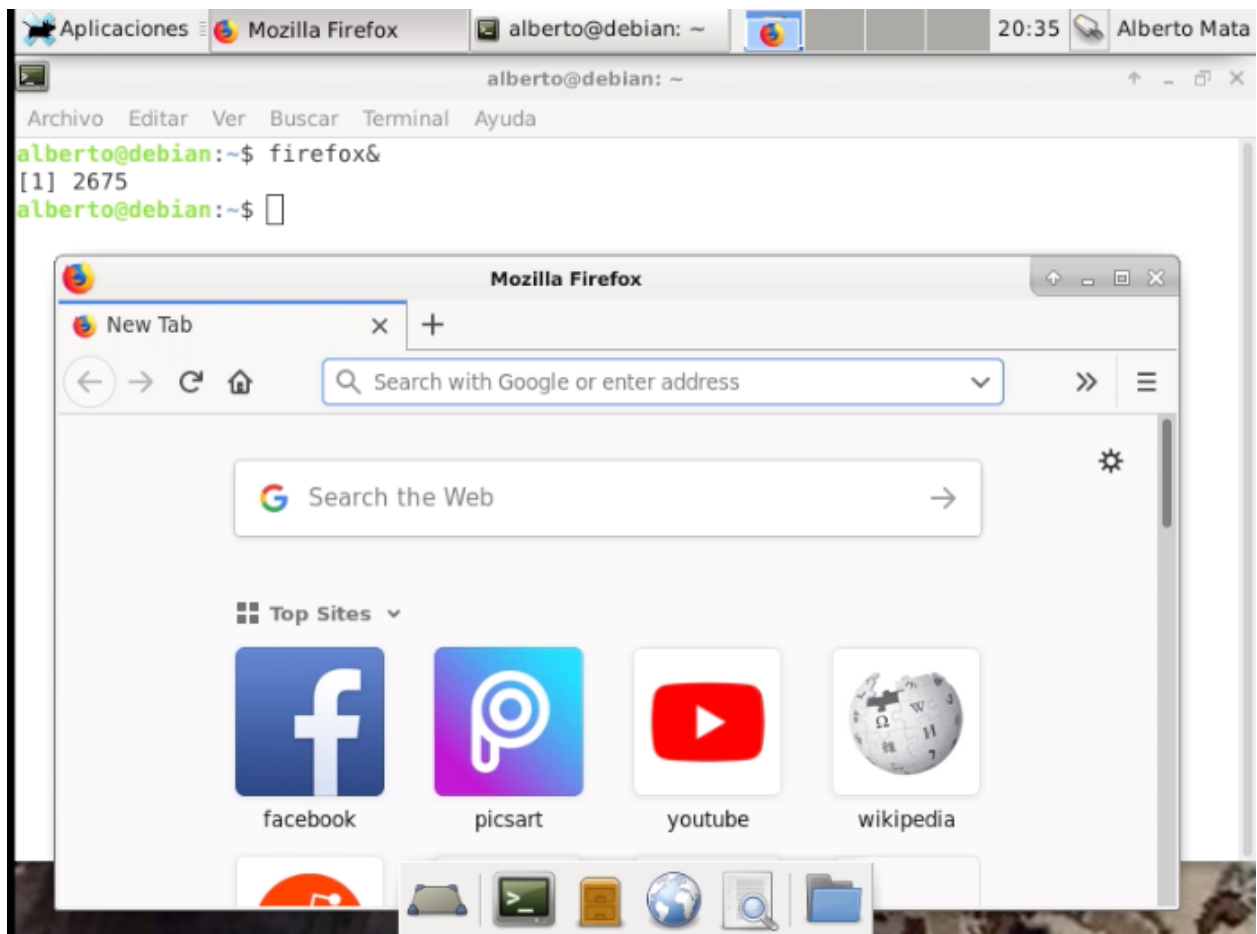


Universidad de Costa Rica  
Escuela de Ingeniería Eléctrica  
Programación Bajo Plataformas Abiertas  
MSc. Andrés Mora Zúñiga  
I Ciclo 2020  
IE-0117

