



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Karina García Morales

Asignatura: Fundamentos de programación

Grupo: 20

No. de práctica(s): 02

Integrante(s): Suzán Herrera Álvaro

No. de lista o brigada: 50

Semestre: 1

Fecha de entrega: 13 septiembre 2022

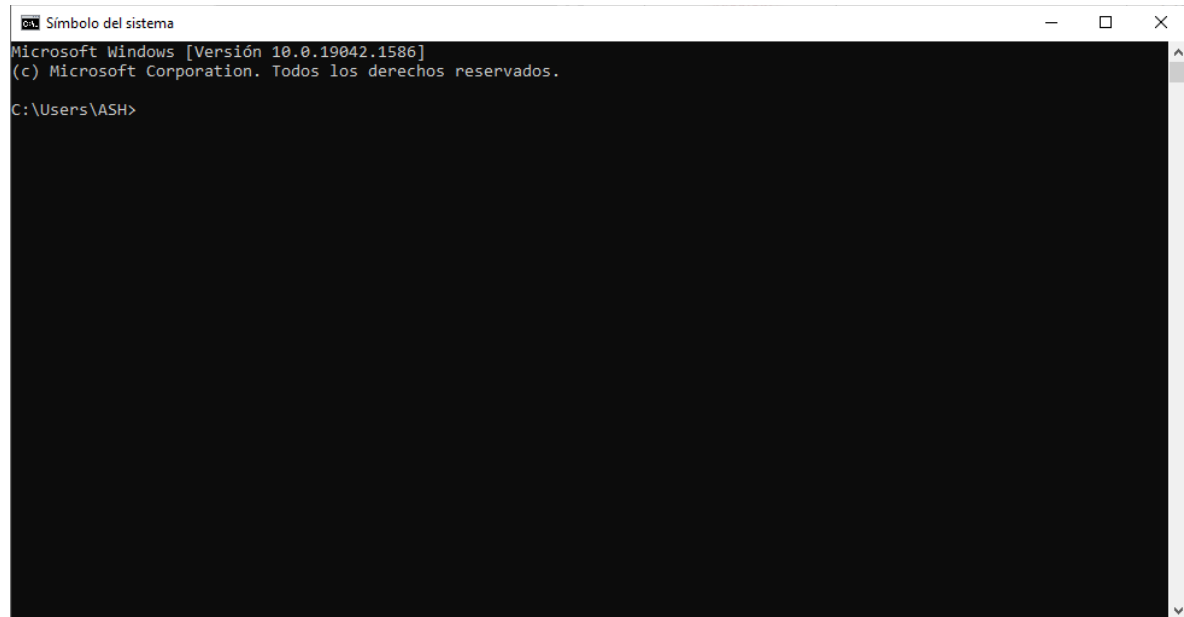
Observaciones:

CALIFICACIÓN: _____

Objetivo:

