



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



SISTEMAS OPERATIVOS

INTRODUCCIÓN AL SISTEMA OPERATIVO LINUX Y WINDOWS

Práctica #1

INTEGRANTES DEL EQUIPO:

COLIN RAMIRO JOEL
HERNÁNDEZ REYES JULIO CÉSAR
MALDONADO CERÓN CARLOS
MENDOZA GARCÍA ELIÚ EDUARDO

Grupo: 4CM1

PROFESOR: CORTÉS GALICIA JORGE

19 de agosto, 2021

I . Introducción teórica

Un sistema operativo es un conjunto de programas que permite manejar el hardware de una computadora y provee servicios a los **programas de aplicación** de software, ejecutándose en modo “privilegiado” respecto de los restantes. Uno de los propósitos del sistema operativo que gestiona el núcleo intermedio consiste en gestionar los recursos de localización y protección de acceso del *hardware*. La mayoría de aparatos electrónicos que utilizan microprocesadores para funcionar, llevan incorporado un sistema operativo ya sean: teléfonos móviles, reproductores de DVD, computadoras, routers, etc. En cuyo caso, son manejados mediante una interfaz gráfica de usuario, un gestor de ventanas o un entorno de escritorio, si es un celular, mediante una consola o control remoto si es un DVD y, mediante una línea de comandos o navegador web si es un enrutador. Dentro de las tareas que realiza el sistema operativo, en particular, se ocupa de gestionar la memoria de nuestro sistema y la carga de los diferentes programas, para ello cada programa tiene una prioridad o jerarquía y en función de la misma contará con los recursos de nuestro sistema por más tiempo que un programa de menor prioridad. El sistema operativo se ocupa también de correr procesos. Podemos llamar proceso a la carga en memoria de cualquier programa, si no está cargado en memoria este programa simplemente no funcionará.

Al realizar un conteo o análisis de todos los dispositivos que requieren de un sistema operativo nos encontramos con diferentes sistemas que realizan estas tareas como lo son Windows, Linux, MAC OS, Android, iOS, etc. En sus diferentes versiones. Dentro de esta práctica únicamente se analizarán los sistemas operativos **Linux** y **Windows**.

Como parte introductoria daremos una breve introducción a ambos sistemas operativos, empezando con Linux. Y es que Linux es la denominación técnica y generalizada que reciben una serie de sistemas operativos de tipo Unix, que también suelen ser de código abierto, multiplataforma, multiusuario y multitarea. Estos sistemas están formados mediante la combinación de varios proyectos, entre los cuales destaca el entorno **GNU**, encabezado por el programador estadounidense Richard Stallman junto a la Free Software Foundation (FSF), una fundación cuyo propósito es difundir el software libre, así como

también el núcleo de sistema operativo conocido como *el mismo S.O Linux*, encabezado por el programador finlandés Linus Torvalds.

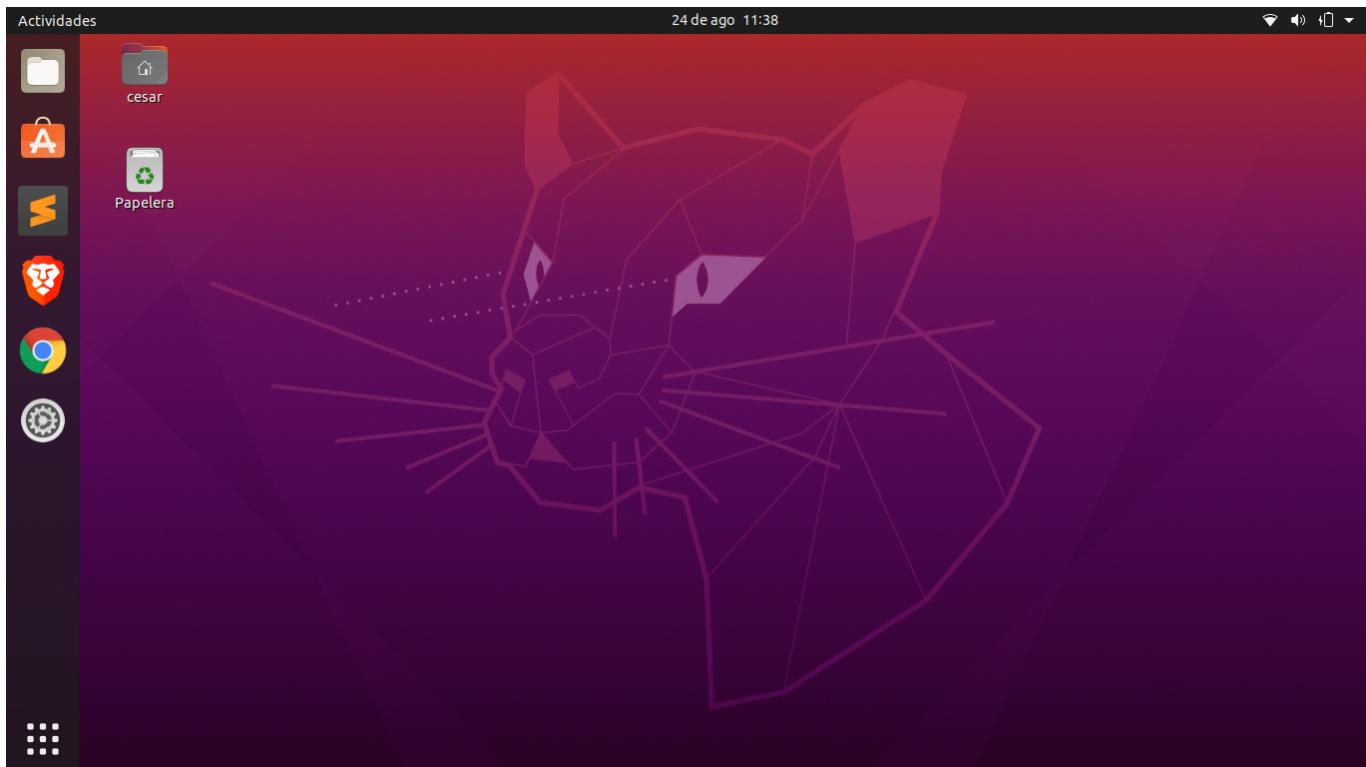
Por otra parte se encuentra el S.O Windows el cual se trata de una familia de distribuciones de software para PC, servidores, sistemas empotrados y antiguamente teléfonos inteligentes desarrollados y vendidos por Microsoft y disponibles para múltiples arquitecturas, tales como x86, x86-64 (x64) y ARM. Si lo llegamos a considerar desde un punto de vista técnico, no son sistemas operativos, sino que contienen uno (**MS-DOS** o el más actual cuyo núcleo es **Windows NT**) junto con una amplia variedad de software; no obstante, es usual denominar al conjunto como sistema operativo en lugar de distribución.