## Práctica # 2:Linux: Más comandos

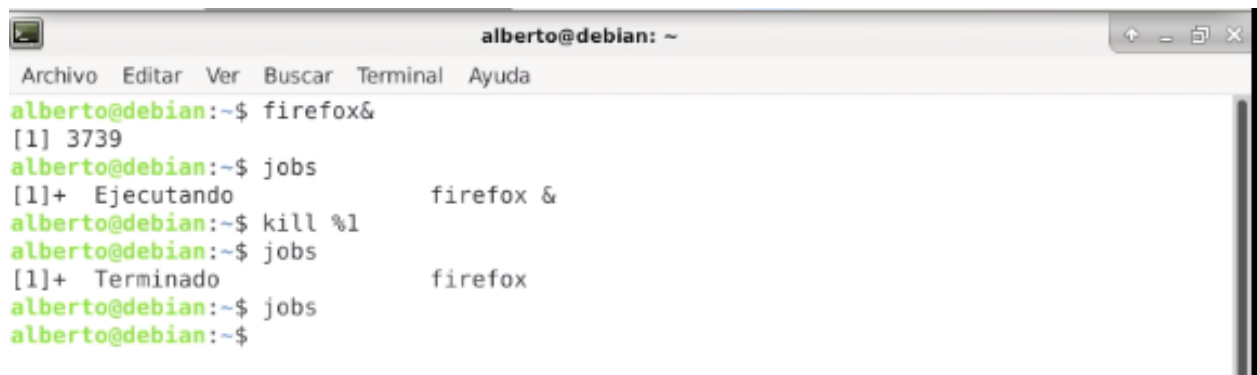
### 1. Comandos básicos de procesos

Primeramente se abre la terminal y se ejecuta el comando `firefox` para que inicie el navegador y corra inmediatamente en el background, sin bloquear la terminal activa para esto se complementa con `&`. Como se ve en la figura 1.



**Figura 1:** Comandos básicos 1.1 y 1.2

Posteriormente se listan los jobs que están corriendo actualmente en la terminal. Estos jobs lo que hacen es que muestran el programa en ejecución. Luego se ejecuta un comando para detener completamente el job que corresponde a firefox, para lo que se usa `kill % 1`, ya que se solicita utilizando sintaxis de jobs. Después de un tiempo si se vuelve a poner jobs ya no sale nada. Tal como se ve en la figura 2



