

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 Josué

**Profesor:** Cortés Galicia Jorge

**Fecha de entrega:**



**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 funcion para la cual se desarrolló ademas de que en Linux todo es gratuito por el contrario en Windows que la mayoría de las aplicaciones son de paga. Componentes con funcion similar:

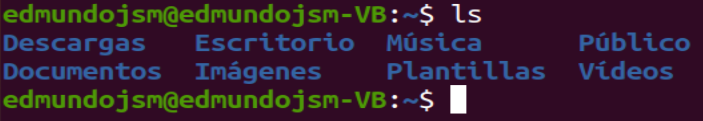
* LibreOffice Writer: Microsoft Office.
* LibreOffice Impress: Microsoft PowerPoint.
* LibreOffice Calc: Microsoft Excel.
* Editor de texto: Bloc de notas.
* Terminal: Administrador: símbolo del sistema.
* Papelera: Papelera.
* Archivos: Explorador de archivos.

**2.4. Función de los comandos**

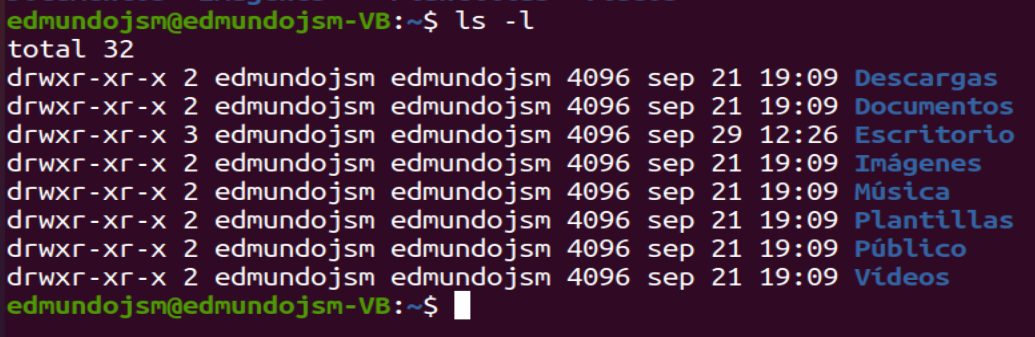
* **ls**: Muestra los directorios y archivos de la carpeta personal.
* **ls -l**: Muestra los directorios y archivos de la carpeta actual con sus propiedades y atributos.
* **ls -a**: Muestra los directorios y archivos de la carpeta actual, incluidos los ocultos y los del sistema.
* **ls -F**: Muestra 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 ultima 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**