II. Desarrollo Experimental

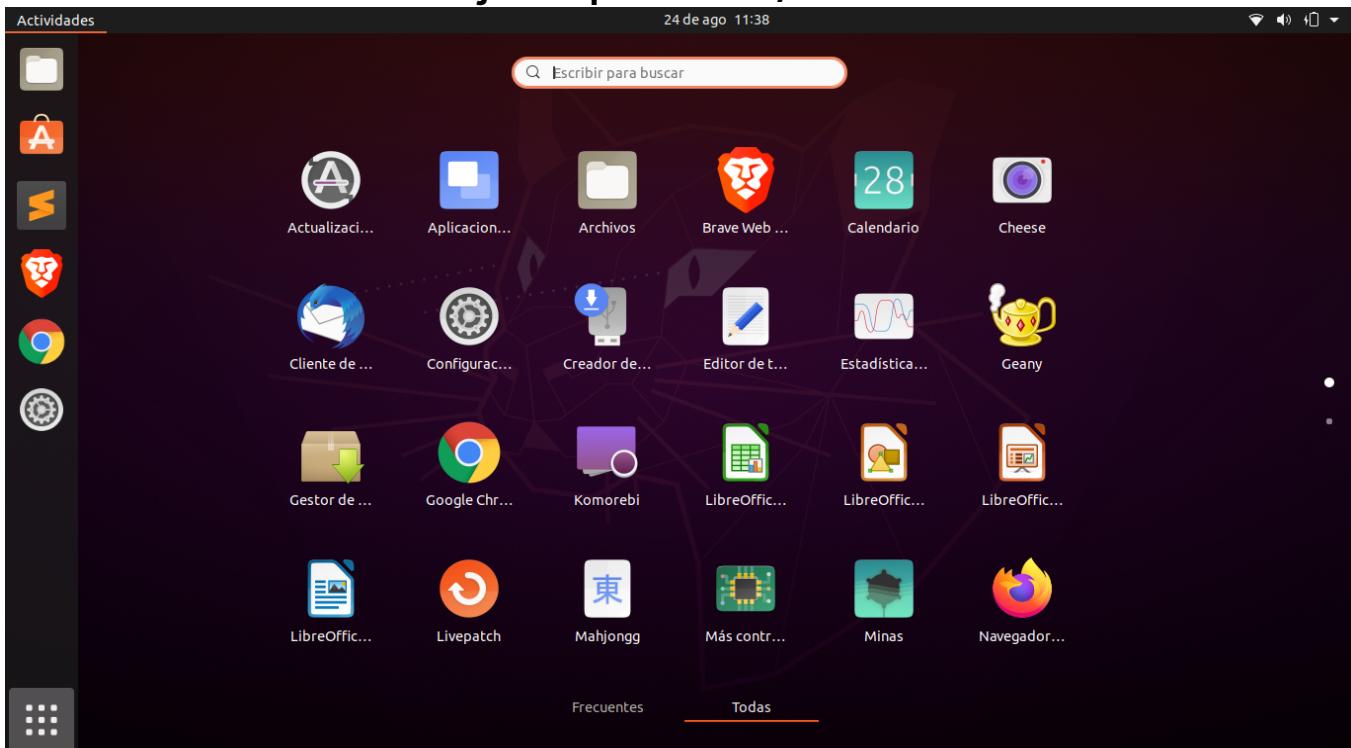
A. Sección Linux

En esta sección de la práctica se trabajará directamente en la máquina virtual “Virtual Box”, se analizará el sistema operativo de Linux particularmente con la distribución **Ubuntu**, sin embargo cabe destacar que existen otras distribuciones del sistema operativo Linux muy útiles, algunas de ellas son: Debian, Linux Mint, MX Linux, Mageia, Solus, Fedora, entre muchas otras. Sin embargo nosotros, decidimos trabajar con Ubuntu en especial debido a que consideramos la más sencilla para comenzar a trabajar con este S.O además de que es la única con la que nos familiarizamos.

Capturas del Sistema



Caja de aplicaciones/menú:



Versión de Linux instalada:

```
cesar@cesar-HP-Notebook:~$ neofetch
  .-+oossssoo+-.
   `:+ssssssssssssssssss+:`-
   -+ssssssssssssssssssssyssss+-+
   .osssssssssssssssssdMMMNyssso.
   /sssssssssssshdmmNNmmyNMNMNhsssss/
   +ssssssssshmydMMMMMMNdddyssssssss+-
   /ssssssssshNMMyhyyyyhmNMNMNhssssssss/
   .sssssssdMMMNhsssssssssshNMNMdssssssss.
   +sssshhhyNMMyssssssssssyNMMyssssssss+
   ossyNMMyMMhsssssssssssssshmmhssssssso
   ossyNMMyMMhsssssssssssssshmmhssssssso
   +sssshhhyNMMyssssssssssssyNMMyssssssss+
   .sssssssdMMMNhsssssssssshNMNMdssssssss.
   /ssssssssshNMMyhyyyyhdNMNMNhssssssss/
   +ssssssssdmydMMMMMMMdddyssssssss+-
   /sssssssssshdmmNNNmyNMNMNhssssss/
   .osssssssssssssssssdMMMNyssso.
   -+ssssssssssssssssyyssss+-+
   `:+ssssssssssssssssss+:`-
   .-+oossssoo+-.

cesar@cesar-HP-Notebook
-----
OS: Ubuntu 20.04.2 LTS x86_64
Host: HP Notebook
Kernel: 5.11.0-27-generic
Uptime: 16 hours, 30 mins
Packages: 1739 (dpkg), 9 (snap)
Shell: bash 5.0.17
Resolution: 1366x768
DE: GNOME
WM: Mutter
WM Theme: Adwaita
Theme: Yaru-dark [GTK2/3]
Icons: Yaru [GTK2/3]
Terminal: gnome-terminal
CPU: Intel Celeron N3060 (2) @ 2.480
GPU: Intel Atom/Celeron/Pentium Proc
Memory: 1181MiB / 3790MiB
```

Comandos en linux:

Comando	Función	Ejemplo de uso
ls	Despliega el contenido de un directorio. Por defecto, se mostrará el contenido de tu directorio de trabajo actual.	ls-R ls-a ls-b ls-all
chmod	Cambia los permisos de lectura, escritura y ejecución de archivos y directorios	chmod 421 Miarchivo.txt
vi	Se utiliza para cambiar un texto en todos los sentidos	vim -t vi -tag
pwd	Imprime la ruta en la que se encuentra el usuario.	pwd
clear	Se limpia la terminal para que solo quede la línea de trabajo.	clear
cd	Navegas por los archivos y directorios	cd Musica cd archivos
cat	Se utiliza para listar el contenido de un archivo en la salida estándar.	cat archivo.txt cat primer_programa.txt cat holamundo.c
grep	Permite buscar a través de todo el texto en un archivo.	grep Hola holamundo.txt grep 2 numeros.txt
rm	Se utiliza para eliminar directorios y todo el contenido dentro de ellos.	rm -r
ps	Describe la información acerca de la selección del proceso actual	ps -e ps -ef
cp	Copia archivos del directorio actual a un directorio diferente.	cp copia1.jpg cp copia2.png
mv	Se utiliza para mover archivos de una carpeta a otra y también para cambiar el nombre del archivo.	mv archivo_movido.jpg archivo_viejo.txt archivo_nuevo

mkdir	Es utilizado para crear un nuevo directorio.	mkdir NuevaCarpeta mkdir Capturas
rmdir	Se utiliza para eliminar directorios pero únicamente vacíos.	rmdir CarpetaElim
whoami	Imprime el nombre de usuario asociado al actual ID de usuario	whoami

Posteriormente se realizaron algunas consultas mediante la consola para verificar su funcionamiento correctamente y de una forma más experimental. Se adjuntan capturas de su realización

