



# Instituto Politécnico Nacional

## Escuela Superior de Cómputo

---

*Sistemas Operativos*

### *“Práctica 1. Introducción al sistema operativo Linux y Windows (1)”*

**Grupo:** 2CM9

**Integrantes:**

- Martínez Coronel Brayan Yosafat
- Monteros Cervantes Miguel Angel
- Ramírez Olvera Guillermo.
- Sánchez Méndez Edmundo Josue.

**Profesor:** Cortés Galicia Jorge



# **Práctica 1. Introducción al sistema operativo Linux y Windows (1)**

## **Introducción**

### **Linux**

Linux® es un sistema operativo (SO) de open source y una plataforma de infraestructura de TI. Originalmente, fue concebido y creado como un pasatiempo por Linus Torvalds pasatiempo de Linus Torvalds. Mientras estaba en la universidad, Linus intentó crear una versión de open source, alternativa y gratuita, del sistema operativo MINIX, que se basaba en los principios y el diseño de Unix. Desde entonces, ese pasatiempo se ha convertido en el SO con la base más grande de usuarios, en el SO más usado en servidores de Internet disponibles públicamente y en el único SO usado en las 500 principales supercomputadoras más rápidas.

Tal vez lo mejor sobre Linux es que es open source. Linux se lanza en virtud de la Licencia de uso público GNU (GPL). Esto significa que todos pueden ejecutar, estudiar, compartir y modificar el software. El código modificado también se puede redistribuir e incluso vender, pero todo esto se debe hacer con la misma licencia. Esto difiere en gran medida de los sistemas operativos tradicionales (por ejemplo, Unix y Windows) que son propietarios, están bloqueados, se entregan como están y no se pueden modificar.

### **Windows**

Windows es un sistema operativo desarrollado por la compañía de software Microsoft Corporation, que cuenta con una interfaz gráfica de usuario basada en el prototipo de windows (su nombre en inglés). Una ventana representa una tarea en ejecución, cada una puede contener su propio menú u otros controles, y el usuario puede acercar o alejar usando un dispositivo señalador como un mouse. La función principal es servir como un puente entre la persona y la máquina, facilitando así la conexión entre ambos.

Ofrece características pensadas y enfocadas en la realidad del usuario, es mucho más fácil y transparente de usar, además de configurar, ya que ofrece decenas de herramientas de autoconfiguración y corrección de problemas. Es importante mencionar que decidir qué sistema es el mejor es muy difícil, algunos optan por apple, otros por linux y otros por windows. En general, se dice que apple y linux son más estables y seguros, y windows es más fácil de usar.

## **1. Competencias.**

- El alumno aprende a familiarizarse con los sistemas operativos Linux y Windows mediante la exploración de sus ambientes de trabajo para desarrollar aplicaciones en lenguaje C.
- El alumno analiza el sistema operativo Linux y Windows mediante la comparación de sus características principales para diferenciarlos en su ambiente de trabajo.

## **2. Desarrollo en Linux:**

### **2.1. Existencia del sistema operativo**

El sistema operativo Linux se encuentra en correcto funcionamiento junto con el entorno gráfico y no hay necesidad de iniciar en modo texto.

### **2.2. Distribución de Linux utilizada**

Ubuntu 20.04.1 LTS.

#### **2.2.1. Tipos de distribuciones**

Linux tiene distintos tipos de distribuciones las cuales están diseñadas para distintas necesidades de los usuarios, algunas de estas distribuciones son:

- Ubuntu
- Red Hat Enterprise Linux
- Debian
- openSUSE
- Fedora
- CentOS
- Arch Linux
- Manjaro
- Linux Mint
- Elementary OS
- Zorin OS
- Peppermint OS
- Tails
- Kali Linux
- BlackArch Linux
- Arch Assault
- Puppy Linux
- Lubuntu
- Damn Small Linux
- SLiTaz
- LXLE
- Bodhi Linux
- Q4OS

- Distro Astro
- SteamOS
- ArtistX
- Ubuntu Studio
- Scientific Linux
- CEELD
- Edubuntu
- Openelec

### **2.2.2. Diferencias respecto a la distribución utilizada**

Dentro de las características que distingue a la distribución utilizada es la facilidad que tiene al ser ocupada por el usuario, a diferencia de otras distribuciones que son soportadas en un distinto tipo de hardware hasta software privado. Se pueden distinguir diferentes distribuciones para el nivel de usuario, a diferencia de la distribución ocupada hay para usuarios más avanzados.

## **2.3. Componentes del menú**

### **2.3.1. Componentes**

- LibreOffice Writer: Procesador de texto para la elaboración de documentos o archivos.
- LibreOffice Impress: Herramienta para elaborar presentaciones.
- LibreOffice Calc: Hoja de cálculo para elaboración de estadísticas.
- Editor de texto: Programa que permite crear y modificar archivos compuestos solo de texto.
- Terminal: Herramienta que permite llevar el control y ejecución de programas, también permite acceder a los directorios.
- Papelera: Programa donde se alojan archivos eliminados. Archivos: Programa donde se administra los directorios y archivos.

### **2.3.2. Diferencia de los componentes con los de Windows**

De las diferencias más notables está el diseño gráfico por parte de Windows tiene un mejor diseño en la elaboración de documentos o presentaciones, aunque actualmente varias distribuciones de Linux intentan asemejar el diseño gráfico de Windows. Por otro lado, cada componente cumple con la función para la cual se desarrolló además de que en Linux todo es gratuito por el contrario en Windows que la mayoría de las aplicaciones son de pago. Componentes con función similar:

- LibreOffice Writer: Microsoft Office.
- LibreOffice Impress: Microsoft PowerPoint.
- LibreOffice Calc: Microsoft Excel.

- Editor de texto: Bloc de notas.
- Terminal: Administrador: símbolo del sistema.
- Papelera: Papelera.
- Archivos: Explorador de archivos.