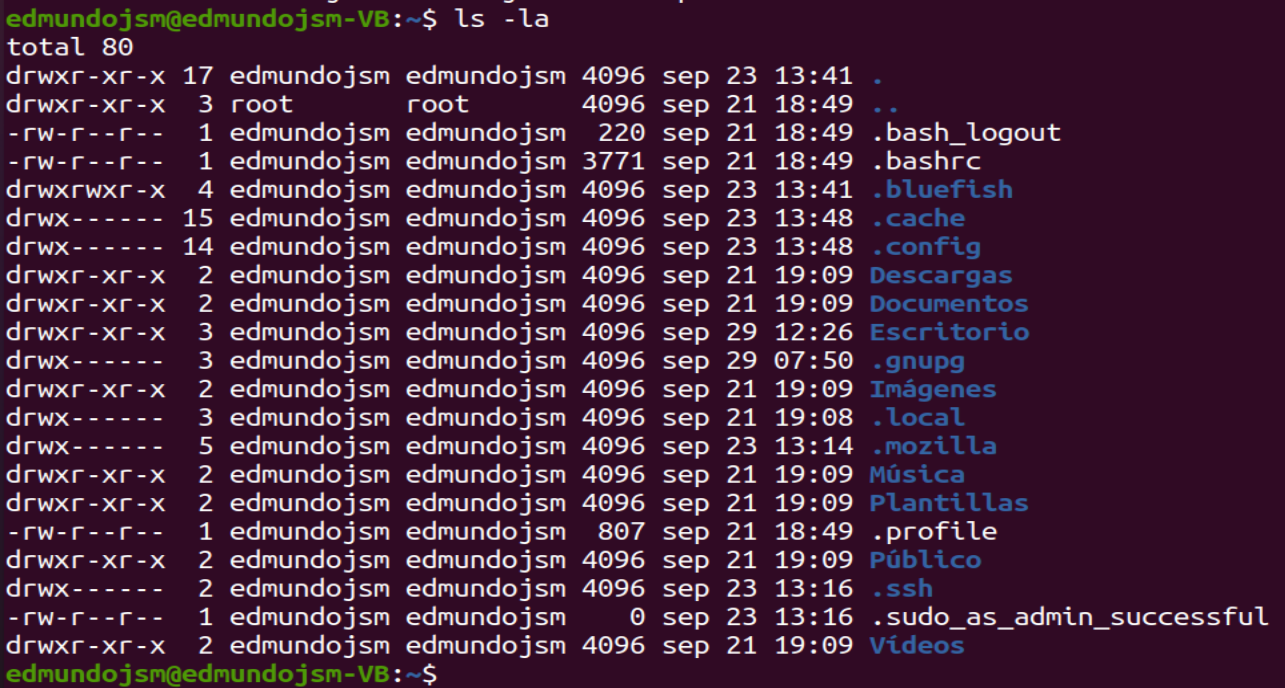
* Comando ls



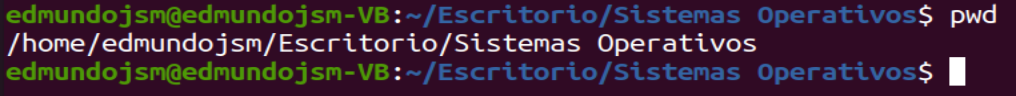
* Comando ls -l



* Comando ls -la



* Comando pwd



* Comando clear

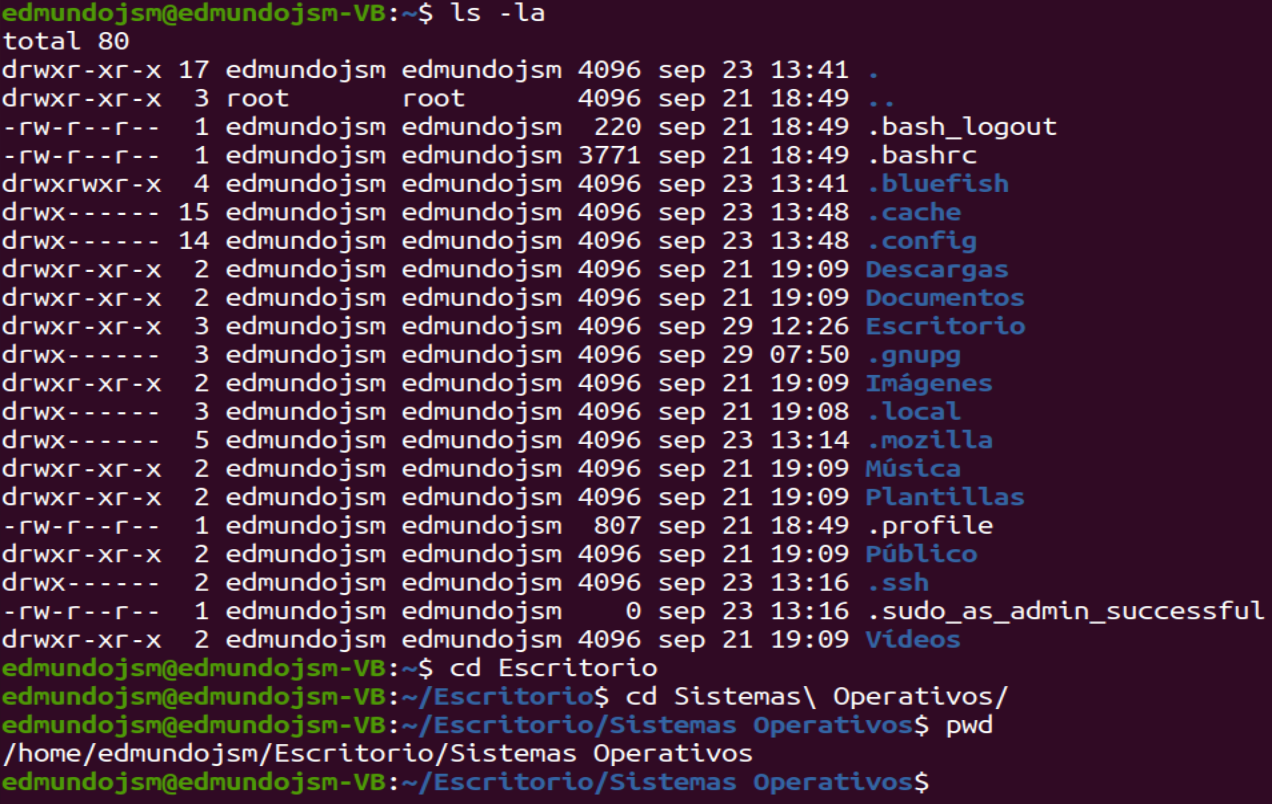


Figura 1: Antes de ejecutar el comando clear

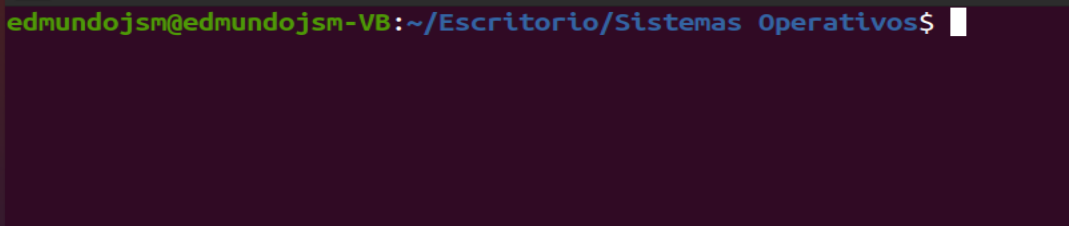
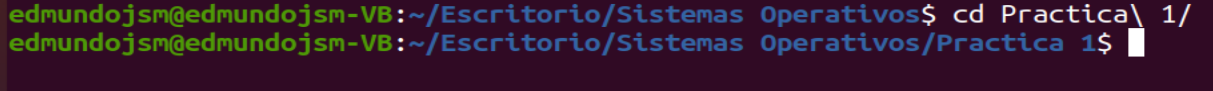
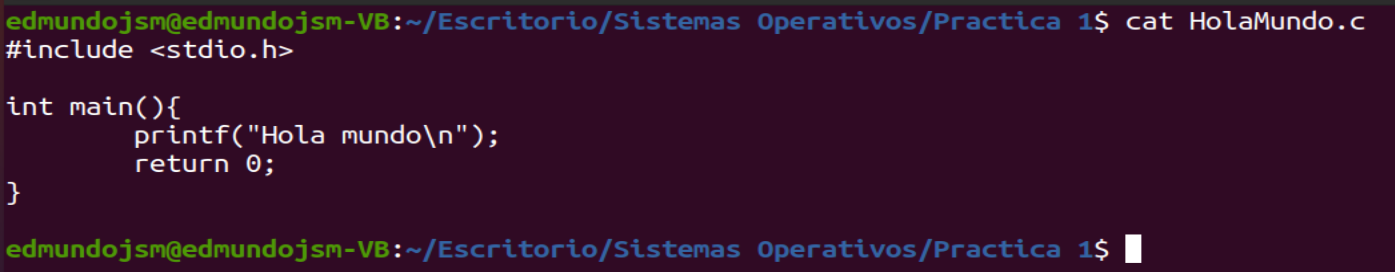