Capturas de ejecución de comandos en Terminal de Linux:

```
cesar@cesar-HP-Notebook:~$ ls
Descargas Escritorio Música Público Vídeos
Documentos Imágenes Plantillas snap
cesar@cesar-HP-Notebook:~$ ls -l
total 36
drwxr-xr-x 2 cesar cesar 4096 abr 11 17:23 Descargas
drwxr-xr-x 4 cesar cesar 4096 ago 24 11:31 Documentos
drwxr-xr-x 2 cesar cesar 4096 ago 24 11:31 Escritorio
drwxr-xr-x 2 cesar cesar 4096 ago 24 11:38 Imágenes
drwxr-xr-x 2 cesar cesar 4096 abr 3 00:44 Música
drwxr-xr-x 2 cesar cesar 4096 abr 3 00:44 Plantillas
drwxr-xr-x 2 cesar cesar 4096 abr 3 00:44 Público
drwxr-xr-x 5 cesar cesar 4096 abr 4 11:29 snap
drwxr-xr-x 2 cesar cesar 4096 abr 3 00:44 Vídeos
```

```
cesar@cesar-HP-Notebook:~$ ls -la
total 96
drwxr-xr-x 18 cesar cesar 4096 ago 24 11:35 .
drwxr-xr-x 3 root root 4096 abr 3 06:27 ..
-rw----- 1 cesar cesar 6292 ago 24 11:45 .bash_history
-rw-r--r-- 1 cesar cesar 220 abr 3 06:27 .bash_logout
-rw-r--r-- 1 cesar cesar 3771 abr 3 06:27 .bashrc
drwx----- 18 cesar cesar 4096 ago 17 22:14 .cache
drwx----- 19 cesar cesar 4096 ago 24 11:29 .config
drwxr-xr-x 2 cesar cesar 4096 abr 11 17:23 Descargas
drwxr-xr-x 4 cesar cesar 4096 ago 24 11:31 Documentos
drwxr-xr-x 2 cesar cesar 4096 ago 24 11:31 Escritorio
drwx----- 3 cesar cesar 4096 ago 24 11:16 .gnupg
drwxr-xr-x 2 cesar cesar 4096 ago 24 11:38 Imágenes
-rw-rw-r-- 1 cesar cesar 116 ago 24 11:35 .Komorebi.prop
drwxr-xr-x 3 cesar cesar 4096 abr 3 00:44 .local
drwx----- 5 cesar cesar 4096 abr 3 00:45 .mozilla
drwxr-xr-x 2 cesar cesar 4096 abr 3 00:44 Música
drwx----- 3 cesar cesar 4096 abr 4 13:42 .pki
drwxr-xr-x 2 cesar cesar 4096 abr 3 00:44 Plantillas
-rw-r--r-- 1 cesar cesar 807 abr 3 06:27 .profile
drwxr-xr-x 2 cesar cesar 4096 abr 3 00:44 Público
drwxr-xr-x 5 cesar cesar 4096 abr 4 11:29 snap
drwx----- 2 cesar cesar 4096 abr 3 06:38 ssh
-rw-r--r-- 1 cesar cesar 0 abr 3 19:07 .sudo_as_admin_successful
drwxr-xr-x 2 cesar cesar 4096 abr 3 00:44 Vídeos
```

```
cesar@cesar-HP-Notebook:~$ pwd
/home/cesar
cesar@cesar-HP-Notebook:~$
```

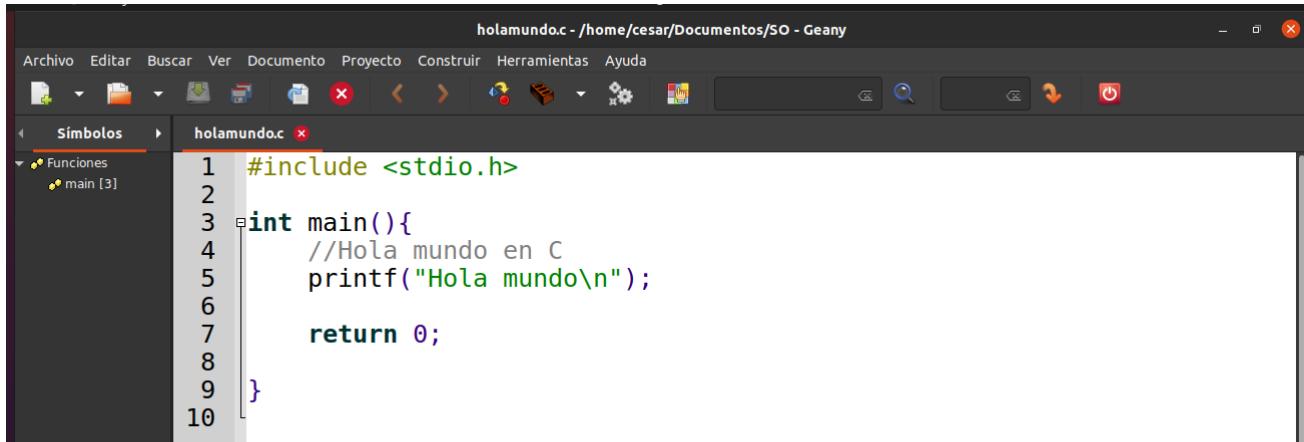
```
cesar@cesar-HP-Notebook:~$ cd nombre_directorio
bash: cd: nombre_directorio: No existe el archivo o el directorio
cesar@cesar-HP-Notebook:~$ cat nombre_archivo
cat: nombre_archivo: No existe el archivo o el directorio
```

```

cesar@cesar-HP-Notebook:~$ la -la | more
total 96
drwxr-xr-x 18 cesar cesar 4096 ago 24 11:35 .
drwxr-xr-x 3 root root 4096 abr 3 06:27 ..
-rw----- 1 cesar cesar 6292 ago 24 11:45 .bash_history
-rw-r--r-- 1 cesar cesar 220 abr 3 06:27 .bash_logout
-rw-r--r-- 1 cesar cesar 3771 abr 3 06:27 .bashrc
drwx----- 18 cesar cesar 4096 ago 17 22:14 .cache
drwx----- 19 cesar cesar 4096 ago 24 11:29 .config
drwxr-xr-x 2 cesar cesar 4096 abr 11 17:23 Descargas
drwxr-xr-x 4 cesar cesar 4096 ago 24 11:31 Documentos
drwxr-xr-x 2 cesar cesar 4096 ago 24 11:31 Escritorio
drwx----- 3 cesar cesar 4096 ago 24 11:16 .gnupg
drwxr-xr-x 2 cesar cesar 4096 ago 24 11:49 Imágenes
-rw-rw-r-- 1 cesar cesar 116 ago 24 11:35 .Komorebi.prop
drwxr-xr-x 3 cesar cesar 4096 abr 3 00:44 .local
drwx----- 5 cesar cesar 4096 abr 3 00:45 .mozilla
drwxr-xr-x 2 cesar cesar 4096 abr 3 00:44 Música
drwx----- 3 cesar cesar 4096 abr 4 13:42 .pki
drwxr-xr-x 2 cesar cesar 4096 abr 3 00:44 Plantillas
-rw-r--r-- 1 cesar cesar 807 abr 3 06:27 .profile
drwxr-xr-x 2 cesar cesar 4096 abr 3 00:44 Público
drwxr-xr-x 5 cesar cesar 4096 abr 4 11:29 snap
drwx----- 2 cesar cesar 4096 abr 3 06:38 .ssh
-rw-r--r-- 1 cesar cesar 0 abr 3 19:07 .sudo_as_admin_successful
drwxr-xr-x 2 cesar cesar 4096 abr 3 00:44 Videos
cesar@cesar-HP-Notebook:~$ rm nombre_archivo
rm: no se puede borrar 'nombre_archivo': No existe el archivo o el directorio
cesar@cesar-HP-Notebook:~$ Ps -
Ps: orden no encontrada
cesar@cesar-HP-Notebook:~$ cp [ruta1]archivo1 [ruta2][archivo 2]
cp: el objetivo '2]' no es un directorio
cesar@cesar-HP-Notebook:~$ cp [ruta1]archivo1 [ruta2][archivo2]
cp: no se puede efectuar `stat' sobre '[ruta1]archivo1': No existe el archivo o el directorio
cesar@cesar-HP-Notebook:~$ mkdir nombre_directorio
cesar@cesar-HP-Notebook:~$ rmdir nombre_directorio
cesar@cesar-HP-Notebook:~$ whoami
cesar
cesar@cesar-HP-Notebook:~$ 