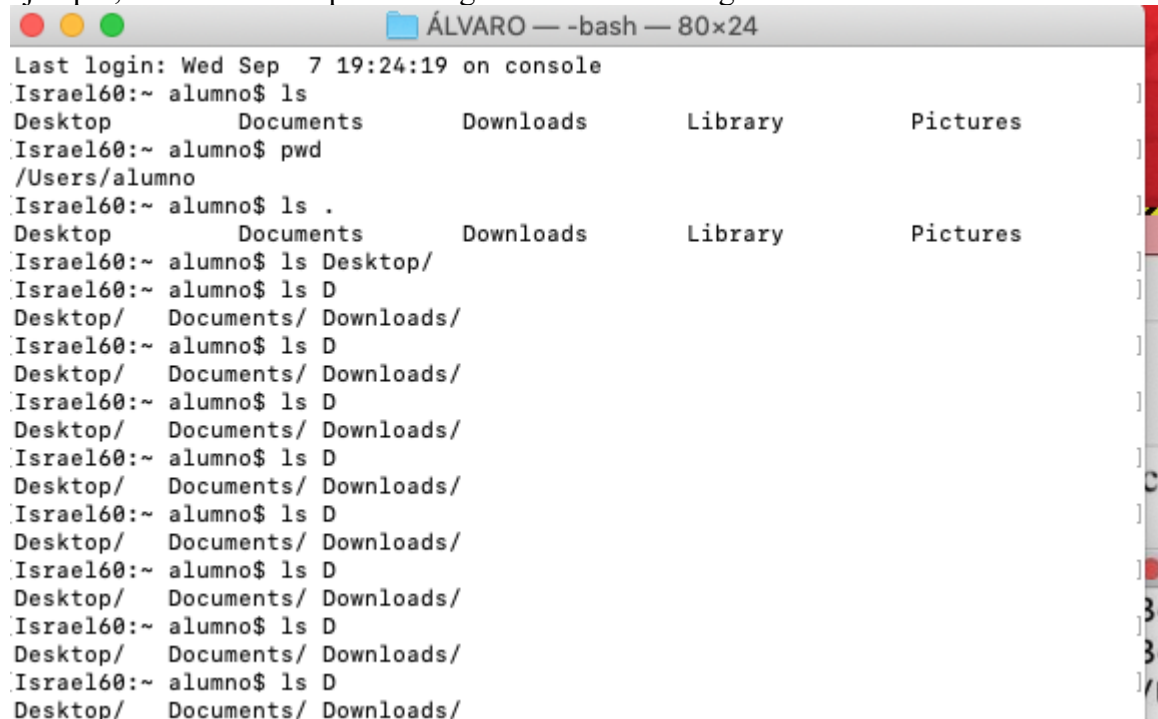
El alumno identificará al sistema operativo como una parte esencial de un sistema de cómputo. Explorará un sistema operativo GNU/Linux con el fin de conocer y utilizar sus comandos básicos.

Terminal



ls

El comando ls realiza acciones distintas dependiendo de las banderas que utilice, por ejemplo, si se utiliza la opción l se genera un listado largo de la ubicación actual:



ls -l

Es posible listar los elementos que existen en cualquier ubicación del sistema de archivos, para ello hay que ejecutar el comando especificando como argumento la ubicación donde se desean listar los elementos.

```
alumno — -bash — 80x24
[Israel60:~ alumno$ -l /
-bash: -l: command not found
[Israel60:~ alumno$ ls -l /
total 13
drwxrwxr-x+ 66 root  admin   2112 Feb 22  2022 Applications
drwxrwxrwx   8 root  wheel    256 Nov 13  2019 Insight Files
drwxr-xr-x   6 root  wheel    192 Feb 21  2022 LC_AB
drwxr-xr-x+ 64 root  wheel   2048 Aug 10 13:23 Library
drwxr-xr-x@  2 root  wheel    64 Feb 25  2019 Network
drwxr-xr-x@  5 root  wheel   160 May  4  2019 System
drwxr-xr-x   7 root  admin   224 Sep  7 11:02 Users
drwxr-xr-x@  3 root  wheel    96 Sep  7 19:07 Volumes
drwxr-xr-x  27 root  wheel   864 Jun 25  2019 anaconda3
drwxr-xr-x@ 37 root  wheel  1184 May  4  2019 bin
drwxrwxr-x-t@ 2 root  admin    64 Feb 25  2019 cores
dr-xr-xr-x   3 root  wheel   4421 Sep  7 19:03 dev
lrwxr-xr-x@  1 root  wheel    11 Jun 17  2019 etc -> private/etc
dr-xr-xr-x   2 root  wheel     1 Sep  7 19:29 home
-rw-r--r--@  1 root  wheel   313 Feb 24  2019 installer.failurerequest
dr-xr-xr-x   2 root  wheel     1 Sep  7 19:29 net
drwxr-xr-x   6 root  wheel   192 Aug  8 18:49 private
drwxr-xr-x@ 64 root  wheel  2048 Jun 17  2019 sbin
lrwxr-xr-x@  1 root  wheel    11 Jun 17  2019 tmp -> private/tmp
```

ls /

Para ver los usuarios del equipo local, revisamos el directorio home que parte de la raíz (/)

ls /home

Tanto las opciones como los argumentos se pueden combinar para generar una ejecución más específica:

ls -l /home

GNU/Linux proporciona el comando man, el cual permite visualizar la descripción de cualquier comando, así como la manera en la que se puede utilizar

```
lrwxr-xr-x@ 1 root wheel 11 Jun 17 2019 var -> private/var
[Israel60:~ alumno$ ls /com ]
ls: /com: No such file or directory
[Israel60:~ alumno$ ls /home ]
[Israel60:~ alumno$ ls /Users/ ]
Shared administrador alumno curso fp06alu21
[Israel60:~ alumno$ mkdir ÀLVARO ]
[Israel60:~ alumno$ mkdir ÁLVARO ]
[Israel60:~ alumno$ pwd ]
/Users/alumno
[Israel60:~ alumno$ mnkdir ÁLVARO ]
-bash: mnkdir: command not found
[Israel60:~ alumno$ mkdir ÁLVARO ]
mkdir: ÁLVARO: File exists
[Israel60:~ alumno$ ls ]
Desktop Downloads Pictures ÁLVARO
Documents Library ÁLVARO
[Israel60:~ alumno$ mkdir SUZÁN HERRERA ]
[Israel60:~ alumno$ ls ]
Desktop Downloads Library SUZÁN ÁLVARO
Documents HERRERA Pictures ÁLVARO
[Israel60:~ alumno$ touch ÁLGEBRA ]
[Israel60:~ alumno$ touch CÁLCULO ]
[Israel60:~ alumno$ touch REDACCIÓN ]
```

man ls

Antes de revisar otros comandos, es importante aprender a “navegar” por el sistema de archivos de Linux en modo texto. Basándonos en la Figura 2 de esta práctica, si deseamos ver la lista de los archivos del directorio usr, podemos escribir el comando:

ls /usr

Esto es, el argumento se inicia con / indicando que es el directorio raíz, seguido de usr que es el nombre del directorio. Cuando especificamos la ubicación de un archivo partiendo de la raíz, se dice que estamos indicando la “ruta absoluta” del archivo.

Existe otra forma de especificar la ubicación de un archivo, esto es empleando la “ruta relativa”. Si bien el punto (.) es para indicar la ubicación actual, el doble punto (..) se utiliza para referirse al directorio “padre”. De esta forma si deseamos listar los archivos que dependen de mi directorio padre se escribe el siguiente comando: **ls ..**

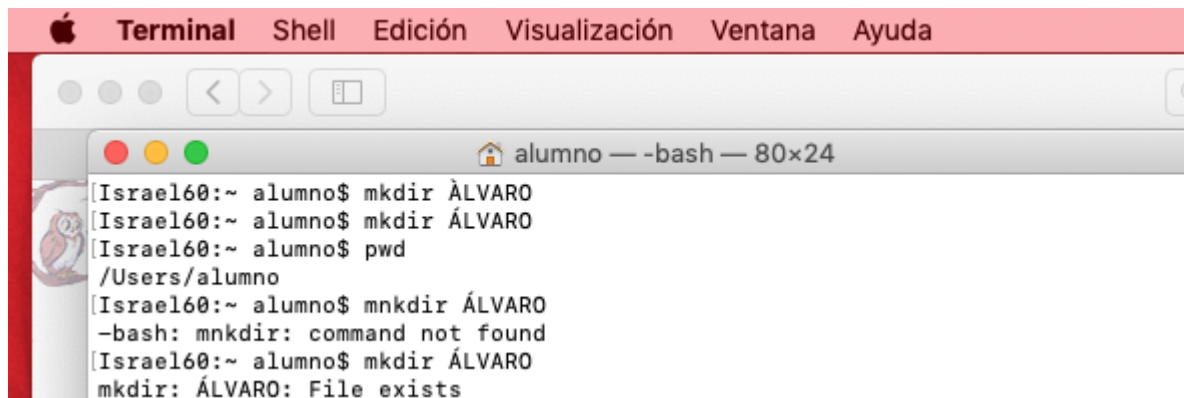
El comando touch permite crear un archivo de texto, su sintaxis es la siguiente:

Touch nombre_archivo[.ext]

```
Israel60:~ alumno$ touch ÁLGEBRA
Israel60:~ alumno$ touch CÁLCULO
Israel60:~ alumno$ touch REDACCIÓN
Israel60:~ alumno$ touch PROGRAMACIÓN
Israel60:~ alumno$ LS
CÁLCULO      Downloads      PROGRAMACIÓN  SUZÁN        ÁLVARO
Desktop      HERRERA        Pictures      ÁLVARO
Documents    Library        REDACCIÓN    ÁLGEBRA
Israel60:~ alumno$ cd ÁLVARO
Israel60:ÁLVARO alumno$ pwd
/Users/alumno/ÁLVARO
Israel60:ÁLVARO alumno$ cd ..
Israel60:~ alumno$
```