Figura 2: Después de ejecutar el comando clear

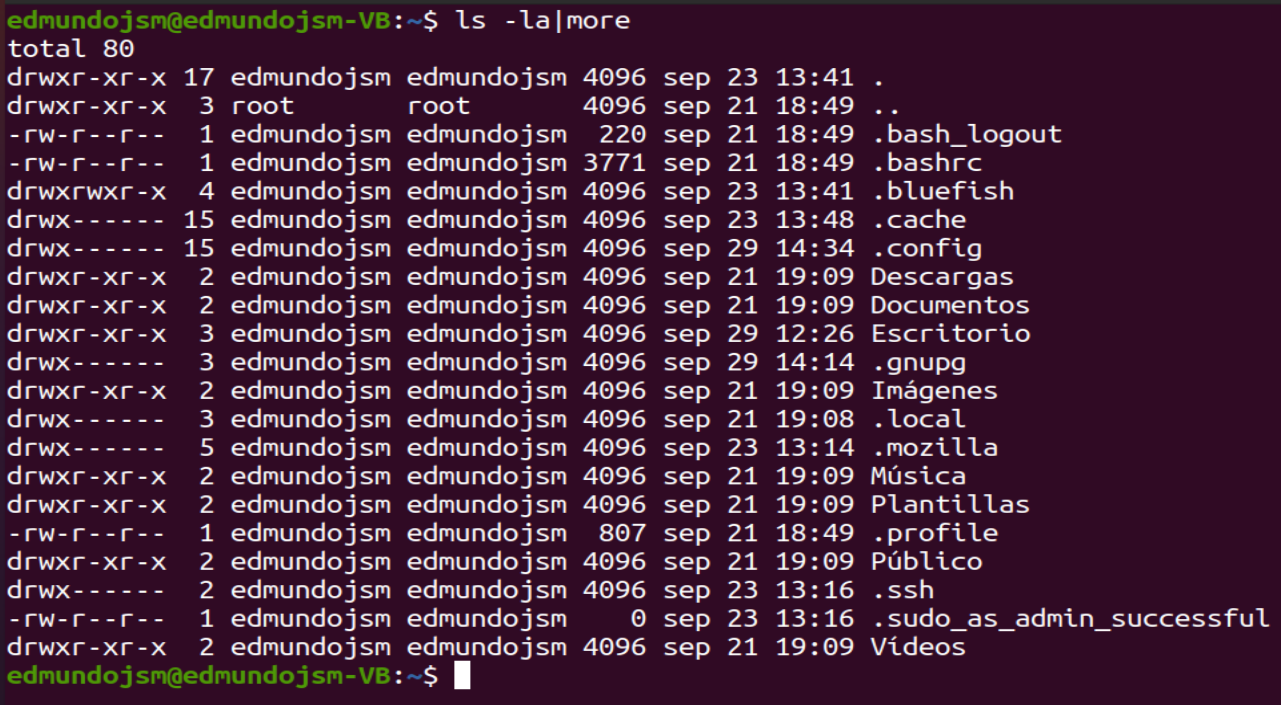
* Comando cd nombre-directorio



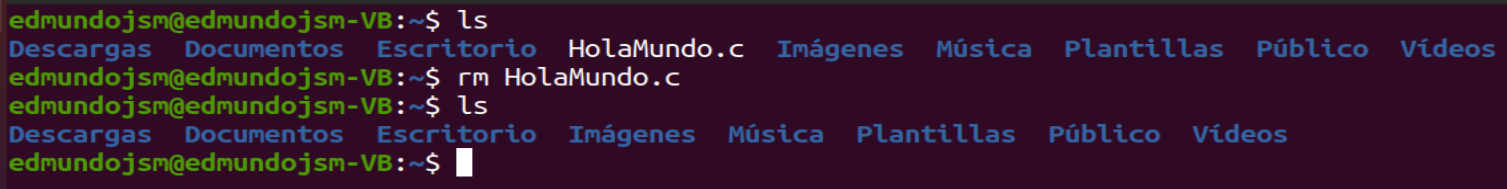
* Comando cat nombre-archivo



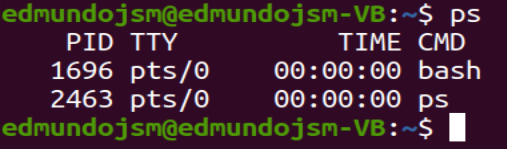
* Comando ls -la|more



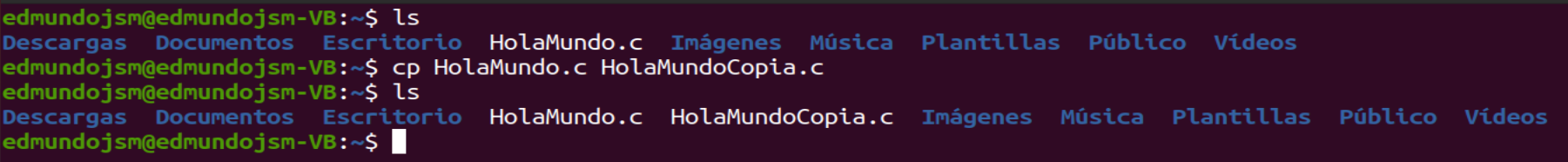
* Comando rm nombre\_archivo



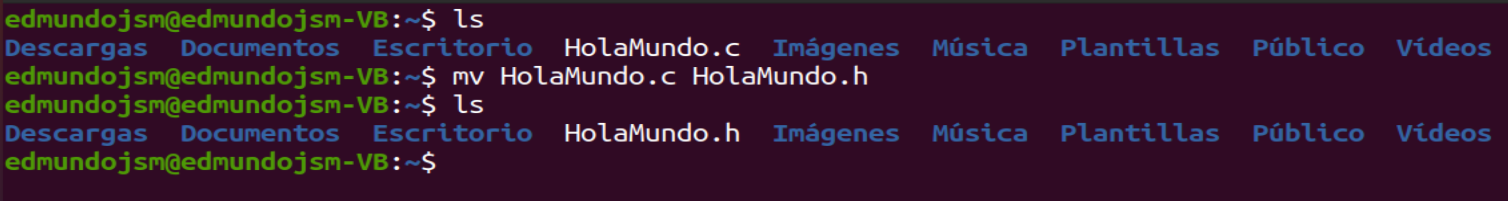
* Comando ps



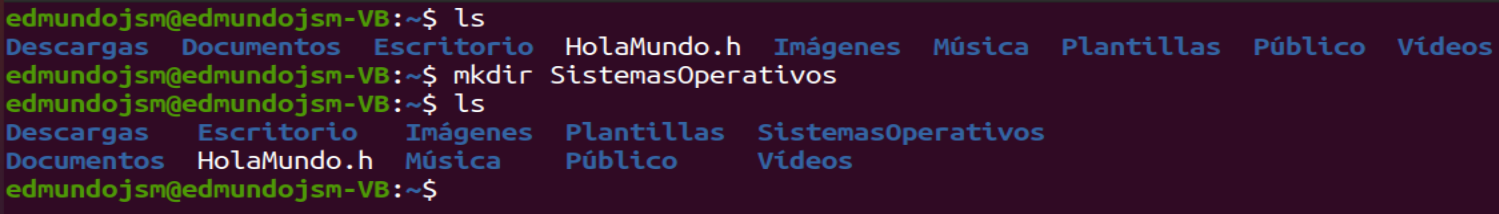
* Comando cp [ruta1] archivo1 [ruta2] archivo2



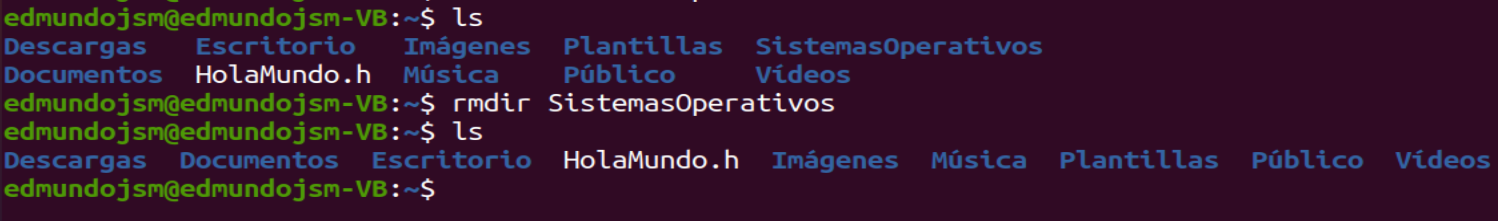
* Comando mv [ruta1] archivo1 [ruta2] archivo2



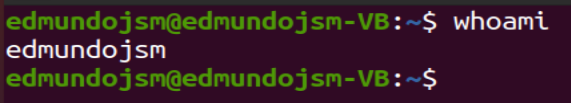
* Comando mkdir nombre\_directorio



* Comando rmdir nombre\_directorio



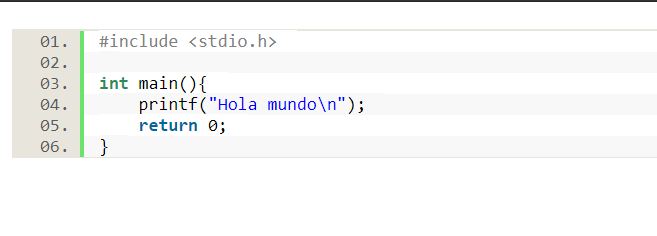
* Comando whoami



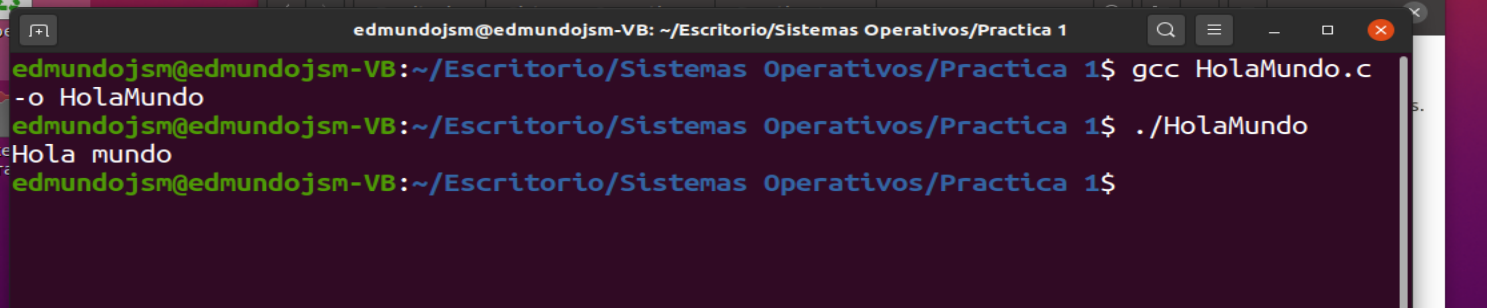
**2.6. Programas**

**2.6.1. Hola mundo**

Código (HolaMundo.c)



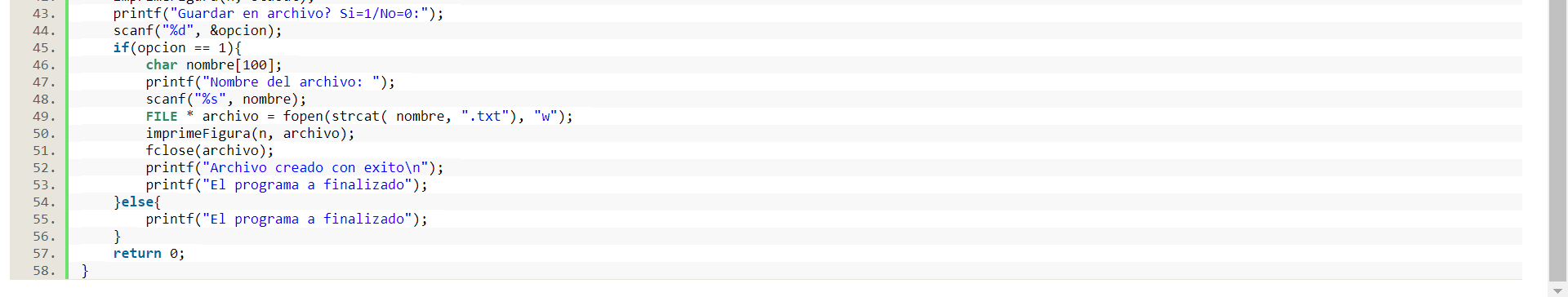
Compilación y ejecución:



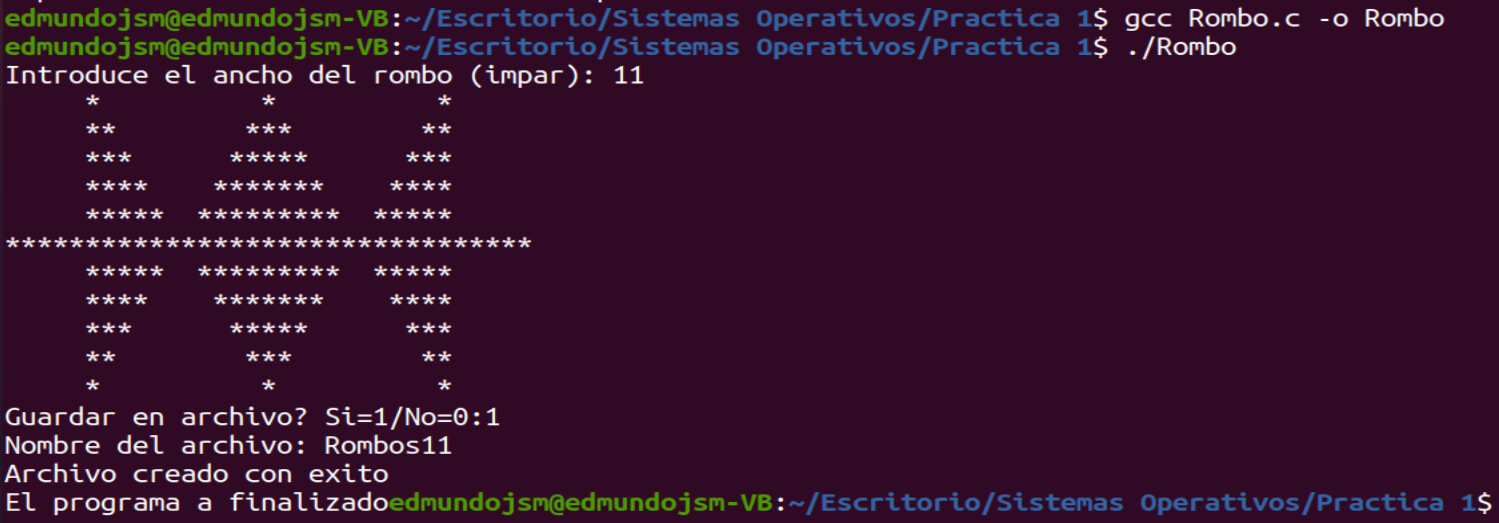
**2.6.3. Rombos**

Código (Rombo.c)