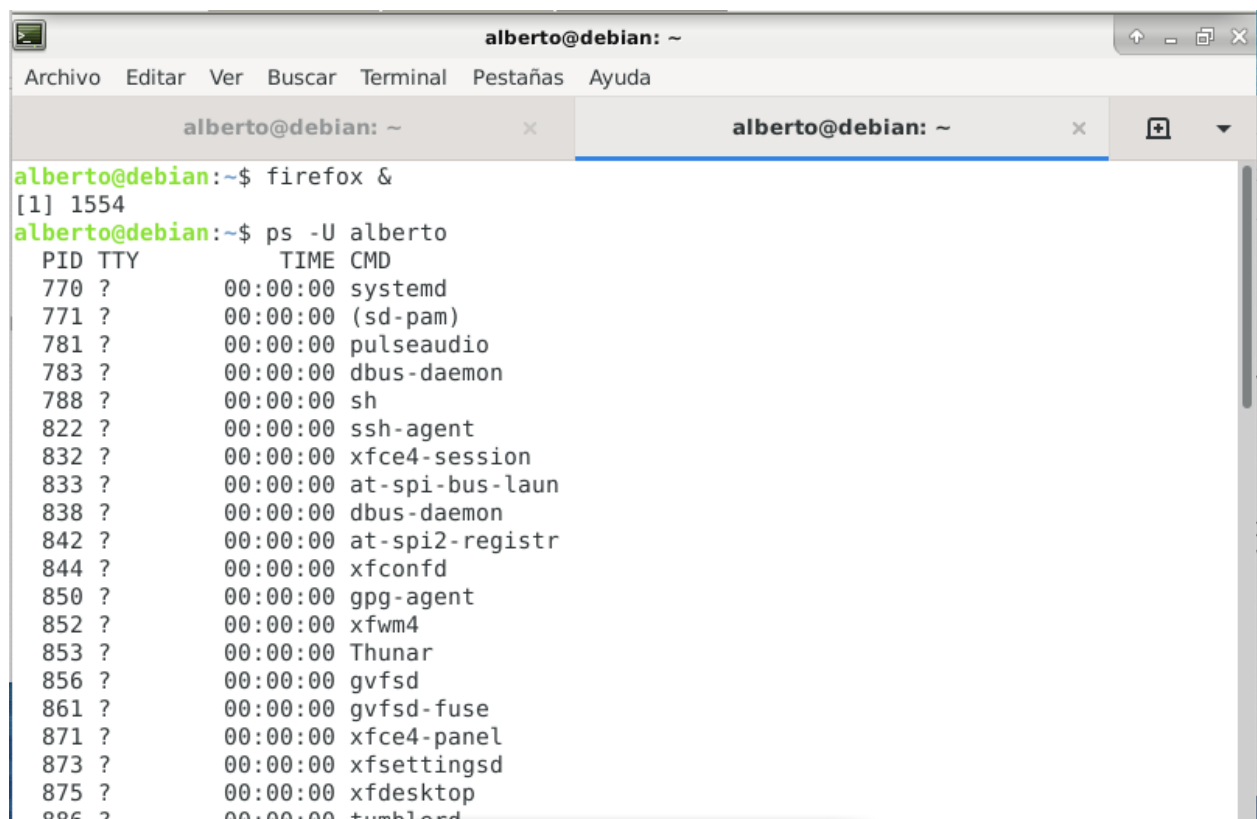
```

alberto@debian: ~
Archivo Editar Ver Buscar Terminal Ayuda
alberto@debian:~$ firefox&
[1] 3739
alberto@debian:~$ jobs
[1]+  Ejecutando                firefox &
alberto@debian:~$ kill %1
alberto@debian:~$ jobs
[1]+  Terminado                firefox
alberto@debian:~$ jobs
alberto@debian:~$