El comando mkdir permite crear una carpeta, su sintaxis es la siguiente:

Mkdir nombre_carpeta



```
Terminal  Shell  Edición  Visualización  Ventana  Ayuda
alumno — -bash — 80x24
[Israel60:~ alumno$ mkdir ÁLVARO
[Israel60:~ alumno$ mkdir ÁLVARO
[Israel60:~ alumno$ pwd
/Users/alumno
[Israel60:~ alumno$ mnkdir ÁLVARO
-bash: mnkdir: command not found
[Israel60:~ alumno$ mkdir ÁLVARO
mkdir: ÁLVARO: File exists
```

Para crear una carpeta en nuestra cuenta, que tenga como nombre “tareas” se escribe el siguiente comando:

Mkdir tareas

```
[Israel60:~ alumno$ mkdir TAREAS
[Israel60:~ alumno$ ls
```

El comando cd permite ubicarse en una carpeta, su sintaxis es la siguiente:

cd nombre_carpeta

```
[Israel60:~ alumno$ cd TAREAS
[Israel60:TAREAS alumno$ ls
CÁLCULO      PROGRAMACIÓN  REDACCIÓN    ÁLGEBRA
[Israel60:TAREAS alumno$ rm CÁLCULO PROGRAMACIÓN REDACCIÓN ÁLGEBRA
[Israel60:TAREAS alumno$ ls
[Israel60:TAREAS alumno$
```

Ahora, si deseamos situarnos en la carpeta de inicio de nuestra cuenta, que es la carpeta padre, escribimos el comando:

cd ..

```
Israel60:ÁLVARO alumno$ cd ..  
Israel60:~ alumno$
```

El comando pwd permite conocer la ubicación actual(ruta), su sintaxis es la siguiente:

Pwd

```
[Israel60:~ alumno$ cd ÁLVARO  
Israel60:ÁLVARO alumno$ pwd  
/Users/alumno/ÁLVARO  
Israel60:ÁLVARO alumno$
```

El comando find permite buscar un elemento dentro del sistema de archivos, su sintaxis es la siguiente:

find . -name cadena_buscar

```
Israel60:Users alumno$ find . -name Á*  
find: ./administrador/.config: Permission denied  
find: ./administrador/Music: Permission denied  
find: ./administrador/.local: Permission denied  
find: ./administrador/Pictures: Permission denied  
find: ./administrador/Desktop: Permission denied  
find: ./administrador/Library: Permission denied
```

El comando clear permite limpiar la consola o terminal, su sintaxis es la siguiente:

clear

```
[root@localhost ~]# hjsdfhsdf  
sh: hjsdfhsdf: command not found  
[root@localhost ~]# sdfsdgfas  
sh: sdfsdgfas: command not found  
[root@localhost ~]# fa  
sh: fa: command not found  
[root@localhost ~]# fasd  
sh: fasd: command not found  
[root@localhost ~]# fasdf  
sh: fasdf: command not found  
[root@localhost ~]# clear
```

```
[root@localhost ~]#
```

El comando cp permite copiar un archivo, su sintaxis es la siguiente:

cp archivo_origen archivo_destino

```
[Israel60:~ alumno$ cp REDACCIÓN /Users/alumno/SUZÁN ]
[Israel60:~ alumno$ ls ]
CÁLCULO      Downloads      PROGRAMACIÓN  SUZÁN          ÁLVARO
Desktop      HERRERA        Pictures      ÁLVARO
Documents    Library        REDACCIÓN     ÁLGBRA
[Israel60:~ alumno$ cp ÁLGBRA /Users/alumno/SUZÁN ]
[Israel60:~ alumno$ cp PROGRAMACIÓN /Users/alumno/HERRERA ]
[Israel60:~ alumno$ ls SUZÁN ]
REDACCIÓN    ÁLGBRA
```