```

Hola mundo en C ejecutado en Consola:

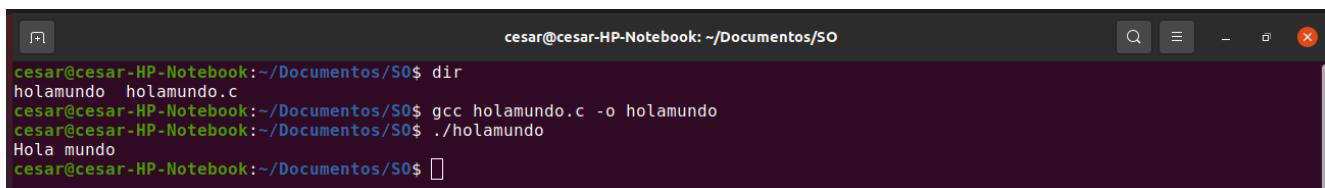


The screenshot shows the Geany IDE interface with the file 'holamundo.c' open. The code contains a simple C program with a main function that prints 'Hola mundo' to the console.

```

#include <stdio.h>
int main(){
    //Hola mundo en C
    printf("Hola mundo\n");
    return 0;
}

```



The terminal window shows the user navigating to the directory containing 'holamundo.c', compiling it with gcc, and then running the resulting executable. The output 'Hola mundo' is displayed.

```

cesar@cesar-HP-Notebook:~/Documentos/SO$ dir
holamundo holamundo.c
cesar@cesar-HP-Notebook:~/Documentos/SO$ gcc holamundo.c -o holamundo
cesar@cesar-HP-Notebook:~/Documentos/SO$ ./holamundo
Hola mundo
cesar@cesar-HP-Notebook:~/Documentos/SO$ 

```

Punto 8

Código

```
#include<stdio.h>
#include<math.h>
int main()
{
    int i,j,k,x,y,a,b,c;

    //PRIMER FIGURA
    // parte superior
    for(a=0;a<3;a++)
    {
        for(b=0;b>3-a;b++)
        {
            printf(" ");
        }
        for(b=0;b<a+1;b++)
        {
            printf("*");
        }
        printf("\n");
    }
    //línea de enmedio
    for(c=0;c<7;c++)
    {
        printf("*");
    }
    printf("\n");

    //Parte inferior
    for(a=2;a>0;a--)
    {
        for(b=0;b>3-a;b++)
        {
            printf(" ");
        }
        for(b=0;b<a+1;b++)
        {
            printf("*");
        }
        printf("\n");
    }

    //SEGUNDA FIGURA
    //SEGUNDA FIGURA
    // parte superior
    for(x=0;x<4;x++)
    {
        for(y=0;y<3-x;y++)
        {
            printf(" ");
        }
        for(y=0;y<x*2+1;y++)
        {
            printf("*");
        }
        printf("\n");
    }

    //TERCER FIGURA
    // parte superior
    for(i=2;i+1>0;i--)
    {
        for(j=0;j<3-i;j++)
        {
            printf(" ");
        }
        for(j=0;j<i+1;j++)
        {
            printf("*");
        }
        printf("\n");
    }

    printf("\n");
}

//Parte inferior
for(x=3;x>0;x--)
{
    for(y=0;y<4-x;y++)
    {
        printf(" ");
    }
    for(y=0;y<2*x-1;y++)
    {
        printf("*");
    }
    printf("\n");
}

//Parte de enmedio
for(k=0;k<7;k++)
{
    printf("*");
}
printf("\n");
//Parte inferior
for(i=2;i+1>0;i--)
{
    for(j=0;j<3-i;j++)
    {
        printf(" ");
    }
    for(j=0;j<i+1;j++)
    {
        printf("*");
    }
    printf("\n");
}
```

Ejecución

Abrimos la terminal de Ubuntu y ponemos “ls” para conocer los directorios y ver en cual esta nuestro archivo, para después poner “cd” espacio y el nombre de nuestro directorio, si lo llegáramos tener en una carpeta hacemos el mismo procedimiento, después para compilar nuestro programa escribimos lo siguiente:

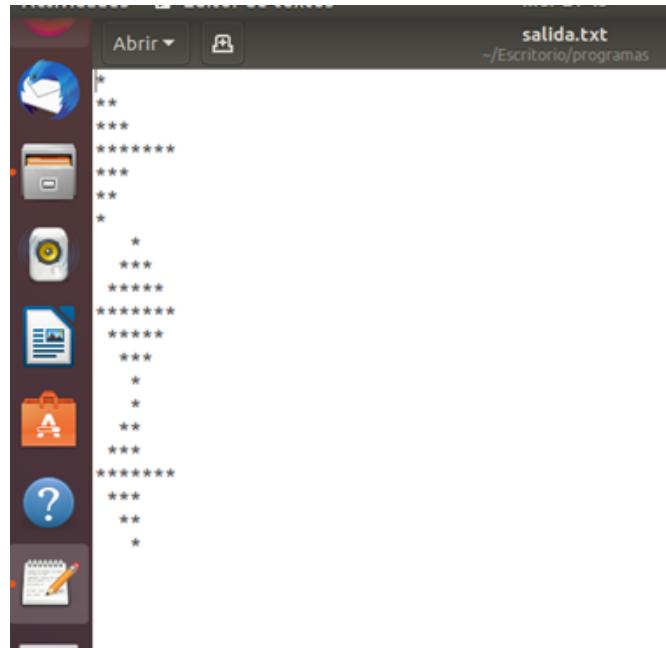
“gcc (nombre del programa).c -o (el nombre que le daremos a nuestro ejecutable)” Y le damos “ENTER”, si no hay errores entonces escribimos

“./(nombre del ejecutable)” Damos “ENTER” y se empieza a ejecutar nuestro programa

```
Descargas Escritorio Imágenes Plantillas Videos
Documentos examples.desktop Música Público
carlos@carlos-VirtualBox:~$ cd Escritorio
carlos@carlos-VirtualBox:~/Escritorio$ ls
figura.c programas
carlos@carlos-VirtualBox:~/Escritorio$ cd programas
carlos@carlos-VirtualBox:~/Escritorio/programas$ gcc P8.c -o P8
carlos@carlos-VirtualBox:~/Escritorio/programas$ ./P8
*
**
***
*****
***
*
*
*****
*****
*
*
*****
*****
*
```