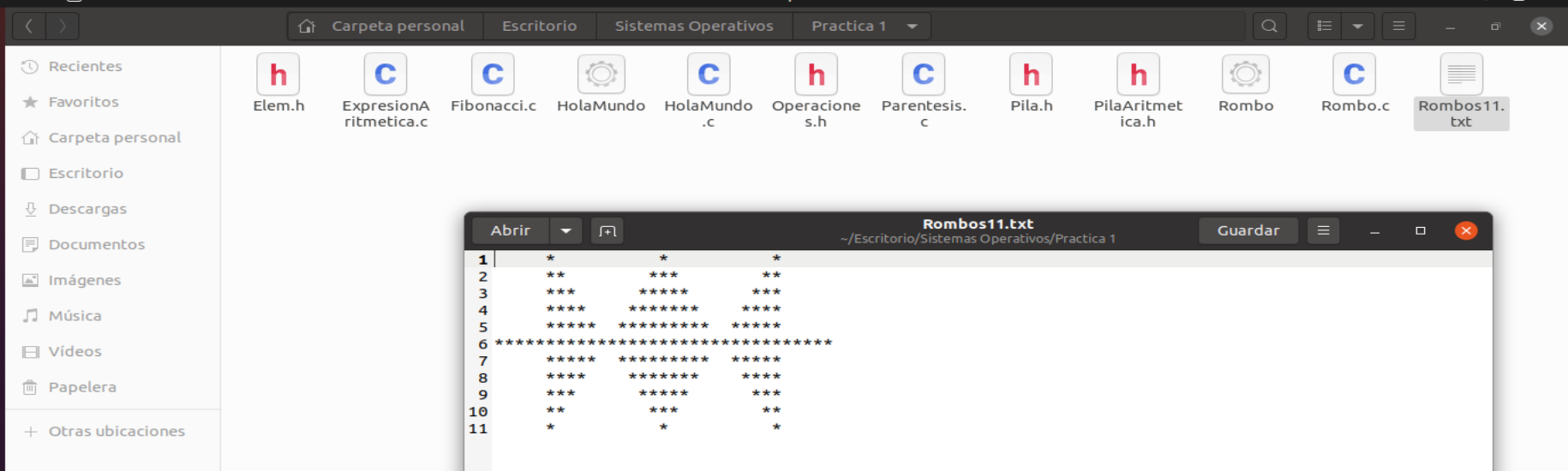




Compilación y ejecución para un ancho de rombo n = 11, mostrando en pantalla y luego exportando a un archivo:

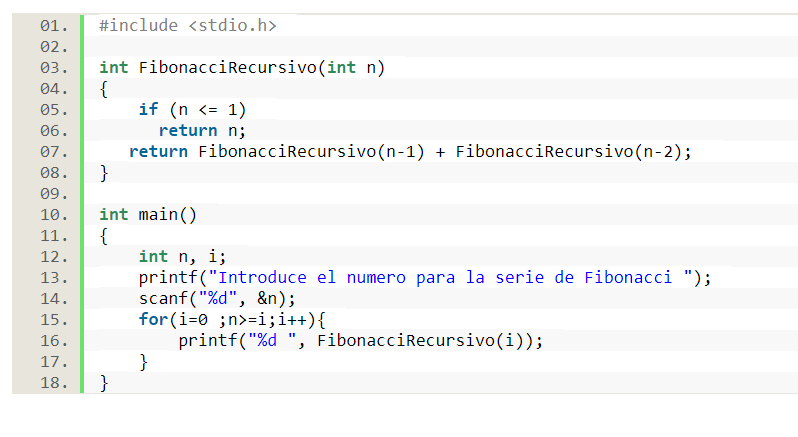


Vemos que el archivo se crea correctamente

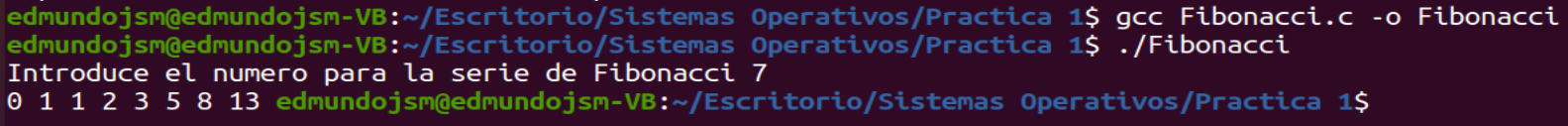


**2.6.3. Serie de Fibonacci**

Código (Fibonacci.c)

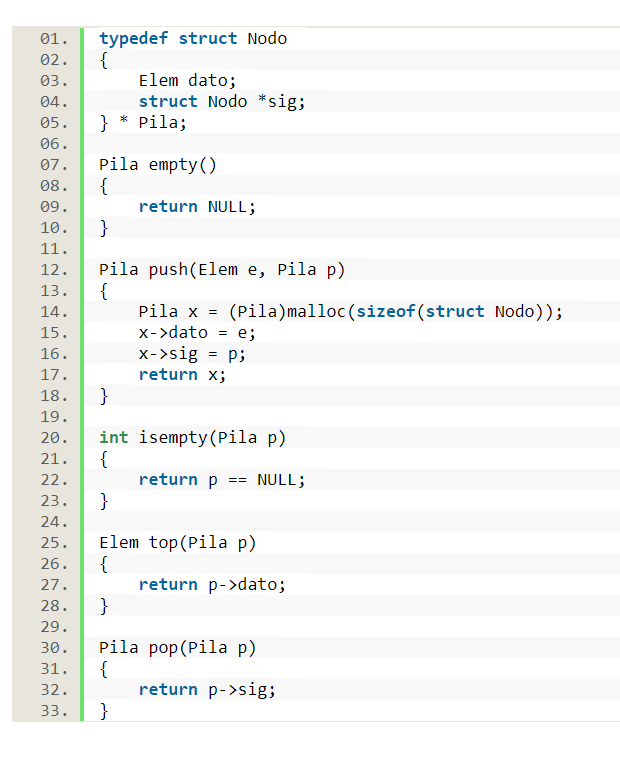


Compilación y ejecución para obtener la serie Fibonacci del número 7 (se escribe el 0 por definición de la serie de Fibonacci):



**2.6.4. Comprobación de emparejamiento de paréntesis**

Código parte 1 (Pila.h)