## 2.4. Función de los comandos

- **ls**: Muestra los directorios y archivos de la carpeta personal.
- **ls -l**: Muestra los directorios y archivos de la carpeta actual con sus propiedades y atributos.
- **ls -a**: Muestra los directorios y archivos de la carpeta actual, incluidos los ocultos y los del sistema.
- **ls -F**: Muestra los directorios y archivos de la carpeta actual, mostrando al final de cada directorio el carácter / diferenciando carpetas de archivos.
- **ls -t**: Muestra los directorios y archivos de la carpeta personal ordenados por fecha de última modificación.
- **vi**: Crea y permite editar un nuevo archivo en el directorio actual.
- **clear**: Limpia la pantalla y recorre al inicio de la consola el prompt.
- **cd -**: Regresa a la ubicación anterior.
- **cd ..**: Regresa un nivel a la raíz del directorio.
- **cd**: Recorre a la raíz del directorio.
- **cd nombre-directorio**: Cambia al directorio indicado (Debe de encontrarse en la ubicación actual).
- **chmod**: Permite cambiar los permisos de acceso de un archivo o directorio.
- **cat nombre-archivo**: Muestra el contenido del archivo en pantalla.
- **grep**: Busca patrones en archivos.
- **rm nombre-archivo**: Elimina el archivo.
- **rm -rf nombre-directorio**: Elimina el directorio y su contenido.
- **rmdir nombre-directorio**: Elimina el directorio, debe estar no vacío.
- **ps**: Muestra la información acerca de los procesos en ejecución.
- **ps -aux**: Lista los programas que se están ejecutando.
- **ps -ef | grep nombre-proceso**: Muestra los procesos relacionados con nombre-proceso.
- **cp nombre-archivo nombre-directorio**: Copia el archivo en el directorio indicado.

- **cp nombre-archivo1 nombre-archivo2:** Copia el archivo1 y lo renombra con el nombre del archivo2.
- **cp nombre-archivo /nombre-directorio:** Copia el archivo con el mismo nombre en el directorio.
- **mv nombre-archivo1 nombre-archivo2:** Renombra el archivo1 por el nombre del archivo2.
- **mv archivo /directorio:** Mueve el archivo dentro del directorio.
- **mkdir nombre-directorio:** Crea un nuevo directorio con el nombre indicado.
- **pwd:** Muestra el nombre del directorio actual.
- **whoami:** Imprime en pantalla el nombre del usuario.
- **whoami --help:** Muestra información acerca del comando.

## 2.5. Ejecución de comandos

- Comando ls

```
edmundojism@edmundojism-VB:~$ ls
Descargas  Escritorio  Música      Público
Documentos Imágenes    Plantillas  Vídeos
edmundojism@edmundojism-VB:~$
```

- Comando ls -l

```
edmundojism@edmundojism-VB:~$ ls -l
total 32
drwxr-xr-x 2 edmundojism edmundojism 4096 sep 21 19:09 Descargas
drwxr-xr-x 2 edmundojism edmundojism 4096 sep 21 19:09 Documentos
drwxr-xr-x 3 edmundojism edmundojism 4096 sep 29 12:26 Escritorio
drwxr-xr-x 2 edmundojism edmundojism 4096 sep 21 19:09 Imágenes
drwxr-xr-x 2 edmundojism edmundojism 4096 sep 21 19:09 Música
drwxr-xr-x 2 edmundojism edmundojism 4096 sep 21 19:09 Plantillas
drwxr-xr-x 2 edmundojism edmundojism 4096 sep 21 19:09 Público
drwxr-xr-x 2 edmundojism edmundojism 4096 sep 21 19:09 Vídeos
edmundojism@edmundojism-VB:~$
```

- Comando ls -la

```

edmundojism@edmundojism-VB:~$ ls -la
total 80
drwxr-xr-x 17 edmundojism edmundojism 4096 sep 23 13:41 .
drwxr-xr-x  3 root        root        4096 sep 21 18:49 ..
-rw-r--r--  1 edmundojism edmundojism  220 sep 21 18:49 .bash_logout
-rw-r--r--  1 edmundojism edmundojism 3771 sep 21 18:49 .bashrc
drwxrwxr-x  4 edmundojism edmundojism 4096 sep 23 13:41 .bluefish
drwx----- 15 edmundojism edmundojism 4096 sep 23 13:48 .cache
drwx----- 14 edmundojism edmundojism 4096 sep 23 13:48 .config
drwxr-xr-x  2 edmundojism edmundojism 4096 sep 21 19:09 Descargas
drwxr-xr-x  2 edmundojism edmundojism 4096 sep 21 19:09 Documentos
drwxr-xr-x  3 edmundojism edmundojism 4096 sep 29 12:26 Escritorio
drwx-----  3 edmundojism edmundojism 4096 sep 29 07:50 .gnupg
drwxr-xr-x  2 edmundojism edmundojism 4096 sep 21 19:09 Imágenes
drwx-----  3 edmundojism edmundojism 4096 sep 21 19:08 .local
drwx-----  5 edmundojism edmundojism 4096 sep 23 13:14 .mozilla
drwxr-xr-x  2 edmundojism edmundojism 4096 sep 21 19:09 Música
drwxr-xr-x  2 edmundojism edmundojism 4096 sep 21 19:09 Plantillas
-rw-r--r--  1 edmundojism edmundojism  807 sep 21 18:49 .profile
drwxr-xr-x  2 edmundojism edmundojism 4096 sep 21 19:09 Público
drwx-----  2 edmundojism edmundojism 4096 sep 23 13:16 .ssh
-rw-r--r--  1 edmundojism edmundojism    0 sep 23 13:16 .sudo_as_admin_successful
drwxr-xr-x  2 edmundojism edmundojism 4096 sep 21 19:09 Vídeos
edmundojism@edmundojism-VB:~$

```

- Comando pwd

```

edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos$ pwd
/home/edmundojism/Escritorio/Sistemas Operativos
edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos$ █

```

- Comando clear

```

edmundojism@edmundojism-VB:~$ cd Escritorio
edmundojism@edmundojism-VB:~/Escritorio$ cd Sistemas\ Operativos/
edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos$ pwd
/home/edmundojism/Escritorio/Sistemas Operativos
edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos$

```

Figura 1: Antes de ejecutar el comando clear

```

edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos$ █

```

Figura 2: Después de ejecutar el comando clear

- Comando cd nombre-directorio

```

edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos$ cd Practica\ 1/
edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 1$ █

```

- Comando cat nombre-archivo



```

edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 1$ cat HolaMundo.c
#include <stdio.h>

int main(){
    printf("Hola mundo\n");
    return 0;
}

edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 1$ █