Programa guardado en un archivo

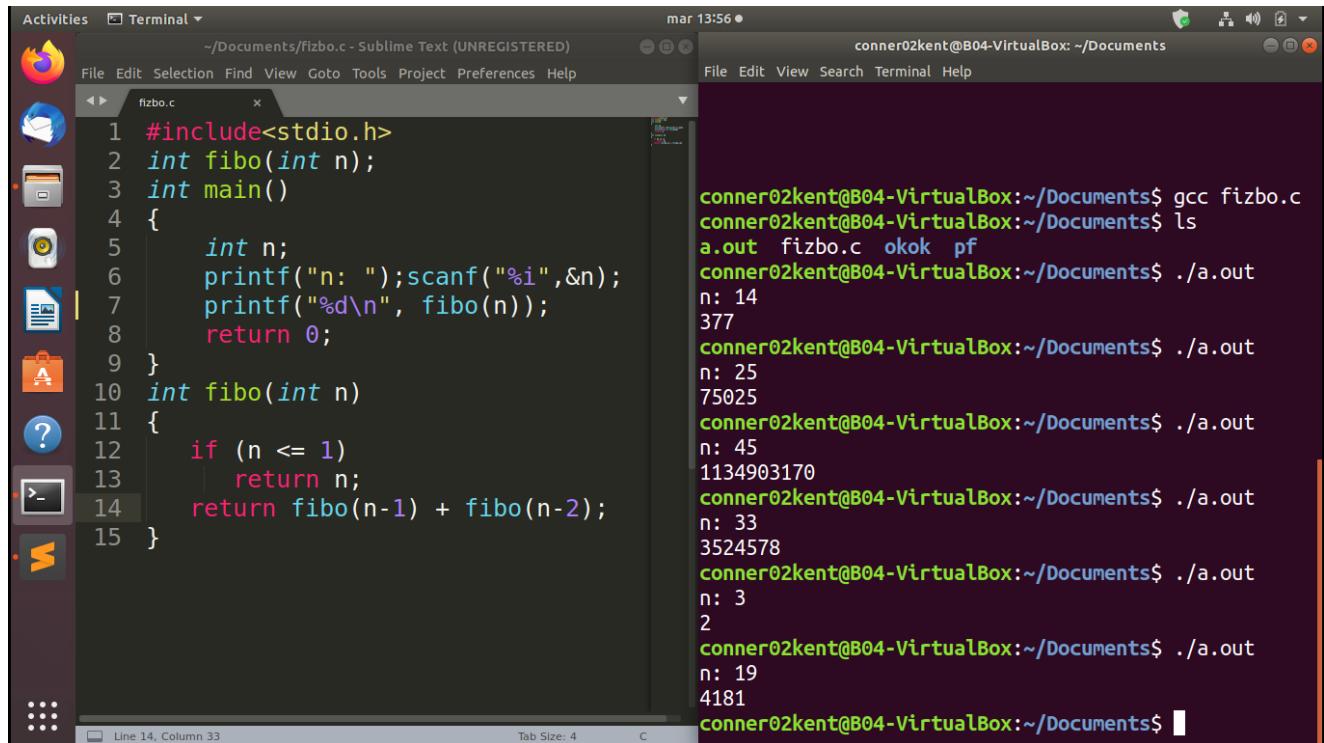
```
carlos@carlos-VirtualBox:~/Escritorio/programas$ ./P8>salida.txt  
carlos@carlos-VirtualBox:~/Escritorio/programas$
```



En esta sección se realizó un programa que calcula la serie de Fibonacci de un número dado como entrada.

CÓDIGO

COMPILEACIÓN Y PRUEBAS



The screenshot shows a Linux desktop environment with a dark theme. On the left, there's a dock with icons for a browser, file manager, terminal, and other applications. In the center, there's a Sublime Text window titled "fizbo.c" containing C code for calculating Fibonacci numbers. To the right of the Sublime Text window is a terminal window with a dark background and light-colored text. The terminal shows the following session:

```
conner02kent@B04-VirtualBox:~/Documents$ gcc fizbo.c
conner02kent@B04-VirtualBox:~/Documents$ ls
a.out fizbo.c okok pf
conner02kent@B04-VirtualBox:~/Documents$ ./a.out
n: 14
377
conner02kent@B04-VirtualBox:~/Documents$ ./a.out
n: 25
75025
conner02kent@B04-VirtualBox:~/Documents$ ./a.out
n: 45
1134903170
conner02kent@B04-VirtualBox:~/Documents$ ./a.out
n: 33
3524578
conner02kent@B04-VirtualBox:~/Documents$ ./a.out
n: 3
2
conner02kent@B04-VirtualBox:~/Documents$ ./a.out
n: 19
4181
conner02kent@B04-VirtualBox:~/Documents$
```

En esta sección se realizó un programa que determine si un grupo de paréntesis están balanceados utilizando pilas. Se consideraron tres tipos de paréntesis: (), { }, [].

CÓDIGO

```
parentesisbalanceados.c x
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 struct nodo {
6     char simbolo;
7     struct nodo *sig;
8 };
9 struct nodo *raiz=NULL;
10 void insertar(char x){
11     struct nodo *nuevo;
12     nuevo = malloc(sizeof(struct nodo));
13     nuevo->simbolo = x;
14     if (raiz == NULL){
15         raiz = nuevo;
16         nuevo->sig = NULL;
17     }else{
18         nuevo->sig = raiz;
19         raiz = nuevo;
20     }
21 }
22 char extraer(){
23     if(raiz != NULL){
24         char informacion = raiz->simbolo;
25         struct nodo *temp = raiz;
26         raiz = raiz->sig;
27         free(temp);
28         return informacion;
29     }else{return -1;}
30 }
31 void liberar(){
32     struct nodo *tempo = raiz;
33     struct nodo *temp;
34     while(tempo != NULL){
35         temp = tempo;
36         tempo = tempo->sig;
37         free(temp);
38     }
39 }
40 int vacia(){
41     if(raiz == NULL){
42         return 1;
43     }else{return 0;}
44 }
45 void cargarFormula(char *formula){
46     printf("Ingresar la formula:");
47     fgets(formula,100,stdin);
48 }
```

```
49 int verificarBalance(char *formula){
50     int f;
51     for (f=0;f < strlen(formula);f++){
52         if(formula[f] == '(' || formula[f] == '[' || formula[f] == '{'){
53             insertar(formula[f]);
54         }else{
55             if(formula[f] == ')'){
56                 if(extraer() != '('){return 0;}
57             }else{
58                 if(formula[f] == ']'){
59                     if(extraer() != '['){return 0;}
60                 }else{
61                     if(formula[f] == '}'){
62                         if(extraer() != '{'){return 0;}
63                     }
64                 }
65             }
66         }
67     }
68     if (vacia()){return 1;}
69     else{return 0;}
70 }
71 int main(){
72     char formula[100];
73     cargarFormula(formula);
74     if(verificarBalance(formula)){
75         printf("El grupo de paréntesis SI esta balanceado\n");
76     }else{
77         printf("El grupo de paréntesis NO esta balanceado\n");
78     }
79     liberar();
80     return 0;
81 }
```

COMPILACIÓN Y PRUEBAS

```
[ ] cesar@cesar-HP-Notebook:~/Documentos/SO$ gcc parentesisbalanceados.c -o parentesisbalanceados
cesar@cesar-HP-Notebook:~/Documentos/SO$ ./parentesisbalanceados
Ingrese la formula:{}  
El grupo de paréntesis SI esta balanceado  
cesar@cesar-HP-Notebook:~/Documentos/SO$ ./parentesisbalanceados
Ingrese la formula:()  
El grupo de paréntesis SI esta balanceado  
cesar@cesar-HP-Notebook:~/Documentos/SO$ ./parentesisbalanceados
Ingrese la formula:[]  
El grupo de paréntesis SI esta balanceado  
cesar@cesar-HP-Notebook:~/Documentos/SO$ ./parentesisbalanceados
Ingrese la formula:{  
El grupo de paréntesis NO esta balanceado  
cesar@cesar-HP-Notebook:~/Documentos/SO$ ./parentesisbalanceados
Ingrese la formula:]  
El grupo de paréntesis NO esta balanceado  
cesar@cesar-HP-Notebook:~/Documentos/SO$ ./parentesisbalanceados
Ingrese la formula:)  
El grupo de paréntesis NO esta balanceado  
cesar@cesar-HP-Notebook:~/Documentos/SO$ ./parentesisbalanceados
Ingrese la formula:(((([[{{}}}}]))))  
El grupo de paréntesis SI esta balanceado  
cesar@cesar-HP-Notebook:~/Documentos/SO$ ./parentesisbalanceados
Ingrese la formula:(((([]))))  
El grupo de paréntesis SI esta balanceado  
cesar@cesar-HP-Notebook:~/Documentos/SO$ ./parentesisbalanceados
Ingrese la formula:(((([])))  
El grupo de paréntesis NO esta balanceado  
cesar@cesar-HP-Notebook:~/Documentos/SO$ ./parentesisbalanceados
Ingrese la formula:)){{{}{  
El grupo de paréntesis NO esta balanceado
```

En esta sección se realizó un programa que evalúa expresiones aritméticas delimitada completamente con paréntesis balanceados. NOTA: este programa se realizó en C++ porque se nos hizo más fácil por las utilidades que este tiene para los stacks, además ya no teníamos mucho tiempo para completar la práctica.

