
7 int main(){
6
5     //Declarar variables
4     int n,res;
3
2     //Apuntador al archivo
1     FILE *a;
10    a=fopen("resultadosGauss.txt","a");
2
3     //Mensaje de bienvenida
4     printf("\n\n\t\t\tSuma de los primeros n números\n\n");
5
6     //Solicitar el número de elementos a sumar
7     printf("¿Cuántos números deseas sumar? ");
8     scanf("%d",&n);
9
10    //Sumar los n números
11    res=0;
12    for(int i=1;i<=n;i++)
13    {
14        fprintf(a,"%d + %d =",res,i);
15        res=res+i;
16        fprintf(a,"%d\n",res);
17    }
18
19    //Mostrar el resultado
20    printf("La suma de los primeros %d números es: %d \n",n,res);
21
22    fclose(a);
23
24    return 0;
25
26 }
```

```
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$ gcc -o gaussArchivos gaussArchivos
.c
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$ ./gaussArchivos

Suma de los primeros n números

¿Cuántos números deseas sumar? 100
La suma de los primeros 100 números es: 5050
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$ gcc -o gaussArchivos gaussArchivos
.c
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$ ./gaussArchivos

Suma de los primeros n números

¿Cuántos números deseas sumar? 10
La suma de los primeros 10 números es: 55
marko@marko-Inspiron-5567:~/Documentos/Facultad/Fundamentos d
e Programación/práctica13$
```

```
16 4371 + 94 =4465
15 4465 + 95 =4560
14 4560 + 96 =4656
13 4656 + 97 =4753
12 4753 + 98 =4851
11 4851 + 99 =4950
10 4950 + 100 =5050
9 0 + 1 =1
8 1 + 2 =3
7 3 + 3 =6
6 6 + 4 =10
5 10 + 5 =15
4 15 + 6 =21
3 21 + 7 =28
2 28 + 8 =36
1 36 + 9 =45
10 45 + 10 =55

resultadosGauss.txt 110,1 Final
```

En esta segunda captura se utiliza el modo de escritura a, el cual añadirá después del último carácter la nueva información, sin eliminar la anterior.

## Ejercicio 9: factorial de un número entero positivo mayor igual a cero

The image shows three terminal windows from the Alacritty application. The leftmost window displays the source code for a C program named 'factorial.c'. The code includes `<stdio.h>` and defines a `main()` function. It declares variables `n` and `r`, opens a file named 'factorial.txt' in write mode ('w'), and prompts the user to enter a number `n`. If `n` is 0, it prints 'El factorial de 0 es: 1'. Otherwise, it calculates the factorial of `n` using a loop and prints the result. The file is then closed. The status bar at the bottom indicates 'factorial.c' is 41 lines long and 603 characters, with the cursor at line 1, column 1.

The middle window shows the terminal output after compiling and running the program. It displays the prompt 'Ingresa un número entero n>=0 5', followed by the output 'El factorial de 5 es: 120'. The status bar at the bottom indicates 'factorial.txt' is 1 line long and 1 character, with the cursor at line 1, column 1.

The rightmost window shows the contents of the 'factorial.txt' file, which contains the output of the program for `n` values from 1 to 5: `(1)(1)=(1)`, `(1)(2)=(2)`, `(2)(3)=(6)`, `(6)(4)=(24)`, and `(24)(5)=(120)`. The status bar at the bottom indicates 'factorial.txt' is 1 line long and 1 character, with the cursor at line 1, column 1.

Para este programa se utilizó el modo de escritura en archivo w, ya que este crea el archivo en caso de no existir y escribe los datos en él borrando los existentes anteriormente, ya que considero que no es de mayor importancia guardar cada uno de los procesos, pues cuando analizamos un código o una operación matemática, solo buscamos los pasos de esta.

## Conclusión

El almacenamiento de datos en archivos permite crear programas más avanzados y que ofrezcan más posibilidades, como la creación y administración de una base de datos mediante un programa en C, para ello es importante conocer los diferentes modos en los que se puede abrir un archivo, ya que esto permitirá un mejor manejo de código y recursos.

## Referencias:

Laboratorio de Computación Salas A y B. (2020). Facultad de Ingeniería. *Manual de Prácticas de Fundamentos de Programación MADO-17 EP*. Recuperado de <http://lcp02.fi-b.unam.mx/>