```

- Comando ls -la|more

```

edmundojism@edmundojism-VB:~$ ls -la|more
total 80
drwxr-xr-x 17 edmundojism edmundojism 4096 sep 23 13:41 .
drwxr-xr-x  3 root        root        4096 sep 21 18:49 ..
-rw-r--r--  1 edmundojism edmundojism 220  sep 21 18:49 .bash_logout
-rw-r--r--  1 edmundojism edmundojism 3771 sep 21 18:49 .bashrc
drwxrwxr-x  4 edmundojism edmundojism 4096 sep 23 13:41 .bluefish
drwx----- 15 edmundojism edmundojism 4096 sep 23 13:48 .cache
drwx----- 15 edmundojism edmundojism 4096 sep 29 14:34 .config
drwxr-xr-x  2 edmundojism edmundojism 4096 sep 21 19:09 Descargas
drwxr-xr-x  2 edmundojism edmundojism 4096 sep 21 19:09 Documentos
drwxr-xr-x  3 edmundojism edmundojism 4096 sep 29 12:26 Escritorio
drwx-----  3 edmundojism edmundojism 4096 sep 29 14:14 .gnupg
drwxr-xr-x  2 edmundojism edmundojism 4096 sep 21 19:09 Imágenes
drwx-----  3 edmundojism edmundojism 4096 sep 21 19:08 .local
drwx-----  5 edmundojism edmundojism 4096 sep 23 13:14 .mozilla
drwxr-xr-x  2 edmundojism edmundojism 4096 sep 21 19:09 Música
drwxr-xr-x  2 edmundojism edmundojism 4096 sep 21 19:09 Plantillas
-rw-r--r--  1 edmundojism edmundojism 807  sep 21 18:49 .profile
drwxr-xr-x  2 edmundojism edmundojism 4096 sep 21 19:09 Público
drwx-----  2 edmundojism edmundojism 4096 sep 23 13:16 .ssh
-rw-r--r--  1 edmundojism edmundojism   0  sep 23 13:16 .sudo_as_admin_successful
drwxr-xr-x  2 edmundojism edmundojism 4096 sep 21 19:09 Vídeos
edmundojism@edmundojism-VB:~$ █

```

- Comando rm nombre\_archivo

```

edmundojism@edmundojism-VB:~$ ls
Descargas Documentos Escritorio HolaMundo.c Imágenes Música Plantillas Público Vídeos
edmundojism@edmundojism-VB:~$ rm HolaMundo.c
edmundojism@edmundojism-VB:~$ ls
Descargas Documentos Escritorio Imágenes Música Plantillas Público Vídeos
edmundojism@edmundojism-VB:~$ █

```

- Comando ps

```

edmundojism@edmundojism-VB:~$ ps
  PID TTY          TIME CMD
 1696 pts/0    00:00:00 bash
 2463 pts/0    00:00:00 ps
edmundojism@edmundojism-VB:~$ █

```

- Comando cp [ruta1] archivo1 [ruta2] archivo2

```

edmundojism@edmundojism-VB:~$ ls
Descargas Documentos Escritorio HolaMundo.c Imágenes Música Plantillas Público Vídeos
edmundojism@edmundojism-VB:~$ cp HolaMundo.c HolaMundoCopia.c
edmundojism@edmundojism-VB:~$ ls
Descargas Documentos Escritorio HolaMundo.c HolaMundoCopia.c Imágenes Música Plantillas Público Vídeos
edmundojism@edmundojism-VB:~$ █

```

- Comando mv [ruta1] archivo1 [ruta2] archivo2



```
edmundojism@edmundojism-VB:~$ ls
Descargas Documentos Escritorio HolaMundo.c Imágenes Música Plantillas Público Vídeos
edmundojism@edmundojism-VB:~$ mv HolaMundo.c HolaMundo.h
edmundojism@edmundojism-VB:~$ ls
Descargas Documentos Escritorio HolaMundo.h Imágenes Música Plantillas Público Vídeos
edmundojism@edmundojism-VB:~$
```

- Comando mkdir nombre\_directorio

```
edmundojism@edmundojism-VB:~$ ls
Descargas Documentos Escritorio HolaMundo.h Imágenes Música Plantillas Público Vídeos
edmundojism@edmundojism-VB:~$ mkdir SistemasOperativos
edmundojism@edmundojism-VB:~$ ls
Descargas Escritorio Imágenes Plantillas SistemasOperativos
Documentos HolaMundo.h Música Público Vídeos
edmundojism@edmundojism-VB:~$
```

- Comando rmdir nombre\_directorio

```
edmundojism@edmundojism-VB:~$ ls
Descargas Escritorio Imágenes Plantillas SistemasOperativos
Documentos HolaMundo.h Música Público Vídeos
edmundojism@edmundojism-VB:~$ rmdir SistemasOperativos
edmundojism@edmundojism-VB:~$ ls
Descargas Documentos Escritorio HolaMundo.h Imágenes Música Plantillas Público Vídeos
edmundojism@edmundojism-VB:~$
```

- Comando whoami

```
edmundojism@edmundojism-VB:~$ whoami
edmundojism
edmundojism@edmundojism-VB:~$
```

## 2.6. Programas

### 2.6.1. Hola mundo

Código (HolaMundo.c)

```
01. #include <stdio.h>
02.
03. int main(){
04.     printf("Hola mundo\n");
05.     return 0;
06. }
```

Compilación y ejecución:

```
edmundojism@edmundojism-VB: ~/Escritorio/Sistemas Operativos/Practica 1
edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 1$ gcc HolaMundo.c
-o HolaMundo
edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 1$ ./HolaMundo
Hola mundo
edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 1$
```

## 2.6.3. Rombos

### Código (Rombo.c)

```
01. #include <stdio.h>
02. #include <string.h>
03.
04. void imprimeChar(char c, int n, FILE * out){
05.     int i;
06.     for(i = 1; i <= n; i++){
07.         fprintf(out, "%c", c);
08.     }
09. }
10.
11. void imprimeFigura(int n, FILE * out){
12.     int mitad = (n - 1) / 2;
13.     int i, j;
14.     for (i = 1; i <= mitad; i++)
15.     {
16.         imprimeChar(' ', mitad, out);
17.         imprimeChar('*', i, out);
18.         imprimeChar(' ', 2*(mitad-i+1), out);
19.         imprimeChar('*', 2*i-1, out);
20.         imprimeChar(' ', 2*(mitad-i+1), out);
21.         imprimeChar('*', i, out);
22.         fprintf(out, "\n");
23.     }
24.     imprimeChar('*', 3 * n, out);
25.     fprintf(out, "\n");
26.     for (i = 1; i <= mitad; i++)
27.     {
28.         imprimeChar(' ', mitad, out);
29.         imprimeChar('*', mitad-i+1, out);
30.         imprimeChar(' ', 2*i, out);
31.         imprimeChar('*', n-2*i, out);
32.         imprimeChar(' ', 2*i, out);
33.         imprimeChar('*', mitad-i+1, out);
34.         fprintf(out, "\n");
35.     }
36. }
37.
38. int main(){
39.     int n, opcion;
40.     printf("Introduce el ancho del rombo (impar): ");
41.     scanf("%d", &n);
42.     imprimeFigura(n, stdout);
43.     printf("Guardar en archivo? Si=1/No=0:");
44.     scanf("%d", &opcion);
45.     if(opcion == 1){
46.         char nombre[100];
47.         printf("Nombre del archivo: ");
48.         scanf("%s", nombre);
49.         FILE * archivo = fopen(strcat( nombre, ".txt"), "w");
50.         imprimeFigura(n, archivo);
51.         fclose(archivo);
52.         printf("Archivo creado con exito\n");
53.         printf("El programa a finalizado");
54.     }else{
55.         printf("El programa a finalizado");
56.     }
57.     return 0;
58. }
```