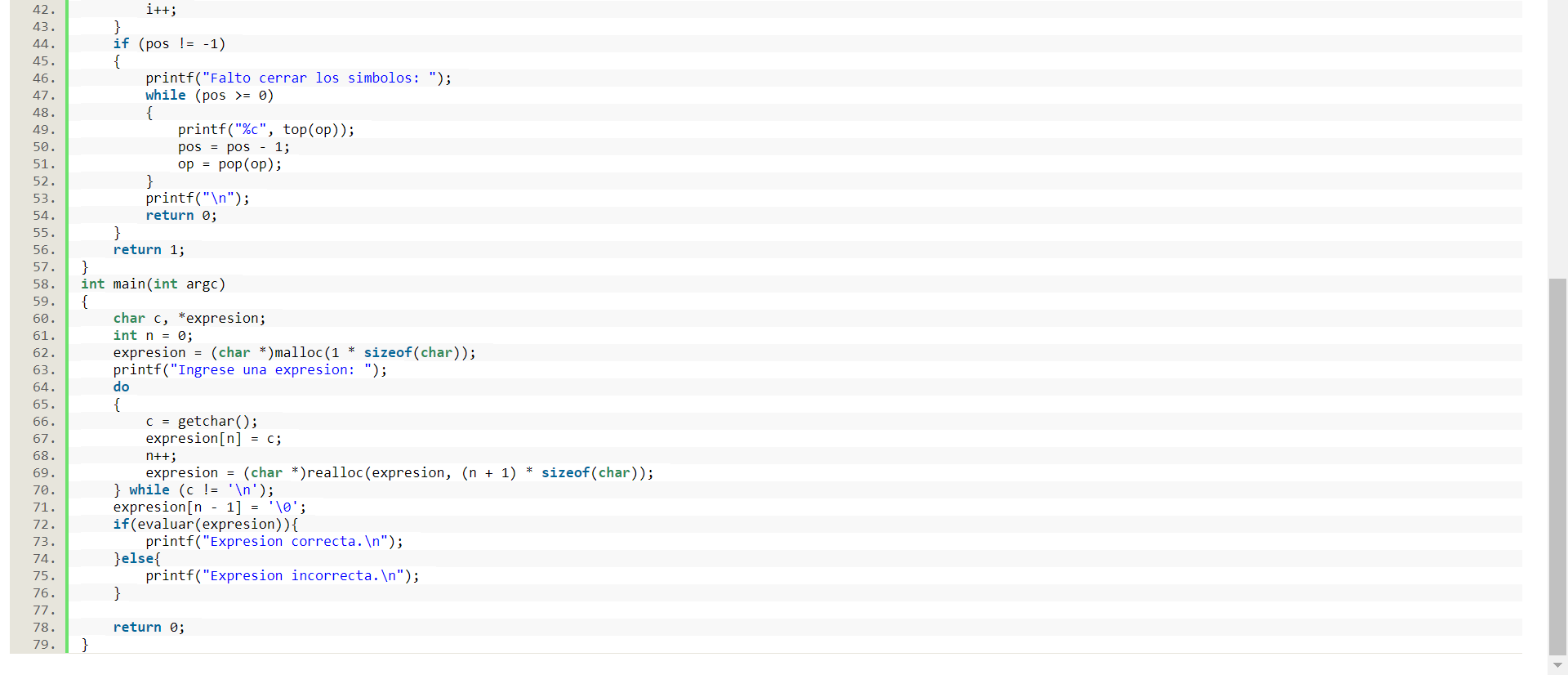


Código parte 2 (Elem.h)

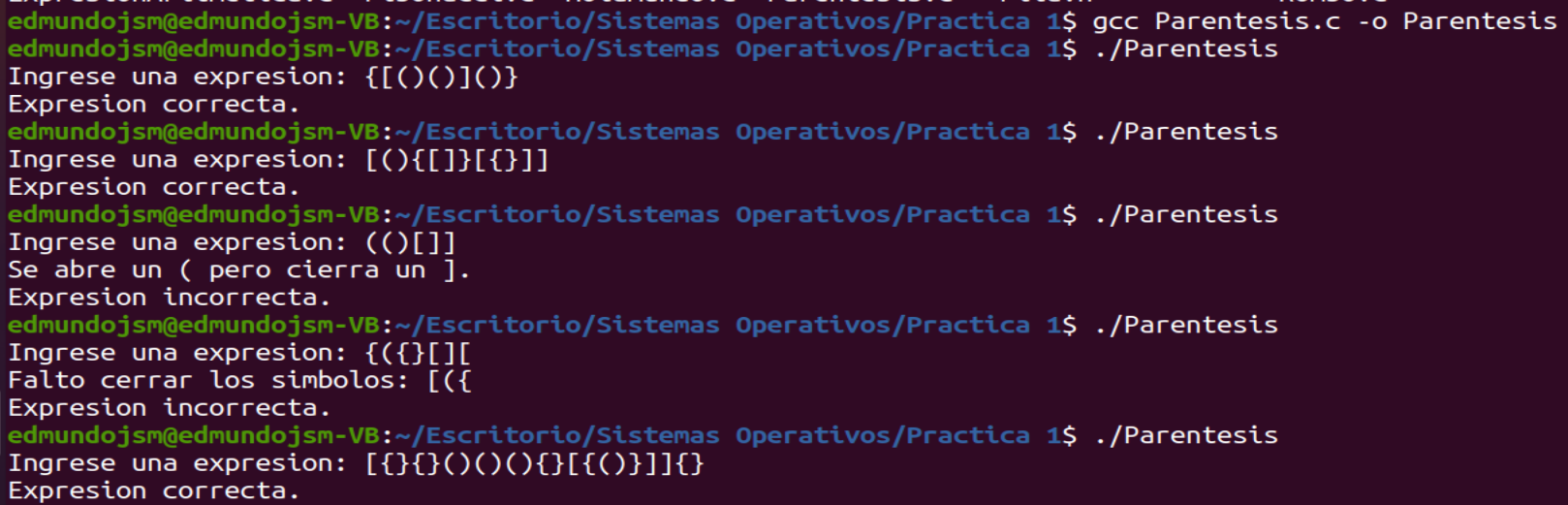


Código parte 3 (Parentesis.c)





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



**2.6.5. Evaluación de expresiones aritméticas**

Código parte 1 (PilaAritmetica.h)