```

**Figura 2:** Comandos básicos 1.3 y 1.4

Ahora se vuelve a ejecutar el comando `firefox` que inicia el navegador y corre inmediatamente en el background, sin bloquear la terminal activa. Además se listan todos los procesos que el usuario está corriendo. Para ello se hace uso de la herramienta `ps` que lista los procesos, y se adjunta con `-U alberto` ya que esta hace que sean los procesos justamente del usuario. Como se muestra en la figura 3.



```

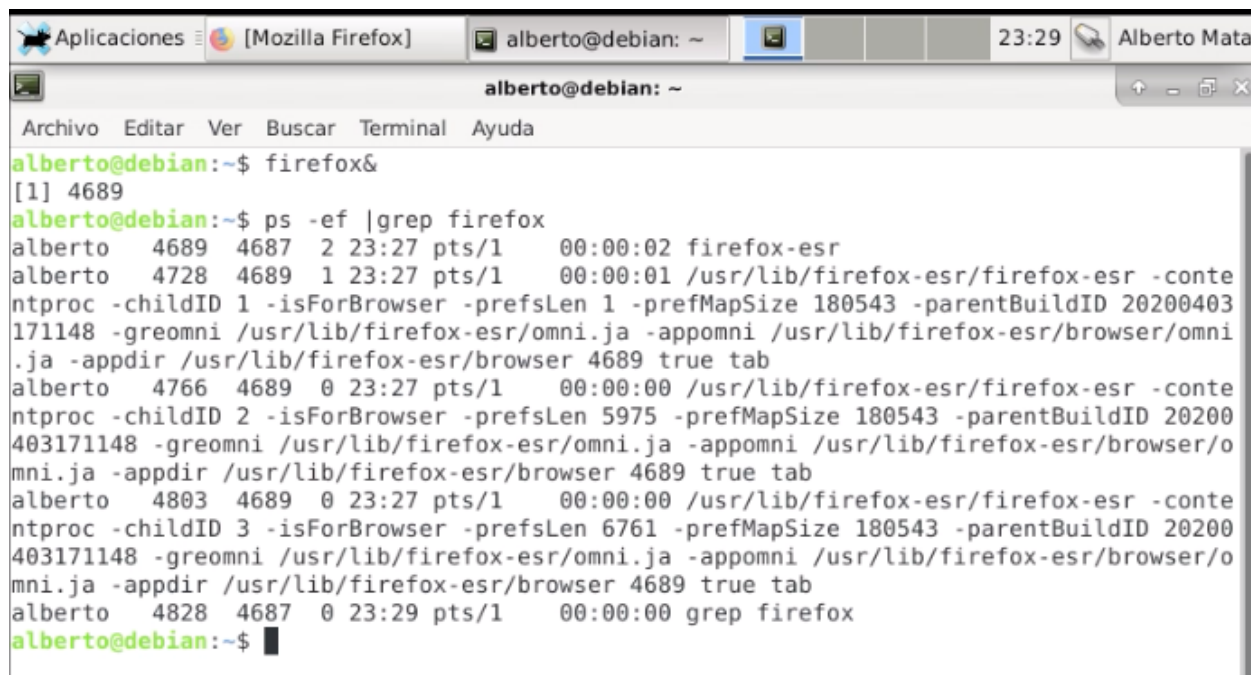
alberto@debian: ~
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
alberto@debian: ~ x alberto@debian: ~ x
alberto@debian:~$ firefox &
[1] 1554
alberto@debian:~$ ps -U alberto
PID TTY          TIME CMD
 770 ?            00:00:00 systemd
 771 ?            00:00:00 (sd-pam)
 781 ?            00:00:00 pulseaudio
 783 ?            00:00:00 dbus-daemon
 788 ?            00:00:00 sh
 822 ?            00:00:00 ssh-agent
 832 ?            00:00:00 xfce4-session
 833 ?            00:00:00 at-spi-bus-laun
 838 ?            00:00:00 dbus-daemon
 842 ?            00:00:00 at-spi2-registr
 844 ?            00:00:00 xfconfd
 850 ?            00:00:00 gpg-agent
 852 ?            00:00:00 xfwm4
 853 ?            00:00:00 Thunar
 856 ?            00:00:00 gvfsd
 861 ?            00:00:00 gvfsd-fuse
 871 ?            00:00:00 xfce4-panel
 873 ?            00:00:00 xfsettingsd
 875 ?            00:00:00 xfdesktop
 896 ?            00:00:00 tumblerd

```

**Figura 3:** Comandos básicos 1.5 y 1.6

Finalmente se identifica el pid del proceso padre de firefox. Para identificar cual es el padre, por medio del comando `grep` se muestran solo los resultados que utilicen firefox. Luego se pone atención en la tercera columna al que tenga el menor numero y este corresponde al PPID. Como se muestra en la figura 4. Cuando ya está identificado, se detiene definitivamente