Compilación y ejecución para un ancho de rombo  $n = 11$ , mostrando en pantalla y luego exportando a un archivo:

```

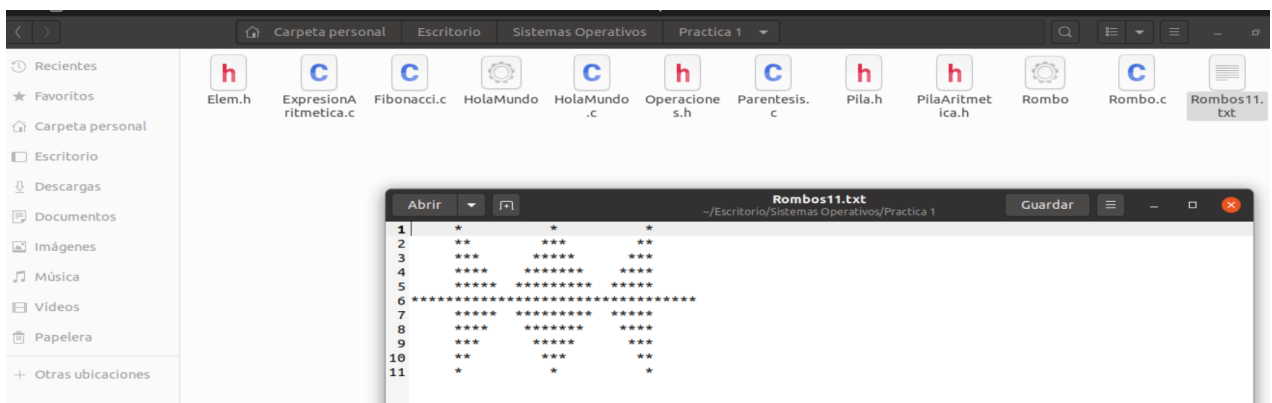
edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 1$ gcc Rombo.c -o Rombo
edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 1$ ./Rombo
Introduce el ancho del rombo (impar): 11

  *      *      *
 **     ***     **
 ***    *****    ***
 ****   *         *   ****
 ***** *         * *****
*****
***** *         * *****
 ****   *         *   ****
 ***    *****    ***
 **     ***     **
 *      *      *

Guardar en archivo? Si=1/No=0:1
Nombre del archivo: Rombos11
Archivo creado con exito
El programa a finalizadoedmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 1$

```

Vemos que el archivo se crea correctamente



### 2.6.3. Serie de Fibonacci

Código (Fibonacci.c)

```

01. #include <stdio.h>
02.
03. int FibonacciRecursivo(int n)
04. {
05.     if (n <= 1)
06.         return n;
07.     return FibonacciRecursivo(n-1) + FibonacciRecursivo(n-2);
08. }
09.
10. int main()
11. {
12.     int n, i;
13.     printf("Introduce el numero para la serie de Fibonacci ");
14.     scanf("%d", &n);
15.     for(i=0 ;n>=i;i++){
16.         printf("%d ", FibonacciRecursivo(i));
17.     }
18. }

```

Compilación y ejecución para obtener la serie Fibonacci del número 7 (se escribe el 0 por definición de la serie de Fibonacci):

```

edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 1$ gcc Fibonacci.c -o Fibonacci
edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 1$ ./Fibonacci
Introduce el numero para la serie de Fibonacci 7
0 1 1 2 3 5 8 13 edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 1$

```

## 2.6.4. Comprobación de emparejamiento de paréntesis

### Código parte 1 (Pila.h)

```
01. typedef struct Nodo
02. {
03.     Elem dato;
04.     struct Nodo *sig;
05. } * Pila;
06.
07. Pila empty()
08. {
09.     return NULL;
10. }
11.
12. Pila push(Elem e, Pila p)
13. {
14.     Pila x = (Pila)malloc(sizeof(struct Nodo));
15.     x->dato = e;
16.     x->sig = p;
17.     return x;
18. }
19.
20. int isempty(Pila p)
21. {
22.     return p == NULL;
23. }
24.
25. Elem top(Pila p)
26. {
27.     return p->dato;
28. }
29.
30. Pila pop(Pila p)
31. {
32.     return p->sig;
33. }
```

### Código parte 2 (Elem.h)

```
01. #include<stdio.h>
02.
03. typedef char Elem;
04.
05. void ImpElem(Elem e){
06.     printf("%c", e);
07. }
08.
09. int EsDigito(char x){
10.     return (x>='0') && (x<='9');
11. }
12.
13. int EsLetra(char x){
14.     return ((x>='a')&&(x<='z'))||((x>='A')&&(x<='Z'));
15. }
16.
17.
18. Elem EsOperando(char x){
19.     return EsDigito(x)|| EsLetra(x);
20. }
21.
22. int Prec(char x){
23.     switch(x){
24.         case '^': return 3;
25.         case '*': case '/': return 2;
26.         case '+': case '-': return 1;
27.         case '(': return 0;
28.         default: return -1;
29.     }
30. }
```

## Código parte 3 (Parentesis.c)

```
01. #include <stdio.h>
02. #include <stdlib.h>
03. #include "Elem.h"
04. #include "Pila.h"
05.
06. char evaluar(char * expresion){
07.     int i = 0, pos = -1, j;
08.     int ver;
09.     char apertura[3] = {'(', '{', '['};
10.     char cierre[3] = {'}', '}', ']'};
11.     Pila op = empty();
12.     while (expresion[i] != '\0')
13.     {
14.         char actual = expresion[i];
15.         for (j = 0; j < 3; j++)
16.         {
17.             if (actual == apertura[j])
18.             {
19.                 op = push(apertura[j], op);
20.                 pos = pos + 1;
21.             }
22.             else if (actual == cierre[j])
23.             {
24.                 if (pos == -1)
25.                 {
26.                     printf("El simbolo %c cierra pero no abre.\n", actual);
27.                     return 0;
28.                 }
29.                 else
30.                 {
31.                     char tope = top(op);
32.                     pos = pos - 1;
33.                     op = pop(op);
34.                     if (tope != apertura[j])
35.                     {
36.                         printf("Se abre un %c pero cierra un %c.\n", tope, actual);
37.                         return 0;
38.                     }
39.                 }
40.             }
41.         }
42.         i++;
43.     }
44.     if (pos != -1)
45.     {
46.         printf("Falto cerrar los simbolos: ");
47.         while (pos >= 0)
48.         {
49.             printf("%c", top(op));
50.             pos = pos - 1;
51.             op = pop(op);
52.         }
53.         printf("\n");
54.         return 0;
55.     }
56.     return 1;
57. }
58. int main(int argc)
59. {
60.     char c, *expresion;
61.     int n = 0;
62.     expresion = (char *)malloc(1 * sizeof(char));
63.     printf("Ingresa una expresion: ");
64.     do
65.     {
66.         c = getchar();
67.         expresion[n] = c;
68.         n++;
69.         expresion = (char *)realloc(expresion, (n + 1) * sizeof(char));
70.     } while (c != '\n');
71.     expresion[n - 1] = '\0';
72.     if (evaluar(expresion)){
73.         printf("Expresion correcta.\n");
74.     }else{
75.         printf("Expresion incorrecta.\n");
76.     }
77.
78.     return 0;
79. }
```