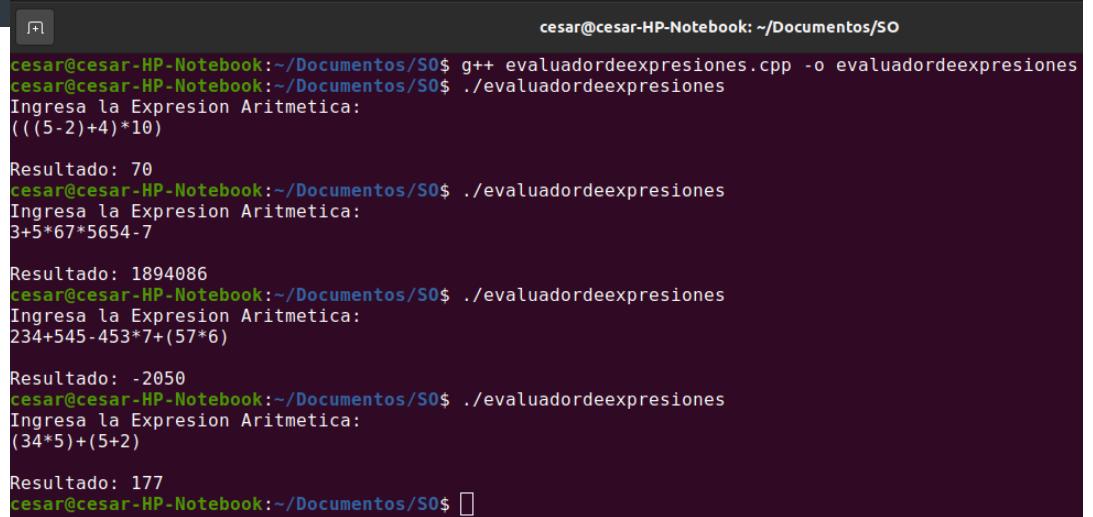
CÓDIGO

```
evaluadordeexpresiones.cpp x
1 // Programa en C++ para
2 // evaluar una expresión dada donde
3 // los símbolos están separados con
4 // un espacio
5 #include <bits/stdc++.h>
6 using namespace std;
7
8 int precedencia(char op){
9     if(op == '+' || op == '-')
10    return 1;
11    if(op == '*' || op == '/')
12    return 2;
13    return 0;
14 }
15
16 int aplicarOper(int a, int b, char op){
17     switch(op){
18         case '+': return a + b;
19         case '-': return a - b;
20         case '*': return a * b;
21         case '/': return a / b;
22     }
23     return 0;
24 }
25
26 int evaluar(string simbolos){
27     int i;
28     stack<int> valores;
29     stack<char> ops;
30
31     for(i = 0; i < simbolos.length(); i++){
32
33         if(simbolos[i] == ' ')
34             continue;
35
36         else if(simbolos[i] == '('){
37             ops.push(simbolos[i]);
38         }
39         else if(isdigit(simbolos[i])){
40             int val = 0;
41
42             while(i < simbolos.length() && isdigit(simbolos[i]))
43             {
44                 val = (val*10) + (simbolos[i] - '0');
45                 i++;
46             }
47             valores.push(val);
48             i--;
49         }
50
51         else if(simbolos[i] == ')'){
52             {
53                 while(!ops.empty() && ops.top() != '(')
54                 {
55                     int val2 = valores.top();
56                     valores.pop();
57
58                     int val1 = valores.top();
59                     valores.pop();
60
61                     if(ops.top() == '+')
62                         valores.push(val1 + val2);
63                     else if(ops.top() == '-')
64                         valores.push(val1 - val2);
65                     else if(ops.top() == '*')
66                         valores.push(val1 * val2);
67                     else if(ops.top() == '/')
68                         valores.push(val1 / val2);
69
70                     ops.pop();
71                 }
72             }
73         }
74     }
75
76     return valores.top();
77 }
```

```

59     valores.pop();
60
61     char op = ops.top();
62     ops.pop();
63
64     valores.push(aplicarOper(val1, val2, op));
65 }
66 if(!ops.empty())
67     ops.pop();
68 }
69 else
70 {
71     while(!ops.empty() && precedencia(ops.top()) >= precedencia(simbolos[i])){
72         int val2 = valores.top();
73         valores.pop();
74
75         int val1 = valores.top();
76         valores.pop();
77
78         char op = ops.top();
79         ops.pop();
80
81         valores.push(aplicarOper(val1, val2, op));
82     }
83     ops.push(simbolos[i]);
84 }
85 }
86 while(!ops.empty()){
87     int val2 = valores.top();
88     valores.pop();
89
90     int val1 = valores.top();
91     valores.pop();
92
93     char op = ops.top();
94     ops.pop();
95
96     valores.push(aplicarOper(val1, val2, op));
97 }
98 return valores.top();
99 }
100
101 int main() {
102     string expresion;
103     cout << ("Ingresa la Expresión Aritmetica: \n");
104     cin >> expresion;
105     cout << "\n";
106     cout << "Resultado: " << evaluar(expresion) << "\n";
107     return 0;
108 }
```

COMPILACIÓN Y PRUEBAS



The terminal window shows the following session:

```

cesar@cesar-HP-Notebook:~/Documentos/SO$ g++ evaluadordeexpresiones.cpp -o evaluadordeexpresiones
cesar@cesar-HP-Notebook:~/Documentos/SO$ ./evaluadordeexpresiones
Ingresa la Expresión Aritmetica:
(((5-2)+4)*10)

Resultado: 70
cesar@cesar-HP-Notebook:~/Documentos/SO$ ./evaluadordeexpresiones
Ingresa la Expresión Aritmetica:
3+5*67*5654-7

Resultado: 1894086
cesar@cesar-HP-Notebook:~/Documentos/SO$ ./evaluadordeexpresiones
Ingresa la Expresión Aritmetica:
234+545-453*7+(57*6)

Resultado: -2050
cesar@cesar-HP-Notebook:~/Documentos/SO$ ./evaluadordeexpresiones
Ingresa la Expresión Aritmetica:
(34*5)+(5+2)

Resultado: 177
cesar@cesar-HP-Notebook:~/Documentos/SO$ 
```

B. Sección Windows

En esta sección se analizará particularmente el sistema operativo Windows en su versión 10.