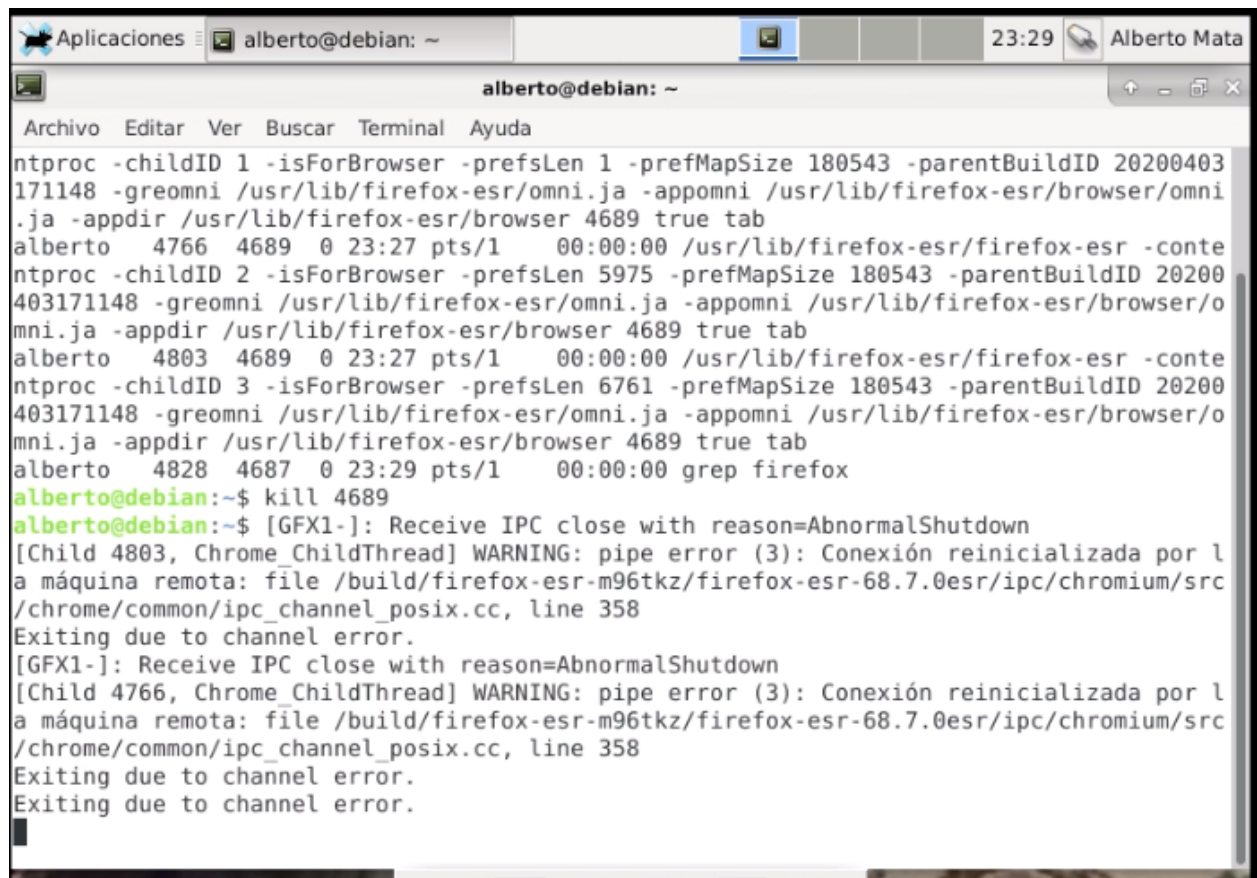
utilizando sintaxis de procesos, osea con kill 4689.Como se muestra en la figura 5.



The image shows a terminal window titled 'alberto@debian: ~' with a menu bar containing 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal output is as follows:

```
alberto@debian:~$ firefox&
[1] 4689
alberto@debian:~$ ps -ef |grep firefox
alberto  4689  4687  2 23:27 pts/1    00:00:02 firefox-esr
alberto  4728  4689  1 23:27 pts/1    00:00:01 /usr/lib/firefox-esr/firefox-esr -conte
ntproc -childID 1 -isForBrowser -prefslen 1 -prefMapSize 180543 -parentBuildID 20200403
171148 -greomni /usr/lib/firefox-esr/omni.ja -appomni /usr/lib/firefox-esr/browser/omni
.ja -appdir /usr/lib/firefox-esr/browser 4689 true tab
alberto  4766  4689  0 23:27 pts/1    00:00:00 /usr/lib/firefox-esr/firefox-esr -conte
ntproc -childID 2 -isForBrowser -prefslen 5975 -prefMapSize 180543 -parentBuildID 20200
403171148 -greomni /usr/lib/firefox-esr/omni.ja -appomni /usr/lib/firefox-esr/browser/o
mni.ja -appdir /usr/lib/firefox-esr/browser 4689 true tab
alberto  4803  4689  0 23:27 pts/1    00:00:00 /usr/lib/firefox-esr/firefox-esr -conte
ntproc -childID 3 -isForBrowser -prefslen 6761 -prefMapSize 180543 -parentBuildID 20200
403171148 -greomni /usr/lib/firefox-esr/omni.ja -appomni /usr/lib/firefox-esr/browser/o
mni.ja -appdir /usr/lib/firefox-esr/browser 4689 true tab
alberto  4828  4687  0 23:29 pts/1    00:00:00 grep firefox
alberto@debian:~$
```