Compilación y ejecución para resolver cinco casos de prueba, tres correctos y dos incorrectos:

```
edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 1$ gcc Parentesis.c -o Parentesis
edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 1$ ./Parentesis
Ingrese una expresion: {[()()](())}
Expresion correcta.
edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 1$ ./Parentesis
Ingrese una expresion: [(())[{}][{}]]
Expresion correcta.
edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 1$ ./Parentesis
Ingrese una expresion: (([[]])
Se abre un ( pero cierra un ].
Expresion incorrecta.
edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 1$ ./Parentesis
Ingrese una expresion: {(())[{}][
Falto cerrar los simbolos: [(
Expresion incorrecta.
edmundojism@edmundojism-VB:~/Escritorio/Sistemas Operativos/Practica 1$ ./Parentesis
Ingrese una expresion: [{()}(){}(){}[{}]]{}
Expresion correcta.
```

## 2.6.5. Evaluación de expresiones aritméticas

### Código parte 1 (PilaAritmetica.h)

```
01. typedef struct Pila{
02.     int pos;
03.     double A[100];
04. } Pila;
05.
06. Pila * empty(){
07.     Pila * p = calloc(1, sizeof(Pila));
08.     p->pos = -1;
09.     return p;
10. }
11.
12. int isempty(Pila * p){
13.     return p->pos < 0;
14. }
15.
16. void push(Pila * p, double x){
17.     p->A[++p->pos] = x;
18. }
19.
20. double top(Pila * p){
21.     return p->A[p->pos];
22. }
23.
24. double pop(Pila * p){
25.     return p->A[p->pos--];
26. }
```

### Código parte 2 (Operaciones.h)

```
01. #include<stdio.h>
02.
03. int esOperador(char c){
04.     return c == '+' || c == '-' || c == '*' || c == '/' || c == '^';
05. }
06.
07. int esDigito(char c){
08.     return ('0' <= c && c <= '9') || c == '.';
09. }
10.
11. int precedencia(char c){
12.     switch(c){
13.         case '^': return 3;
14.         case '*': case '/': return 2;
15.         case '+': case '-': return 1;
16.         case '(': return 0;
17.         default: return -1;
18.     }
19. }
```



```

20.
21. double operacion(double operando1, double operando2, char operador){
22.     switch(operador){
23.         case '+': return operando1 + operando2;
24.         case '-': return operando1 - operando2;
25.         case '*': return operando1 * operando2;
26.         case '/': return operando1 / operando2;
27.         case '^': return pow(operando1, operando2);
28.         default: return 0;
29.     }
30. }

```

### Código parte 3 (ExpresionAritmetica.c)

```

01. #include <stdio.h>
02. #include <stdlib.h>
03. #include <string.h>
04. #include <math.h>
05. #include "PilaAritmetica.h"
06. #include "Operaciones.h"
07.
08. double evaluar(char * expresion){
09.     Pila * operadores = empty();
10.     Pila * operandos = empty();
11.     int tamaño = strlen(expresion);
12.     int i = 0;
13.     double operando = 0;
14.     char Stroperando[100];
15.     double operando1, operando2;
16.     while(expresion[i]!='\0'){
17.         char actual = expresion[i];
18.         if(esDigito(actual)){
19.             int j = 0;
20.             while(i < tamaño && esDigito(expresion[i])){
21.                 Stroperando[j++] = expresion[i++];
22.             }
23.             Stroperando[j] = '\0';
24.             sscanf(Stroperando, "%lf", &operando);
25.             push(operandos, operando);
26.             continue;
27.         }else if(actual == '('){
28.             push(operadores, actual);
29.         }else if(actual == ')'){
30.             while(top(operadores) != '('){
31.                 operando2 = pop(operandos);
32.                 operando1 = pop(operandos);
33.                 push(operandos, operacion(operando1, operando2, pop(operadores)));
34.             }
35.             pop(operadores);
36.         }else if(esOperador(actual)){
37.             while(!isempty(operadores) && precedencia(top(operadores)) >= precedencia(actual)){
38.                 operando2 = pop(operandos);
39.                 operando1 = pop(operandos);
40.                 push(operandos, operacion(operando1, operando2, pop(operadores)));
41.             }
42.             push(operadores, actual);
43.         }
44.         i++;
45.     }
46.     while(!isempty(operadores)){
47.         operando2 = pop(operandos);
48.         operando1 = pop(operandos);
49.         push(operandos, operacion(operando1, operando2, pop(operadores)));
50.     }
51.     return pop(operandos);
52. }
53.
54. int main(){
55.     char expresion[100];
56.     printf("Introduce la expresion aritmetica: ");
57.     scanf("%s", expresion);
58.     printf("%.5f\n", evaluar(expresion));
59.     return 0;
60. }

```

Compilación y ejecución para cinco casos de prueba:

```
edmundojm@edmundojm-VB:~/Escritorio/Sistemas Operativos/Practica 1$ gcc ExpresionAritmetica.c -o ExpresionAritmetica -lm
edmundojm@edmundojm-VB:~/Escritorio/Sistemas Operativos/Practica 1$ ./ExpresionAritmetica
Introduce la expresion aritmetica: (((5-2)+4)*10)
70.00000
edmundojm@edmundojm-VB:~/Escritorio/Sistemas Operativos/Practica 1$ ./ExpresionAritmetica
Introduce la expresion aritmetica: 1+2*5-7/(3+4)
10.00000
edmundojm@edmundojm-VB:~/Escritorio/Sistemas Operativos/Practica 1$ ./ExpresionAritmetica
Introduce la expresion aritmetica: 5^(1+5-2*2)+3*(10*(5*3-10)-4/6)
173.00000
edmundojm@edmundojm-VB:~/Escritorio/Sistemas Operativos/Practica 1$ ./ExpresionAritmetica
Introduce la expresion aritmetica: 5.4*(3-(5-8*2))+4^(1/(1+1))
77.60000
edmundojm@edmundojm-VB:~/Escritorio/Sistemas Operativos/Practica 1$ ./ExpresionAritmetica
Introduce la expresion aritmetica: 5.5*9.6-(4.12*(3.1416-2.36^3))+9.04
103.05094
edmundojm@edmundojm-VB:~/Escritorio/Sistemas Operativos/Practica 1$
```