El comando mv mueve un archivo de un lugar a otro, en el sistema de archivos; su sintaxis es la siguiente:

mv ubicación_origen/archivo ubicación_destino

El comando mueve el archivo desde su ubicación origen hacia la ubicación deseada(destino).

```
[Israel60:~ alumno$ mv CÁLCULO /Users/alumno/TAREAS ]
[Israel60:~ alumno$ mv REDACCIÓN /Users/alumno/TAREAS ]
[Israel60:~ alumno$ mv ÁLGBRA /Users/alumno/TAREAS ]
[Israel60:~ alumno$ mv PROGRAMACIÓN /Users/alumno/TAREAS ]
[Israel60:~ alumno$ ls ]
Desktop      Downloads      Library        SUZÁN          ÁLVARO
Documents    HERRERA        Pictures      TAREAS         ÁLVARO
[Israel60:~ alumno$ ls TAREAS ]
CÁLCULO      PROGRAMACIÓN  REDACCIÓN     ÁLGBRA
[Israel60:~ alumno$ ]
```

rm nombre_archivo rm nombre_carpeta

```

Israel60:~ alumno$ rm SUZÁN
rm: SUZÁN: is a directory
Israel60:~ alumno$ rm -r SUZÁN
Israel60:~ alumno$ █

Israel60:~ alumno$ rm -r ÁLVARO SUZÁN HERRERA
rm: SUZÁN: No such file or directory
Israel60:~ alumno$ ls
Desktop          Downloads        Pictures        ÁLVARO
Documents        Library         TAREAS
Israel60:~ alumno$ rm -r ÁLVARO
rm: ÁLVARO: No such file or directory
Israel60:~ alumno$ ls
Desktop          Downloads        Pictures        ÁLVARO
Documents        Library         TAREAS
Israel60:~ alumno$ ls
Desktop          Downloads        Pictures        ÁLVARO
Documents        Library         TAREAS
Israel60:~ alumno$ rm -r ÁLVARO
Israel60:~ alumno$ ls
Desktop          Downloads        Pictures
Documents        Library         TAREAS
Israel60:~ alumno$ rm -r TAREAS
Israel60:~ alumno$ ls
Desktop          Documents        Downloads        Library        Pictures
Israel60:~ alumno$ █

```

Podemos crear directorios (carpetas) directamente desde la consola/terminal con el comando: **- mkdir**

```
[root@localhost ~]# mkdir LAB
[root@localhost ~]# ls
bench.py  hello.c  LAB
[root@localhost ~]#
```


Podemos ingresar a esta carpeta mediante el comando: **cd**, para ingresar a LAB, ingresamos: **“cd LAB”** y comprobamos con **ls**
Creamos el directorio "PRACTICA_LINUX" con **mkdir**

```
[root@localhost LAB]# mkdir PRACTICA_LINUX
[root@localhost LAB]# ls
Linux  PRACTICA
[root@localhost LAB]#
```

Entramos al nuevo directorio y creamos los archivos con el comando: **touch**; que a diferencia de **mkdir** crea archivos ejecutables que pueden contener diferentes extensiones. Se crean 3 archivos: “Álvaro”, “Materias”, “Datos Álvaro”

```
[root@localhost LAB]# cd PRACTICA_LINUX
[root@localhost PRACTICA_LINUX]# ls
[root@localhost PRACTICA_LINUX]# touch Álvaro
[root@localhost PRACTICA_LINUX]# touch Materias
[root@localhost PRACTICA_LINUX]# touch Datos_Álvaro
[root@localhost PRACTICA_LINUX]# ls
Álvaro  Datos_Álvaro  Materias
[root@localhost PRACTICA_LINUX]#
```

Salimos del directorio con **cd ..**

```
[root@localhost PRACTICA_LINUX]# cd ..  
[root@localhost LAB]#
```

En LAB usaremos el comando para copiar, creamos un nuevo directorio llamado COPIA, luego, dentro de este directorio copiaremos los archivos de la carpeta principal.

```
[root@localhost PRACTICA_LINUX]# cp Álvaro /root/LAB/LAB/COPIA  
[root@localhost PRACTICA_LINUX]# cp Materias /root/LAB/LAB/COPIA  
[root@localhost PRACTICA_LINUX]# cp Datos_Álvaro /root/LAB/LAB/COPIA  
[root@localhost PRACTICA_LINUX]# cd ..  
[root@localhost LAB]# ls  
COPIA PRACTICA_LINUX
```

Contenido de COPIA

```
[root@localhost LAB]# cd COPIA  
[root@localhost COPIA]# ls  
Álvaro Datos_Álvaro Materias  
[root@localhost COPIA]#
```