Consola de Windows



COMANDO	FUNCIÓN
dir	Muestra la lista de subdirectorios y archivos de un directorio
ipconfig	De forma predeterminada, se muestra solamente la dirección IP, la máscara de subred y la puerta de enlace predeterminada para cada adaptador enlazado con TCP/IP
cls	Borra la pantalla
ver	Muestra la versión de Windows
tree	Muestra de forma gráfica la estructura de carpetas de una unidad o ruta
cd <i>nombre_directorio</i>	Cambia de directorio
type <i>nombre_archivo</i>	Muestra el contenido de uno o más archivos de texto

<code>mkdir nombre_directorio</code>	Crea un directorio
<code>rmdir nombre_directorio</code>	Quita (elimina) un directorio
<code>del nombre_archivo</code>	Elimina uno o más archivos
<code>copy nombre_archivo nuevo_nombre_archivo</code>	Copia uno o más archivos en otra ubicación
<code>ren nombre_archivo nuevo_nombre_archivo</code>	Cambia el nombre de uno o más archivos
<code>chdir nombre_directorio</code>	Muestra el nombre del directorio actual o cambia de directorio
<code>echo "Hola mundo"</code>	Muestra mensajes o activa y desactiva el eco del comando
<code>find "cadena_buscar" nombre_archivo</code>	Busca una cadena de texto en uno o más archivos

Pruebas de ejecución de comandos

cls

```
C:\Users\Windows 10\Documents\ESCOMaywey\4>ver  
Microsoft Windows [Versión 10.0.22000.160]
```

```
C:\Users\Windows 10\Documents\ESCOMaywey\4>tree  
Listado de rutas de carpetas  
El número de serie del volumen es 0A40-B8E0  
C:  
    └── DSDigitales  
    └── EAnalógica  
    └── MAvanzadas  
    └── ProbaEstadística  
    └── SOperativos  
        └── 1  
            └── a  
    └── TDAWeb  
    └── TeoríaComputacional
```

```
C:\Users\Windows 10\Documents\ESCOMaywey\4>cd soperativos  
C:\Users\Windows 10\Documents\ESCOMaywey\4\SOperativos>
```

```
C:\Users\Windows 10\Documents\ESCOMaywey\4\SOperativos>type hola.txt  
Archivo de texto de ejemplo  
:)
```

```
C:\Users\Windows 10\Documents\ESCOMaywey\4\SOperativos>mkdir si  
C:\Users\Windows 10\Documents\ESCOMaywey\4\SOperativos>tree  
Listado de rutas de carpetas  
El número de serie del volumen es 0A40-B8E0  
C:  
    └── 1  
        └── a  
    └── si
```

```
C:\Users\Windows 10\Documents\ESCOMaywey\4\SOperativos>rmdir si  
C:\Users\Windows 10\Documents\ESCOMaywey\4\SOperativos>tree  
Listado de rutas de carpetas  
El número de serie del volumen es 0A40-B8E0  
C:  
    └── 1  
        └── a
```

```
C:\Users\Windows 10\Documents\ESCOMaywey\4\SOperativos>copy fizbo.c fizbo1.c  
1 archivo(s) copiado(s).
```

```
Directorio de C:\Users\Windows 10\Documents\ESCOMaywey\4\SOperativos

25/08/2021 12:14 p. m.    <DIR>      .
22/08/2021 08:49 p. m.    <DIR>      ..
25/08/2021 12:11 p. m.    <DIR>      1
24/08/2021 01:25 p. m.        44,980 a.exe
24/08/2021 02:02 p. m.    261,081 Captura de pantalla 2021-08-24 140229.png
25/08/2021 12:06 p. m.    38,999 Captura de pantalla 2021-08-25 120617.png
24/08/2021 01:36 p. m.        221 fizbo.c
25/08/2021 12:13 p. m.        31 hola.txt
22/08/2021 08:45 p. m.    964,543 Unidad1_Actividad1_MendozaGarciaEliuEduan
do.pdf
              6 archivos      1,309,855 bytes
              3 dirs  51,633,881,088 bytes libres
```

```
C:\Users\Windows 10\Documents\ESCOMaywey\4\SOperativos>del hola.txt
```

```
C:\Users\Windows 10\Documents\ESCOMaywey\4\SOperativos>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 0A40-B8E0
```

```
Directorio de C:\Users\Windows 10\Documents\ESCOMaywey\4\SOperativos
```

```
25/08/2021 12:15 p. m.    <DIR>      .
22/08/2021 08:49 p. m.    <DIR>      ..
25/08/2021 12:11 p. m.    <DIR>      1
24/08/2021 01:25 p. m.        44,980 a.exe
24/08/2021 02:02 p. m.    261,081 Captura de pantalla 2021-08-24 140229.png
25/08/2021 12:06 p. m.    38,999 Captura de pantalla 2021-08-25 120617.png
24/08/2021 01:36 p. m.        221 fizbo.c
22/08/2021 08:45 p. m.    964,543 Unidad1_Actividad1_MendozaGarciaEliuEduan
do.pdf
              5 archivos      1,309,824 bytes
              3 dirs  51,633,741,824 bytes libres
```

```
C:\Users\Windows 10\Documents\ESCOMaywey\4\SOperativos>ren fizbo1.c fizbo2.c
```

```
C:\Users\Windows 10\Documents\ESCOMaywey\4\SOperativos>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 0A40-B8E0
```

```
Directorio de C:\Users\Windows 10\Documents\ESCOMaywey\4\SOperativos
```

```
25/08/2021 12:18 p. m.    <DIR>      .
22/08/2021 08:49 p. m.    <DIR>      ..
25/08/2021 12:11 p. m.    <DIR>      1
24/08/2021 01:25 p. m.        44,980 a.exe
24/08/2021 02:02 p. m.    261,081 Captura de pantalla 2021-08-24 140229.png
25/08/2021 12:06 p. m.    38,999 Captura de pantalla 2021-08-25 120617.png
24/08/2021 01:36 p. m.        221 fizbo.c
24/08/2021 01:36 p. m.        221 fizbo2.c
```

```
C:\Users\Windows 10\Documents\ESCOMaywey\4\SOperativos>chdir 1
```

```
C:\Users\Windows 10\Documents\ESCOMaywey\4\SOperativos\1>
```

```
C:\Users\Windows 10\Documents\ESCOMaywey\4\SOperativos\1>echo "hello world"
"hello world"
```

```
C:\Users\Windows 10\Documents\ESCOMaywey\4>find "excelente" hola.txt
```

```
----- HOLA.TXT
excelente
```

COMPILACIÓN Y EJECUCIÓN DE LOS PROGRAMAS EN WINDOWS

Hola Mundo

The screenshot shows a Sublime Text window with three tabs: 'holamundo.c', 'parentesisbalanceados.c', and 'evaluadordeexpresiones.cpp'. The 'holamundo.c' tab is active, displaying the following C code:

```
1 #include <stdio.h>
2
3 int main(){
4     //Hola mundo en C
5     printf("Hola mundo\n");
6
7     return 0;
8
9 }
```

The status bar at the bottom shows 'Símbolo del sistema'. The terminal below the editor displays the command 'gcc holamundo.c -o hola mundo' followed by the output 'Hola mundo'.

Figura punto 8:

Fibonacci

The screenshot shows two windows side-by-side. On the left is Sublime Text with the file 'fizbo.c' open. The code implements a recursive Fibonacci function. On the right is a command-line window (cmd.exe) showing the execution of the program and its output for different input values.

```
#include<stdio.h>
int fibo(int n);
int main()
{
    int n;
    printf("n: ");scanf("%i",&n);
    printf("%d", fibo(n));
    return 0;
}
int fibo(int n)
{
    if (n <= 1)
        return n;
    return fibo(n-1) + fibo(n-2);
}
```

C:\Users\Windows 10\Documents\ESCOMaywey\4\S Operativos>gcc fizbo.c
C:\Users\Windows 10\Documents\ESCOMaywey\4\S Operativos>a
n: 33
3524578
C:\Users\Windows 10\Documents\ESCOMaywey\4\S Operativos>a
n: 25
75025
C:\Users\Windows 10\Documents\ESCOMaywey\4\S Operativos>a
n: 19
4181
C:\Users\Windows 10\Documents\ESCOMaywey\4\S Operativos>a
n: 45
1134903170