### 3. Desarrollo Windows

#### 3.1. Comandos

- **DIR:** Muestra una lista de archivos y subdirectorios en un directorio.
- **ipconfig:** Permite conocer la dirección IP y otra información relacionada con la configuración de internet del ordenador.
- **Cls:** Borra los símbolos o el texto en la pantalla de la consola.
- **Ver:** Muestra la versión de Windows.
- **Tree:** Muestra gráficamente la estructura de directorios de una unidad o ruta de acceso.
- **cd nombre-directorio:** Cambia al directorio escrito en el comando.
- **Type nombre-archivo:** Muestra el contenido del archivo.
- **FType:** Muestra o modifica los tipos de archivo usados en una asociación de extensión de archivo.
- **Mkdir nombre-directorio:** Crea un directorio con el nombre escrito en “nombre-archivo”.
- **Rmdir nombre-directorio:** Elimina el directorio. del nombre-archivo: Elimina el archivo.
- **Copy nombre-archivo nuevo-nombre-archivo:** Copia el archivo y lo renombra con el nuevo nombre.
- **XCOPY:** Copia archivos y arboles de directorios.
- **Ren nombre-archivo nuevo-nombre-archivo:** Cambia el nombre del archivo por el nuevo nombre.
- **Chdir nombre-directorio:** Cambia al directorio indicado.
- **echo “Hola mundo”:** Muestra el mensaje “Hola mundo”.

- Find “cadena-buscar” nombre-archivo: Busca la cadena en el archivo.

## 3.2 Ejecución de comandos

- Comando Dir

```
C:\Users\Edmundo J Sanchez M\Desktop\SO>dir
El volumen de la unidad C es Windows-SSD
El número de serie del volumen es: 8A89-DEDC

Directorio de C:\Users\Edmundo J Sanchez M\Desktop\SO

28/09/2020  09:03 p. m.  <DIR>      .
28/09/2020  09:03 p. m.  <DIR>      ..
29/09/2020  06:07 p. m.  <DIR>      P1
                0 archivos                0 bytes
                3 dirs 62,353,117,184 bytes libres
```

- Comando ipconfig

```
C:\Users\Edmundo J Sanchez M\Desktop\SO>ipconfig

Configuración IP de Windows

Adaptador de Ethernet VirtualBox Host-Only Network:

    Sufixo DNS específico para la conexión. . . :
    Vínculo: dirección IPv6 local. . . . : fe80::d143:8446:10fc:e0be%6
    Dirección IPv4. . . . . : 192.168.56.1
    Máscara de subred. . . . . : 255.255.255.0
    Puerta de enlace predeterminada. . . . . :

Adaptador de LAN inalámbrica Conexión de área local* 1:

    Estado de los medios. . . . . : medios desconectados
    Sufixo DNS específico para la conexión. . . :

Adaptador de LAN inalámbrica Conexión de área local* 2:

    Estado de los medios. . . . . : medios desconectados
    Sufixo DNS específico para la conexión. . . :

Adaptador de LAN inalámbrica Wi-Fi:

    Sufixo DNS específico para la conexión. . . :
    Vínculo: dirección IPv6 local. . . . : fe80::c4eb:343d:484f:1b2d%12
    Dirección IPv4. . . . . : 192.168.50.231
    Máscara de subred. . . . . : 255.255.255.0
    Puerta de enlace predeterminada. . . . . : 192.168.50.1

Adaptador de Ethernet Conexión de red Bluetooth:

    Estado de los medios. . . . . : medios desconectados
    Sufixo DNS específico para la conexión. . . :
```

- Comando cls

```
C:\Users\Edmundo J Sanchez M\Desktop\SO>ver

Microsoft Windows [Versión 10.0.18362.1082]

C:\Users\Edmundo J Sanchez M\Desktop\SO>Tree
Listado de rutas de carpetas para el volumen Windows-SSD
El número de serie del volumen es 8A89-DEDC
C:.
└──P1

C:\Users\Edmundo J Sanchez M\Desktop\SO>
```

Figura 1: Antes de ejecutar el comando cls

```
C:\Users\Edmundo J Sanchez M\Desktop\S0>
```

Figura 2: Después de ejecutar el comando cls

- Comando ver

```
C:\Users\Edmundo J Sanchez M\Desktop\S0>ver  
Microsoft Windows [Versión 10.0.18362.1082]
```

- Comando Tree

```
C:\Users\Edmundo J Sanchez M\Desktop\S0>Tree  
Listado de rutas de carpetas para el volumen Windows-SSD  
El número de serie del volumen es 8A89-DEDC  
C:.  
└─P1
```

- Comando cd

```
C:\Users\Edmundo J Sanchez M\Desktop\S0>cd P1  
C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>
```

- Comando Type

```
C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>Type HolaMundo.c  
#include <stdio.h>  
  
int main(){  
    printf("Hola mundo\n");  
    return 0;  
}  
  
C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>
```

- Comando Mkdir

```
C:\Users\Edmundo J Sanchez M\Desktop\S0>dir  
El volumen de la unidad C es Windows-SSD  
El número de serie del volumen es: 8A89-DEDC  
  
Directorio de C:\Users\Edmundo J Sanchez M\Desktop\S0  
  
28/09/2020 09:03 p. m. <DIR> .  
28/09/2020 09:03 p. m. <DIR> ..  
29/09/2020 06:07 p. m. <DIR> P1  
0 archivos 0 bytes  
3 dirs 62,353,825,792 bytes libres  
  
C:\Users\Edmundo J Sanchez M\Desktop\S0>mkdir CarpetaPrueba  
  
C:\Users\Edmundo J Sanchez M\Desktop\S0>dir  
El volumen de la unidad C es Windows-SSD  
El número de serie del volumen es: 8A89-DEDC  
  
Directorio de C:\Users\Edmundo J Sanchez M\Desktop\S0  
  
29/09/2020 08:35 p. m. <DIR> .  
29/09/2020 08:35 p. m. <DIR> ..  
29/09/2020 08:35 p. m. <DIR> CarpetaPrueba  
29/09/2020 06:07 p. m. <DIR> P1  
0 archivos 0 bytes  
4 dirs 62,352,515,072 bytes libres  
  
C:\Users\Edmundo J Sanchez M\Desktop\S0>
```