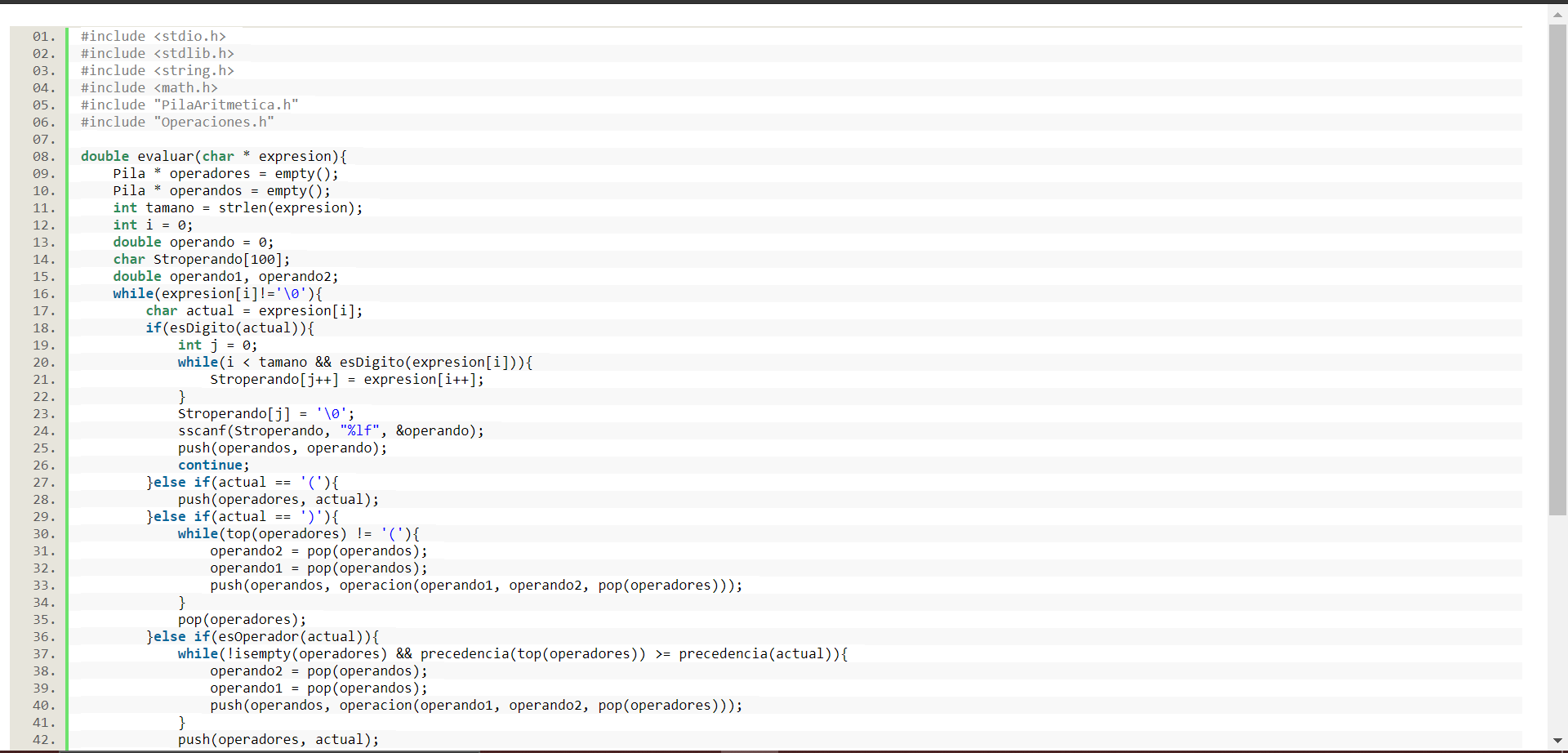


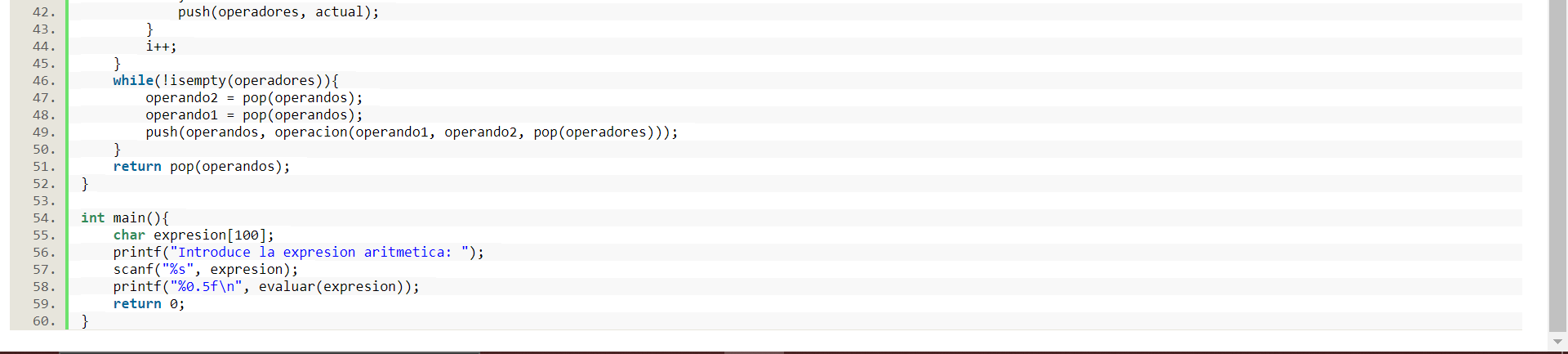
Código parte 2 (Operaciones.h)



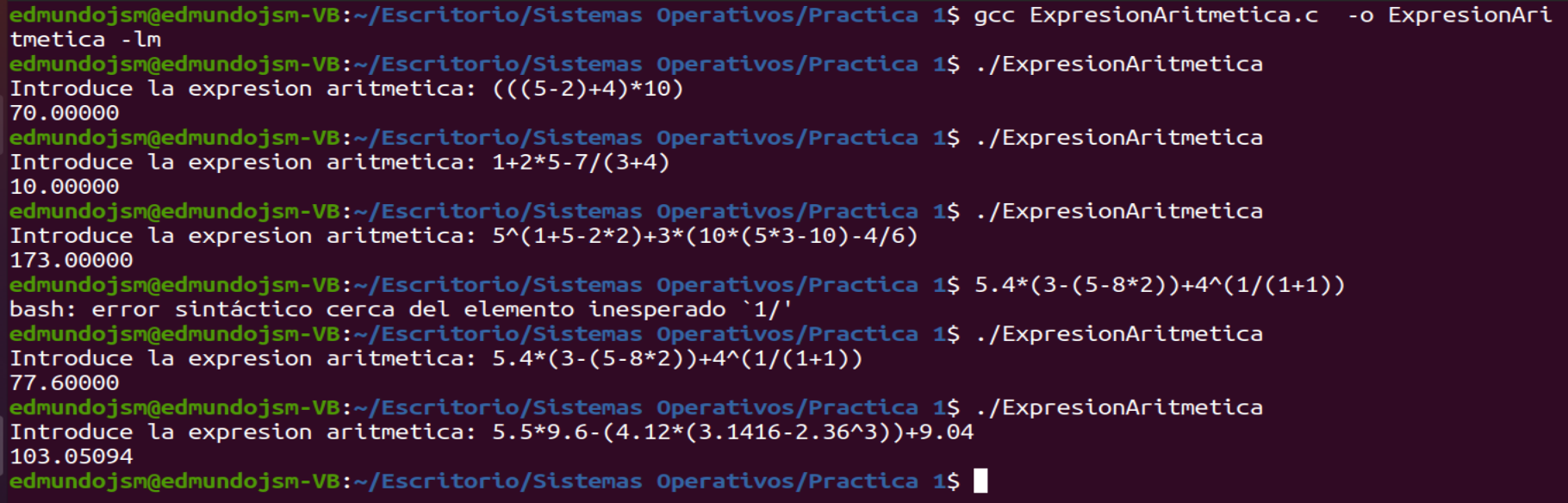


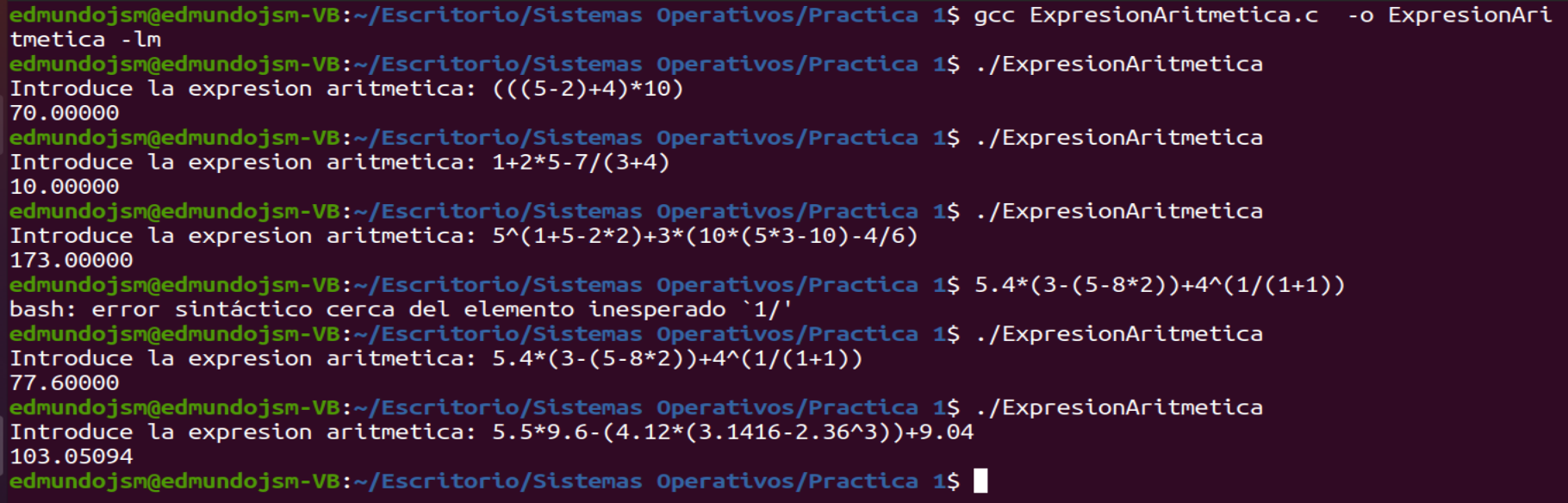
Código parte 3 (ExpresionAritmetica.c)





Compilación y ejecución para cinco casos de prueba:





**3. Desarrollo Windows**

**3.1. Comandos**

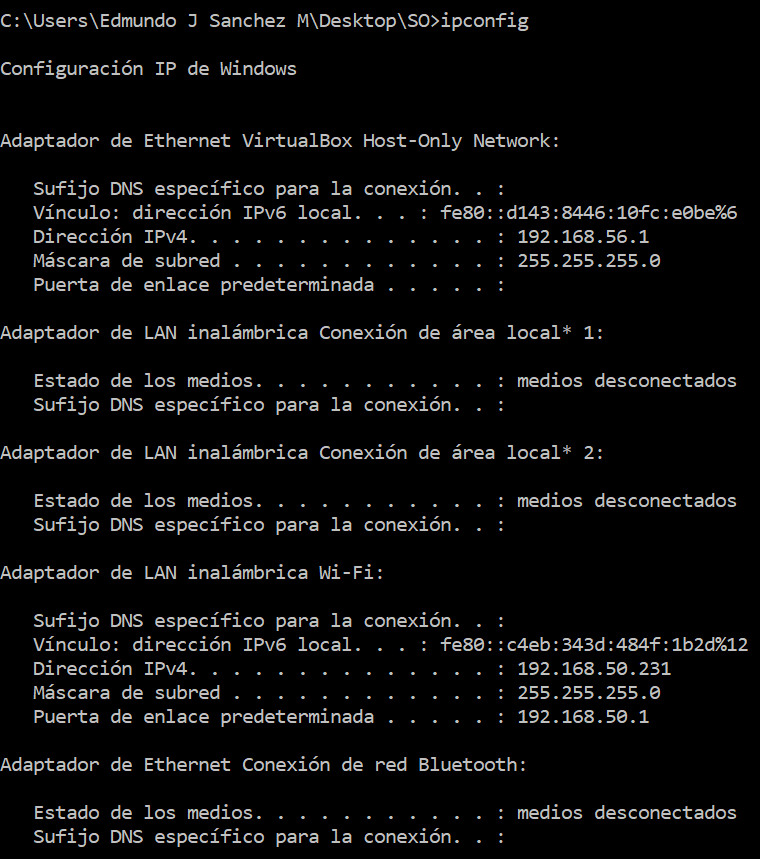
* **DIR**: Muestra una lista de archivos y subdirectorios en un directorio.
* **ipconfig**: Permite conocer la dirección IP y otra información relacionada con la configuración de internet del ordenador.
* **Cls**: Borra los símbolos o el texto en la pantalla de la consola.
* **Ver**: Muestra la versión 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