Ahora movemos el archivo Datos_Álvaro a la COPIA, usando el comando **mv**

```
[root@localhost PRACTICA_LINUX]# mv Datos_Álvaro /root/LAB/LAB/COPIA  
[root@localhost PRACTICA_LINUX]# ls  
Álvaro Materias  
[root@localhost PRACTICA_LINUX]# cd ..  
[root@localhost LAB]# cd COPIA  
[root@localhost COPIA]# ls  
Álvaro Datos_Álvaro Materias  
[root@localhost COPIA]#
```

Permisos

Para ver los permisos de una carpeta en específico se utiliza el comando **ls -l**

Veremos los permisos de LAB y COPIA

```
[root@localhost COPIA]#  
[root@localhost COPIA]# ls -l  
total 0  
-rw-r--r-- 1 root root 0 Sep 13 12:51 Álvaro  
-rw-r--r-- 1 root root 0 Sep 13 12:23 Datos_Álvaro  
-rw-r--r-- 1 root root 0 Sep 13 12:52 Materias  
[root@localhost COPIA]# cd ..  
[root@localhost LAB]# ls -l  
total 8  
drwxr-xr-x 2 root root 116 Sep 13 12:32 COPIA  
drwxr-xr-x 2 root root 86 Sep 13 12:23 PRACTICA_LINUX  
[root@localhost LAB]#
```

Comandos auxiliares y de ubicación

Para revisar las rutas del archivo o directorio utilizamos el comando: **pwd**. Para visualizar este comando lo aplicaremos en la carpeta LAB y COPIA

```
[root@localhost LAB]# pwd
/root/LAB/LAB
[root@localhost LAB]# cd COPIA
[root@localhost COPIA]# pwd
/root/LAB/LAB/COPIA
[root@localhost COPIA]#
```

Observamos que COPIA pertenece a LAB pues la ruta de COPIA pasa por el directorio LAB

Comando cal

```
[root@localhost COPIA]# cal
    September 2022
Su Mo Tu We Th Fr Sa
                1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30

[root@localhost COPIA]#
```

El comando cal muestra el calendario del mes actual en un calendario tradicional.

Comando date

```
[root@localhost COPIA]# date
Tue Sep 13 01:04:38 PM UTC 2022
[root@localhost COPIA]#
```

Nos despliega la fecha y hora en la que se escribe el comando.

Comando auxiliar es: man

```
MAN(1)                                Manual pager utils                                MAN(1)

NAME
    man - an interface to the system reference manuals

SYNOPSIS
    man [man options] [[section] page ...] ...
    man -k [apropos options] regexp ...
    man -K [man options] [section] term ...
    man -f [whatis options] page ...
    man -l [man options] file ...
    man -w|-W [man options] page ...

DESCRIPTION
    man is the system's manual pager. Each page argument given to man is
    normally the name of a program, utility or function. The manual page
    associated with each of these arguments is then found and displayed. A
    section, if provided, will direct man to look only in that section of
    the manual. The default action is to search in all of the available
    sections following a pre-defined order (see DEFAULTS), and to show only
    the first page found, even if page exists in several sections.

    The table below shows the section numbers of the manual followed by the
    types of pages they contain.

    1   Executable programs or shell commands
    2   System calls (functions provided by the kernel)
    3   Library calls (functions within program libraries)
    4   Special files (usually found in /dev)

Manual page man(1) line 1 (press h for help or q to quit)
```

Muestra descripciones sobre Funciones exactas, por ejemplo para saber lo que hace este comando podemos utilizar la instrucción **man man** que nos mostrará el siguiente mensaje: “man – an interface to the system reference manuals”

Conclusiones

El sistema Linux es muy útil y es una forma de ver las diferentes maneras en las que podemos acceder a las utilidades de la computadora, digamos que es una manera más técnica de hacer y ver cosas que hay dentro de la computadora. La verdad yo no tenía ni idea de que existía, pensaba que era “hackear” la máquina, pero es entrar dentro de ella, acceder a comandos y trabajar con la compu de una manera más técnica e interesante. Aprendí bastante con esta práctica y estoy seguro de que me servirá en ocasiones.

<https://github.com/3201050964/Practica-2>