- Comando rmdir

```
C:\Users\Edmundo J Sanchez M\Desktop\S0>dir
El volumen de la unidad C es Windows-SSD
El número de serie del volumen es: 8A89-DEDC

Directorio de C:\Users\Edmundo J Sanchez M\Desktop\S0

29/09/2020  08:35 p. m.    <DIR>        .
29/09/2020  08:35 p. m.    <DIR>        ..
29/09/2020  08:35 p. m.    <DIR>        CarpetaPrueba
29/09/2020  06:07 p. m.    <DIR>        P1
                0 archivos                0 bytes
                4 dirs  62,352,515,072 bytes libres

C:\Users\Edmundo J Sanchez M\Desktop\S0>rmdir CarpetaPrueba

C:\Users\Edmundo J Sanchez M\Desktop\S0>dir
El volumen de la unidad C es Windows-SSD
El número de serie del volumen es: 8A89-DEDC

Directorio de C:\Users\Edmundo J Sanchez M\Desktop\S0

29/09/2020  08:36 p. m.    <DIR>        .
29/09/2020  08:36 p. m.    <DIR>        ..
29/09/2020  06:07 p. m.    <DIR>        P1
                0 archivos                0 bytes
                3 dirs  62,354,345,984 bytes libres

C:\Users\Edmundo J Sanchez M\Desktop\S0>
```

- Comando del

```
C:\Users\Edmundo J Sanchez M\Desktop\S0>dir
El volumen de la unidad C es Windows-SSD
El número de serie del volumen es: 8A89-DEDC

Directorio de C:\Users\Edmundo J Sanchez M\Desktop\S0

29/09/2020  08:37 p. m.    <DIR>        .
29/09/2020  08:37 p. m.    <DIR>        ..
28/09/2020  08:52 a. m.    <FILE>        247 Clases.txt
29/09/2020  06:07 p. m.    <DIR>        P1
                1 archivos                247 bytes
                3 dirs  62,354,477,056 bytes libres

C:\Users\Edmundo J Sanchez M\Desktop\S0>del Clases.txt

C:\Users\Edmundo J Sanchez M\Desktop\S0>dir
El volumen de la unidad C es Windows-SSD
El número de serie del volumen es: 8A89-DEDC

Directorio de C:\Users\Edmundo J Sanchez M\Desktop\S0

29/09/2020  08:37 p. m.    <DIR>        .
29/09/2020  08:37 p. m.    <DIR>        ..
29/09/2020  06:07 p. m.    <DIR>        P1
                0 archivos                0 bytes
                3 dirs  62,353,432,576 bytes libres

C:\Users\Edmundo J Sanchez M\Desktop\S0>
```

- Comando Copy

```

C:\Users\Edmundo J Sanchez M\Desktop\SO>dir
El volumen de la unidad C es Windows-SSD
El número de serie del volumen es: 8A89-DEDC

Directorio de C:\Users\Edmundo J Sanchez M\Desktop\SO

29/09/2020  08:39 p. m.    <DIR>        .
29/09/2020  08:39 p. m.    <DIR>        ..
29/09/2020  06:07 p. m.    <DIR>        P1
01/09/2020  01:34 a. m.                559 Prueba.py
                1 archivos                559 bytes
                3 dirs  62,354,722,816 bytes libres

C:\Users\Edmundo J Sanchez M\Desktop\SO>Copy Prueba.py ArchivoCopia.py
1 archivo(s) copiado(s).

C:\Users\Edmundo J Sanchez M\Desktop\SO>dir
El volumen de la unidad C es Windows-SSD
El número de serie del volumen es: 8A89-DEDC

Directorio de C:\Users\Edmundo J Sanchez M\Desktop\SO

29/09/2020  08:39 p. m.    <DIR>        .
29/09/2020  08:39 p. m.    <DIR>        ..
01/09/2020  01:34 a. m.                559 ArchivoCopia.py
29/09/2020  06:07 p. m.    <DIR>        P1
01/09/2020  01:34 a. m.                559 Prueba.py
                2 archivos                1,118 bytes
                3 dirs  62,354,284,544 bytes libres

C:\Users\Edmundo J Sanchez M\Desktop\SO>

```

## ▪ Comando Ren

```

C:\Users\Edmundo J Sanchez M\Desktop\SO>dir
El volumen de la unidad C es Windows-SSD
El número de serie del volumen es: 8A89-DEDC

Directorio de C:\Users\Edmundo J Sanchez M\Desktop\SO

29/09/2020  08:40 p. m.    <DIR>        .
29/09/2020  08:40 p. m.    <DIR>        ..
29/09/2020  06:07 p. m.    <DIR>        P1
01/09/2020  01:34 a. m.                559 Prueba.py
                1 archivos                559 bytes
                3 dirs  62,352,756,736 bytes libres

C:\Users\Edmundo J Sanchez M\Desktop\SO>Ren Prueba.py Renombre.py

C:\Users\Edmundo J Sanchez M\Desktop\SO>dir
El volumen de la unidad C es Windows-SSD
El número de serie del volumen es: 8A89-DEDC

Directorio de C:\Users\Edmundo J Sanchez M\Desktop\SO

29/09/2020  08:40 p. m.    <DIR>        .
29/09/2020  08:40 p. m.    <DIR>        ..
29/09/2020  06:07 p. m.    <DIR>        P1
01/09/2020  01:34 a. m.                559 Renombre.py
                1 archivos                559 bytes
                3 dirs  62,352,429,056 bytes libres

C:\Users\Edmundo J Sanchez M\Desktop\SO>

```

## ▪ Comando Chdir

```

C:\Users\Edmundo J Sanchez M\Desktop\SO>Chdir P1

C:\Users\Edmundo J Sanchez M\Desktop\SO\P1>

```

## ▪ Comando echo

```

C:\Users\Edmundo J Sanchez M\Desktop\SO\P1>echo "Hola Mundo"
"Hola Mundo"

C:\Users\Edmundo J Sanchez M\Desktop\SO\P1>

```

## ▪ Comando Find



```
C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>Find "Hola" HolaMundo.c
----- HOLAMUNDO.C
    printf("Hola mundo\n");
C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>
```