* Comando ipconfig



* Comando cls

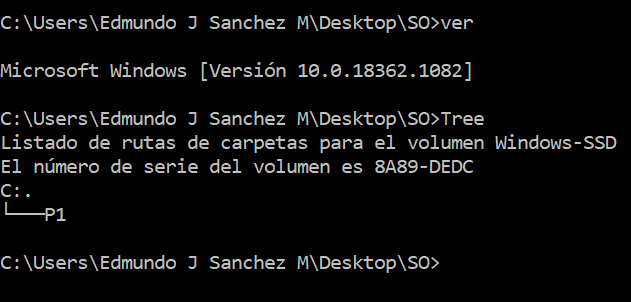


Figura 1: Antes de ejecutar el comando cls

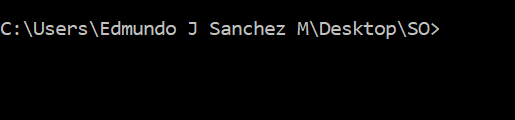
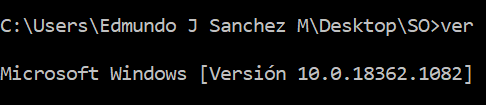
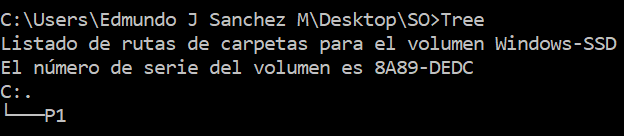