Figura 4: Comandos básicos 1.7a

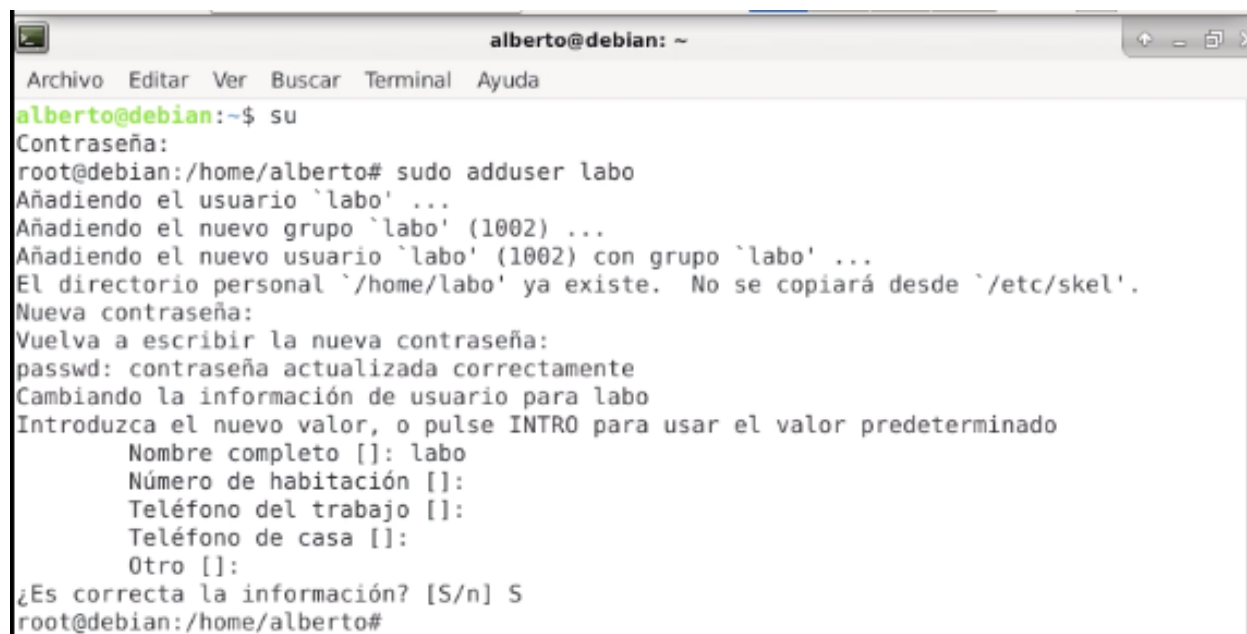


```
Aplicaciones | alberto@debian: ~ | 23:29 | Alberto Mata
alberto@debian: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
ntproc -childID 1 -isForBrowser -prefsLen 1 -prefMapSize 180543 -parentBuildID 20200403
171148 -greomni /usr/lib/firefox-esr/omni.ja -appomni /usr/lib/firefox-esr/browser/omni
.ja -appdir /usr/lib/firefox-esr/browser 4689 true tab
alberto 4766 4689 0 23:27 pts/1 00:00:00 /usr/lib/firefox-esr/firefox-esr -conte
ntproc -childID 2 -isForBrowser -prefsLen 5975 -prefMapSize 180543 -parentBuildID 20200
403171148 -greomni /usr/lib/firefox-esr/omni.ja -appomni /usr/lib/firefox-esr/browser/o
mni.ja -appdir /usr/lib/firefox-esr/browser 4689 true tab
alberto 4803 4689 0 23:27 pts/1 00:00:00 /usr/lib/firefox-esr/firefox-esr -conte
ntproc -childID 3 -isForBrowser -prefsLen 6761 -prefMapSize 180543 -parentBuildID 20200
403171148 -greomni /usr/lib/firefox-esr/omni.ja -appomni /usr/lib/firefox-esr/browser/o
mni.ja -appdir /usr/lib/firefox-esr/browser 4689 true tab
alberto 4828 4687 0 23:29 pts/1 00:00:00 grep firefox
alberto@debian:~$ kill 4689
alberto@debian:~$ [GFX1-]: Receive IPC close with reason=AbnormalShutdown
[Child 4803, Chrome_ChildThread] WARNING: pipe error (3): Conexión reiniciada por l
a máquina remota: file /build/firefox-esr-m96tkz/firefox-esr-68.7.0esr/ipc/chromium/src
/chrome/common/ipc_channel_posix.cc, line 358
Exiting due to channel error.
[GFX1-]: Receive IPC close with reason=AbnormalShutdown
[Child 4766, Chrome_ChildThread] WARNING: pipe error (3): Conexión reiniciada por l
a máquina remota: file /build/firefox-esr-m96tkz/firefox-esr-68.7.0esr/ipc/chromium/src
/chrome/common/ipc_channel_posix.cc, line 358
Exiting due to channel error.
Exiting due to channel error.
```