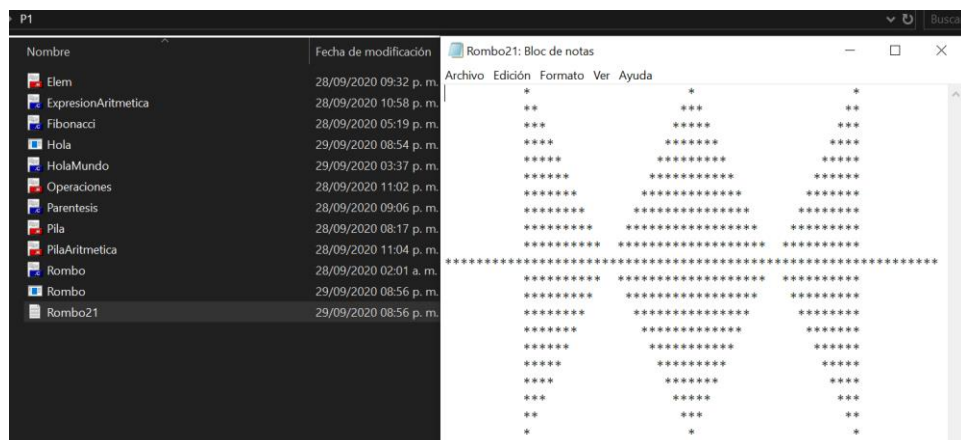
### 3.3. Programas

Nota: el Código es exactamente el mismo a los utilizados en Linux

### 3.3.1. Hola mundo

```
C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>gcc HolaMundo.c -o HolaMundo.exe
C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>HolaMundo.exe
Hola mundo
C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>
```

### 3.3.2. Rombos

[illegible]

### 3.3.3. Serie de Fibonacci

```
C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>gcc Fibonacci.c -o Fibonacci
C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>Fibonacci.exe
Introduce el numero para la serie de Fibonacci 7
0 1 1 2 3 5 8 13
C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>
```

### 3.3.4. Comprobación de emparejamiento de paréntesis

```
C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>gcc Parentesis.c -o Parentesis
C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>Parentesis.exe
Ingresa una expresion: {[()>()()}
Expresion correcta.

C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>Parentesis.exe
Ingresa una expresion: [(){}[]{}]]
Expresion correcta.

C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>Parentesis.exe
Ingresa una expresion: (())[ ]
Se abre un ( pero cierra un ].
Expresion incorrecta.

C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>Parentesis.exe
Ingresa una expresion: {(})[[[
Falto cerrar los simbolos: [( {
Expresion incorrecta.

C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>
```

### 3.3.5. Evaluación de expresiones aritméticas

```
C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>gcc ExpresionAritmetica.c -o ExpresionAritmetica
C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>ExpresionAritmetica.exe
Introduce la expresion aritmetica: (((5-2)+4)*10)
70.00000

C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>ExpresionAritmetica.exe
Introduce la expresion aritmetica: 1+2*5-7/(3+4)
10.00000

C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>ExpresionAritmetica.exe
Introduce la expresion aritmetica: 5^(1+5-2*2)+3*(10*(5*3-10)-4/6)
173.00000

C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>ExpresionAritmetica.exe
Introduce la expresion aritmetica: 5.4*(3-(5-8*2))+4^(1/(1+1))
77.60000

C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>ExpresionAritmetica.exe
Introduce la expresion aritmetica: 5.5*9.6-(4.12*(3.1416-2.36^3))+9.04
103.05094

C:\Users\Edmundo J Sanchez M\Desktop\S0\P1>
```

## 4. Análisis de la practica

Con esta práctica nos familiarizamos más con los comandos más importantes de Linux y Windows, y nos dimos cuenta de que algunas operaciones básicas, como el manejo de archivos y directorios, se vuelven muy sencillas de realizar y de automatizar, mejor que usando una interfaz gráfica. Con el uso de comandos se

puede verificar que se puede lograr hacer lo mismo que usando herramientas graficas. A la misma vez nos percatamos que cada vez que se hace alguna tarea con herramientas graficas por lo general hay algún comando equivalente que también resuelve la misma tarea.

Finalmente observamos que el lenguaje C nos permite desarrollar aplicaciones que funcionen en ambos sistemas operativos, en este caso todos los programas corrieron en Linux y Windows sin tener que cambiarle nada al código.

## **5. Observaciones**

### **Comandos:**

- Linux es sensible a mayúsculas, mientras que Windows no. Una consecuencia de esto es que en Linux podemos tener dos archivos distintos llamados archivo.txt y Archivo.txt, pero en Windows serían el mismo archivo.
- Windows y Linux cuentan con comandos similares en función para la organización y edición de los archivos y directorios, entre otros comandos para distintos procesos.
- Por parte de ambos sistemas operativos no permiten visualizar más haya de texto plano.
- Las consolas de los temas operativos no son para usuarios primerizos, en ambos sistemas operativos cuentan con una gran variedad de comandos y se debe tener una instrucción previa de estos.
- La ejecución de los comandos tiene un mejor control de los procesos y contenido del equipo.

### **Desarrollo de aplicaciones en C:**

- En Linux es muy sencillo instalar el compilador gcc, pues solo basta con ejecutar en la terminal el comando `sudo apt-get install build-essential` y ya lo tendremos disponible en la terminal (en ocasiones ya viene preinstalado). En Windows es muy diferes, pues siempre requerimos instalar MinGW, ya sea de forma directa o mediante algún IDE, y después agregar la ruta de los binarios a la variable de entorno PATH para que podamos compilar desde la consola sin tener que ir siempre a la carpeta de binarios.
- En Windows se tarda un poco más el proceso de compilación que en Linux a pesar de que la PC con Windows contaba con más recursos, esto lo sabemos ya que el entorno Linux ocupado está en una máquina virtual y tiene menos recursos que la PC en donde se ejecuta, ademas de que esto es debido a la prioridad que cada sistema operativo le asignó al proceso gcc.
- Al correr un programa en la consola de Windows que no imprima como ultimo carácter un salto de línea, este se agrega automáticamente; mientras que en Linux esto no sucede.

- Para compilar archivos en C con la biblioteca math.h, en Linux se tiene que agregar la opción -lm para vincularla, mientras que en Windows no es necesario.

## **6. Conclusiones**

Tanto Linux como Windows ofrecen una gran variedad de comandos para trabajar a nivel de consola de una forma relativamente sencilla, sin embargo, en Linux hay más variedad y podemos hacer más cosas que en Windows.

Configurar un compilador en Linux es también más sencillo, porque puede que ya venga preinstalado o que su instalación solo tome un comando, mientras que en Windows hay que descargarlo de la página oficial, instalarlo y configurarlo en las variables de entorno. Pero ya teniendo el compilador, el proceso de creación de aplicaciones es muy similar en ambos sistemas operativos, porque solo necesitamos cualquier editor de texto, la lógica no cambia y la portabilidad se mantiene en el lenguaje C (mientras no hagamos llamadas al sistema).

El entorno y diseño del SO Windows es más amigable y “fácil” de usar a comparación del SO Linux, aunque, como ya se dijo anteriormente, el SO Linux es tiene una variedad más grande de posibilidades para realizar alguna tarea. Y esto posiblemente es lo que lo hace que sea menos usado que Windows, porque es un SO más enfocado a funcionalidad.