Figura 2: Después de ejecutar el comando cls

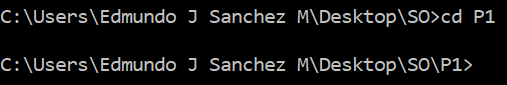
* Comando ver



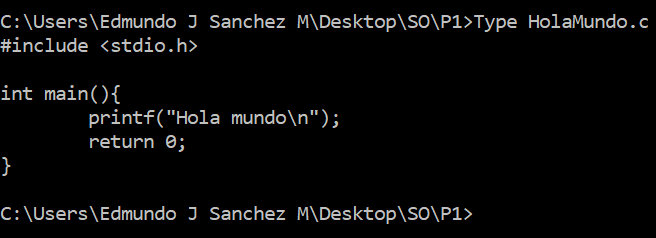
* Comando Tree



* Comando cd



* Comando Type



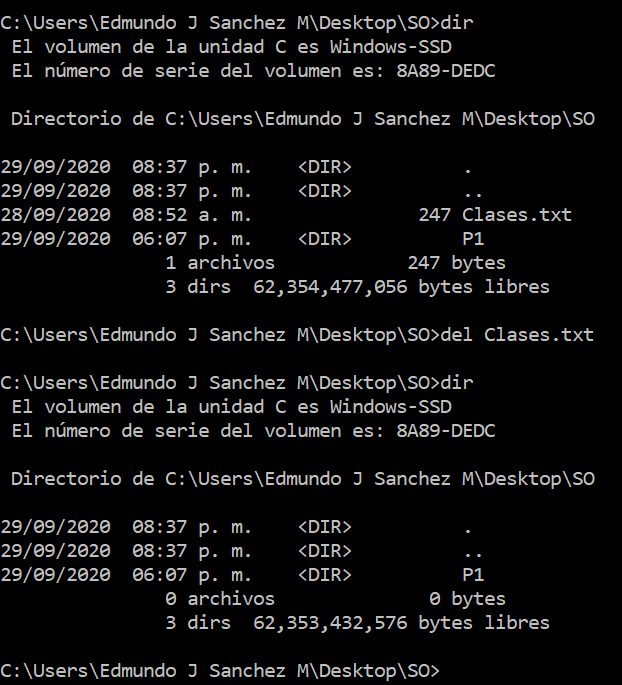
* Comando Mkdir



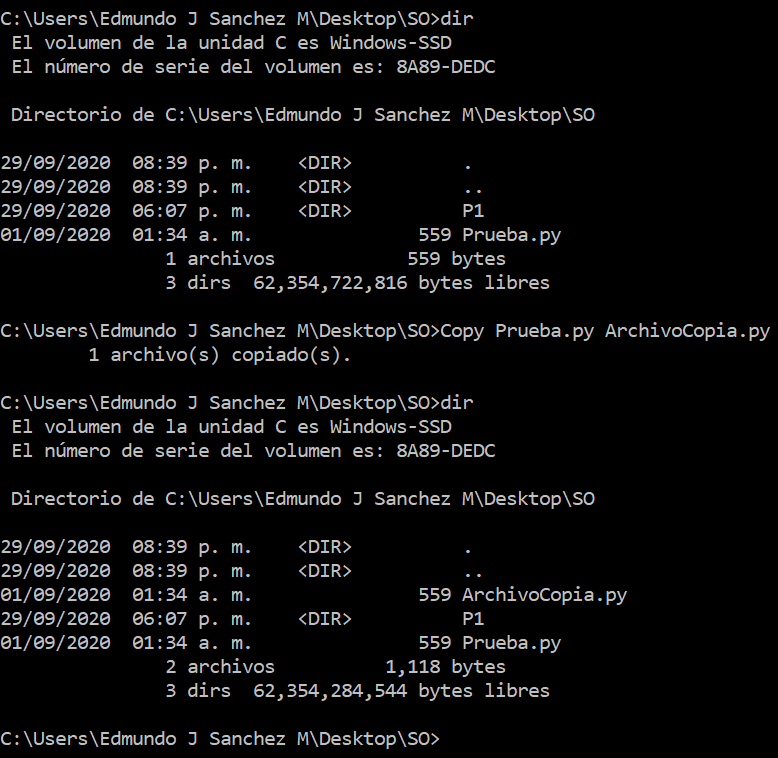
* Comando rmdir



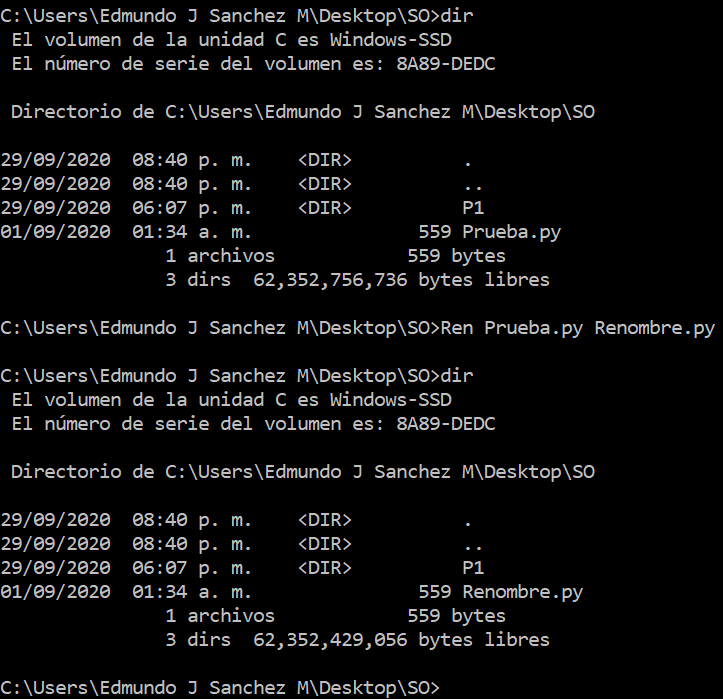
* Comando del



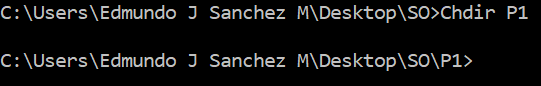
* Comando Copy



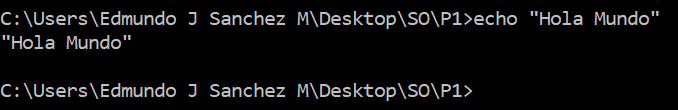
* Comando Ren



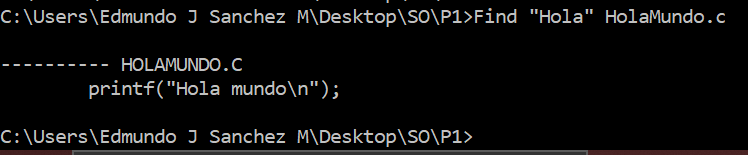
* Comando Chdir



* Comando echo



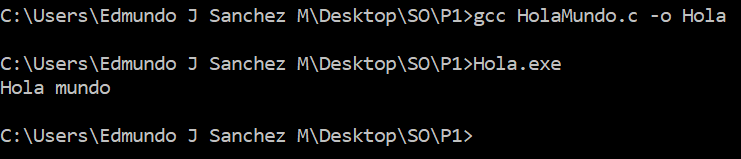
* Comando Find



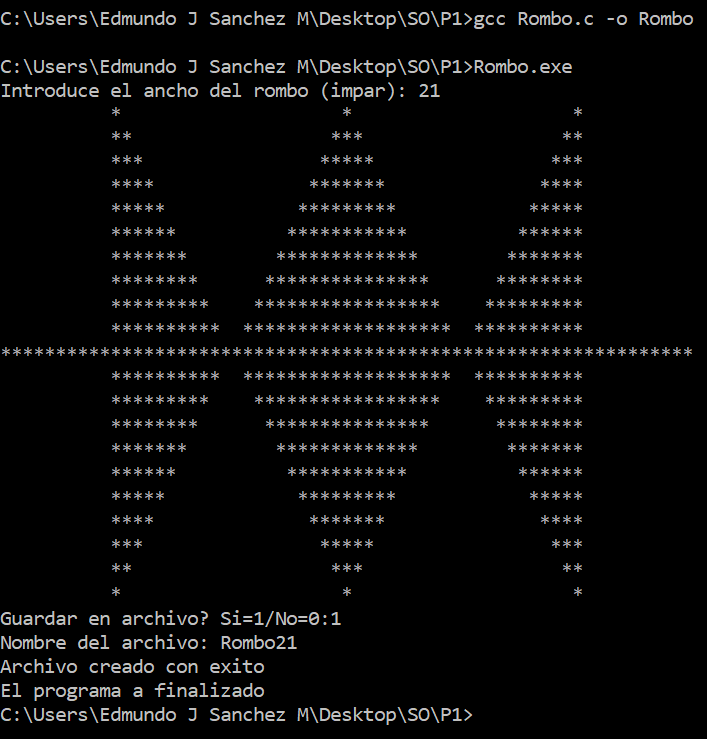
**3.3. Programas**

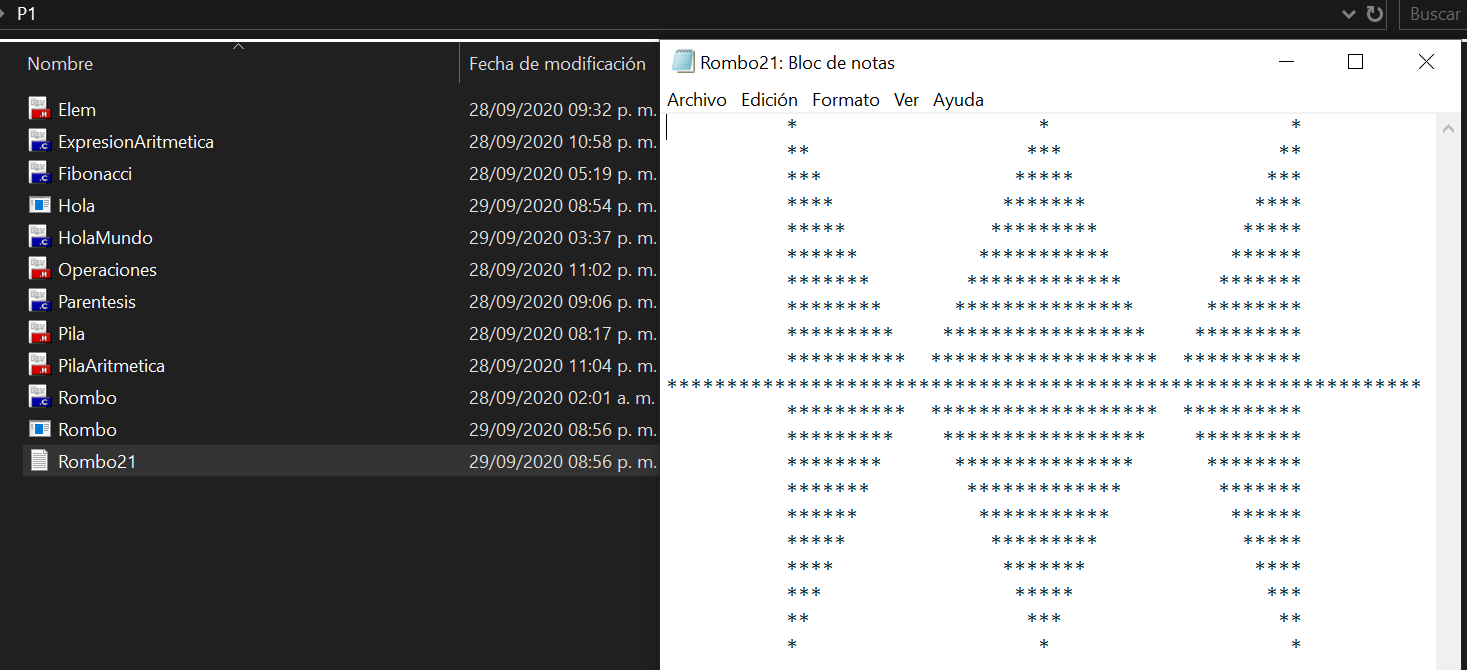
Nota: el Código es exactamente el mismo a los utilizados en Linux

**3.3.1. Hola mundo**



**3.3.2. Rombos**

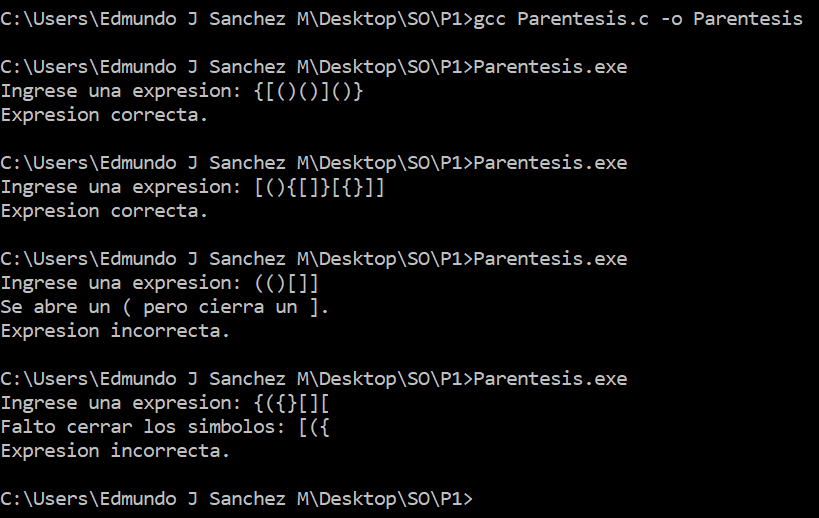




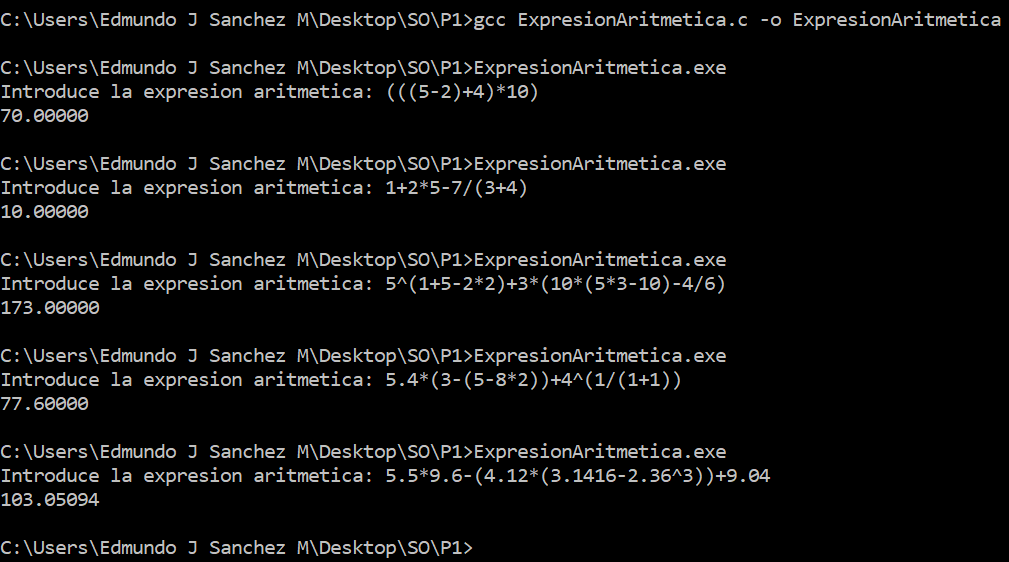
**3.3.3. Serie de Fibonacci**



**3.3.4. Comprobación de emparejamiento de paréntesis**



**3.3.5. Evaluación de expresiones aritméticas**



**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.