Paréntesis balanceados

The screenshot shows a code editor with three tabs: 'holamundo.c', 'parentesisbalanceados.c', and 'evaluadordeexpresiones.cpp'. The 'parentesisbalanceados.c' tab contains a C program that checks if a given string of parentheses is balanced. The 'cmd.exe' window on the right shows the execution of this program and its output for various test cases.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct nodo {
    char simbolo;
    struct nodo *sig;
};
struct nodo *raiz=NULL;
void insertar(char x){
    struct nodo *nuevo;
    nuevo = malloc(sizeof(struct nodo));
    nuevo->simbolo = x;
    if (raiz == NULL){
        raiz = nuevo;
        nuevo->sig = NULL;
    }else{
        nuevo->sig = raiz;
        raiz = nuevo;
    }
}
```

C:\Users\cesar\OneDrive\Documentos\Cuarto Semestre\SISTEMAS OPERATIVOS\Primer Parcial\Codigos>gcc parentesisbalanceados.c -o parentesisbalanceados
C:\Users\cesar\OneDrive\Documentos\Cuarto Semestre\SISTEMAS OPERATIVOS\Primer Parcial\Codigos>parentesisbalanceados
Ingrese la formula:{}
El grupo de paréntesis SI esta balanceado
C:\Users\cesar\OneDrive\Documentos\Cuarto Semestre\SISTEMAS OPERATIVOS\Primer Parcial\Codigos>parentesisbalanceados
Ingrese la formula:[]
El grupo de paréntesis SI esta balanceado
C:\Users\cesar\OneDrive\Documentos\Cuarto Semestre\SISTEMAS OPERATIVOS\Primer Parcial\Codigos>parentesisbalanceados
Ingrese la formula:()
El grupo de paréntesis SI esta balanceado
C:\Users\cesar\OneDrive\Documentos\Cuarto Semestre\SISTEMAS OPERATIVOS\Primer Parcial\Codigos>parentesisbalanceados
Ingrese la formula:[
El grupo de paréntesis NO esta balanceado
C:\Users\cesar\OneDrive\Documentos\Cuarto Semestre\SISTEMAS OPERATIVOS\Primer Parcial\Codigos>parentesisbalanceados
Ingrese la formula:}
El grupo de paréntesis NO esta balanceado
C:\Users\cesar\OneDrive\Documentos\Cuarto Semestre\SISTEMAS OPERATIVOS\Primer Parcial\Codigos>parentesisbalanceados
Ingrese la formula:{}}
El grupo de paréntesis NO esta balanceado
C:\Users\cesar\OneDrive\Documentos\Cuarto Semestre\SISTEMAS OPERATIVOS\Primer Parcial\Codigos>parentesisbalanceados
Ingrese la formula:((())
El grupo de paréntesis NO esta balanceado
C:\Users\cesar\OneDrive\Documentos\Cuarto Semestre\SISTEMAS OPERATIVOS\Primer Parcial\Codigos>parentesisbalanceados
Ingrese la formula:[]
El grupo de paréntesis NO esta balanceado
C:\Users\cesar\OneDrive\Documentos\Cuarto Semestre\SISTEMAS OPERATIVOS\Primer Parcial\Codigos>

Evaluador de Expresiones Aritméticas

The screenshot shows a terminal window with five separate command-line sessions, each displaying an arithmetic expression, its result, and a prompt for the user to press a key to continue.

```
C:\Users\cesar\OneDrive\Documentos\Cuarto Semestre\SISTEMAS OPERATIVOS\Primer Parcial\Codigos\evaluadordeexpresiones.cpp
```

```
File Edit Selection Find View Goto Tools Project Preferences Help
```

```
holamundo.c parentesisbalanceados.c evaluadordeexpresiones.cpp
```

```
1 // Programa en C++ para
2 // evaluar una expresion dada donde
3 // los simbolos estan separados con
4 // un espacio
5 #include <bits/stdc++.h>
6 using namespace std;
7
8 int precedencia(char op){
9     if(op == '+' || op == '-')
10    return 1;
11    if(op == '*' || op == '/')
12    return 2;
13    return 0;
14 }
15
16 int aplicarOper(int a, int b, char op){
17     switch(op){
18         case '+': return a + b;
19         case '-': return a - b;
20         case '*': return a * b;
21         case '/': return a / b;
22     }
23     return 0;
24 }
25
26 int evaluar(string simbolos){
27     int i;
28     stack<int> valores;
29     stack<char> ops;
30
31     for(i = 0; i < simbolos.length(); i++){
32         if(simbolos[i] == ' ')
33     }
```

Line 5, Column 25

```
Ingresla Expresion Aritmetica:  
((5-2)+4)*10  
Resultado: 70  
-----  
Process exited after 17.85 seconds with return value 0  
Presione una tecla para continuar . . .
```

```
Ingresla Expresion Aritmetica:  
3+5*67*5654-7  
Resultado: 1894086  
-----  
Process exited after 30.83 seconds with return value 0  
Presione una tecla para continuar . . .
```

```
Ingresla Expresion Aritmetica:  
(3*4(7+2/8))  
Resultado: 28  
-----  
Process exited after 1.746 seconds with return value 0  
Presione una tecla para continuar . . .
```

```
Ingresla Expresion Aritmetica:  
(9*4+(4-8)/4*2)  
Resultado: 34  
-----  
Process exited after 1.183 seconds with return value 0  
Presione una tecla para continuar . . .
```

```
Ingresla Expresion Aritmetica:  
(2/(4*6)+3(6-5))  
Resultado: 4  
-----  
Process exited after 1.492 seconds with return value 0  
Presione una tecla para continuar . . .
```

III.Diferencias

Ambos sistemas operativos, son muy útiles en la vida cotidiana, más que nada en el uso personal, sin embargo si se encuentran algunas diferencias que podemos destacar en cuanto a funcionalidades, sintaxis, entorno, etc.

El sistema de Windows es más intuitivo para el usuario, mientras que en linux se requiere pasar por una cierta curva de aprendizaje para poder manejar el sistema de forma adecuada.

En Linux es más cómodo trabajar por consola en comparación que en windows pues en este último, la mayoría de usuarios ni siquiera sabe que existe.

Existen más aplicaciones y más soporte en el sistema Windows que en el de Linux, puesto que la mayoría de usuarios usa de forma más común Windows y entonces las empresas se centran en mayor parte en mejorar la experiencia de estos usuarios.

IV.Similitudes

En ambos sistemas al menos en la distro de ubuntu tienen un entorno gráfico el cual es más amigable con el usuario a la hora de empezar a usar los SO.

Se tiene una terminal en ambos sistemas.

Dejando de lado que si quieras utilizar Linux tienes que instalarlo tú mismo, la forma de instalar los SO tanto windows como una distro de Linux es prácticamente la misma, solo tenemos que seleccionar siguiente-siguiente.

V. Conclusiones

A lo largo de la realización de esta práctica, nos percatamos lo sencillo que es de utilizar la terminal de comandos en ambos Sistemas Operativos, sin embargo si notamos que el SO de Windows, es un poco más amigable debido a su diseño gráfico y más que nada a que es más utilizado que Ubuntu.

Algo que podemos destacar es que ambos tienen funciones muy similares el uno del otro, obviamente con una sintaxis algo diferente, pero fácil de aprender para nuevos usuarios. Algo que nos pareció agradable fue que por el lado de Ubuntu, tiene adjunto con su terminal un manual que se puede ejecutar únicamente con la palabra “man” seguida con el nombre de algún comando para saber su función, esto por obvias razones, se encuentra en el idioma inglés, pero los integrantes del equipo tenemos buen nivel y fue concretada la práctica con practicidad y en tiempo.

VI. Bibliografía

1. <https://www.linux.org/>
2. <https://desarrollarinclusion.cilsa.org/>
3. <https://www.microsoft.com/es-mx/windows/>
4. <https://www.ionos.mx/digitalguide/servidores/know-how/>