Figura 5: Comandos básicos 1.7b

## 2. Usuarios, grupos y permisos:

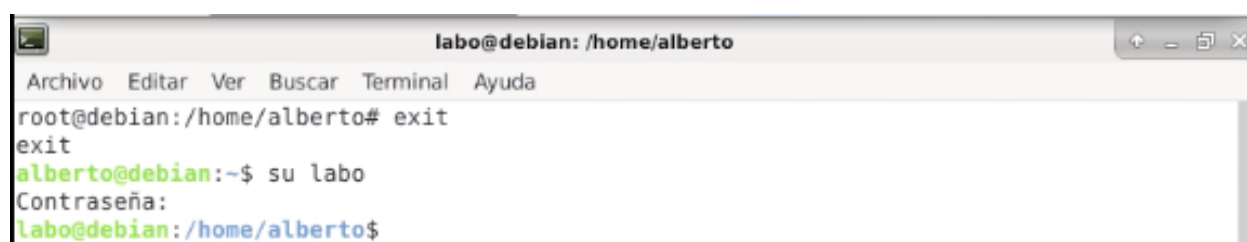
Primeramente se cambia de usuario a root mediante el comando `su`. Luego se crea un usuario nuevo llamado `labo`, con la contraseña `labo`, esto con el comando `adduser`. Como se ve en la figura 6.



```
alberto@debian: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
alberto@debian:~$ su  
Contraseña:  
root@debian:/home/alberto# sudo adduser labo  
Añadiendo el usuario 'labo' ...  
Añadiendo el nuevo grupo 'labo' (1002) ...  
Añadiendo el nuevo usuario 'labo' (1002) con grupo 'labo' ...  
El directorio personal '/home/labo' ya existe. No se copiará desde '/etc/skel'.  
Nueva contraseña:  
Vuelva a escribir la nueva contraseña:  
passwd: contraseña actualizada correctamente  
Cambiando la información de usuario para labo  
Introduzca el nuevo valor, o pulse INTRO para usar el valor predeterminado  
Nombre completo []: labo  
Número de habitación []:  
Teléfono del trabajo []:  
Teléfono de casa []:  
Otro []:  
¿Es correcta la información? [S/n] S  
root@debian:/home/alberto#
```

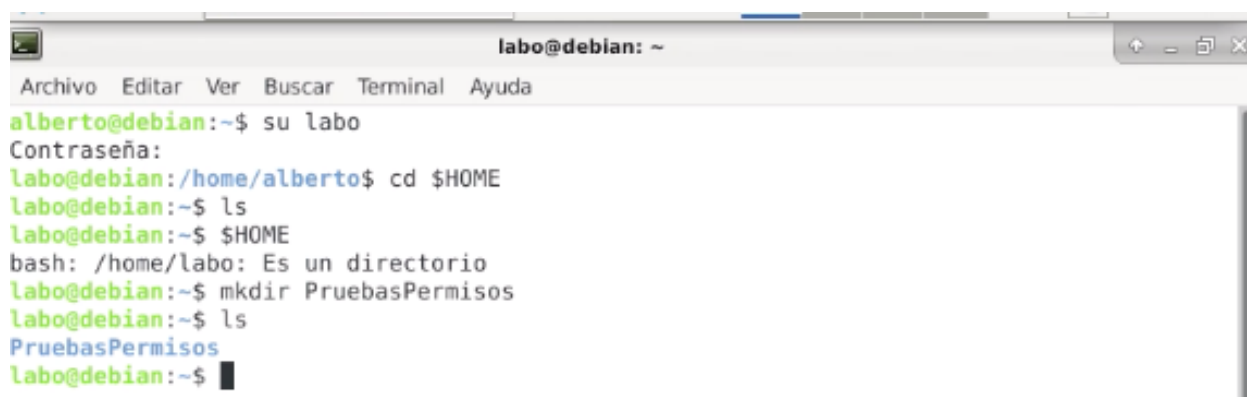
Figura 6: Usuarios, grupos y permisos 2.1 y 2.2

Posteriormente se cierra la sesión de root con el comando `exit`. Se cambia de usuario a `labo` con el comando `su labo`. Como se muestra en la figura 7. Además se crea un directorio llamado `PruebasPermisos` en el `$HOME` de este usuario mediante `mkdir`. Como se ve en la figura 8.



```
labo@debian: /home/alberto  
Archivo Editar Ver Buscar Terminal Ayuda  
root@debian:/home/alberto# exit  
exit  
alberto@debian:~$ su labo  
Contraseña:  
labo@debian:/home/alberto$
```

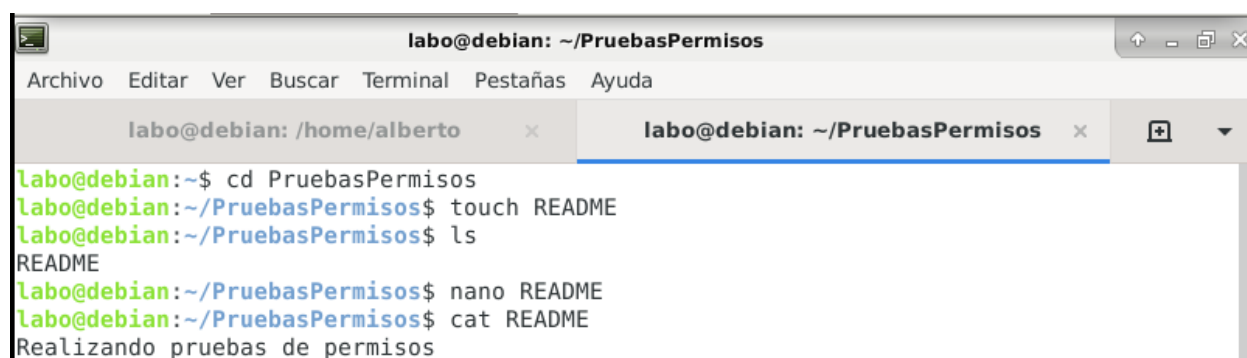
Figura 7: Usuarios, grupos y permisos 2.3a



```
labo@debian: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
alberto@debian:~$ su labo  
Contraseña:  
labo@debian:/home/alberto$ cd $HOME  
labo@debian:~$ ls  
labo@debian:~$ $HOME  
bash: /home/labo: Es un directorio  
labo@debian:~$ mkdir PruebasPermisos  
labo@debian:~$ ls  
PruebasPermisos  
labo@debian:~$
```

**Figura 8:** Usuarios, grupos y permisos 2.3b

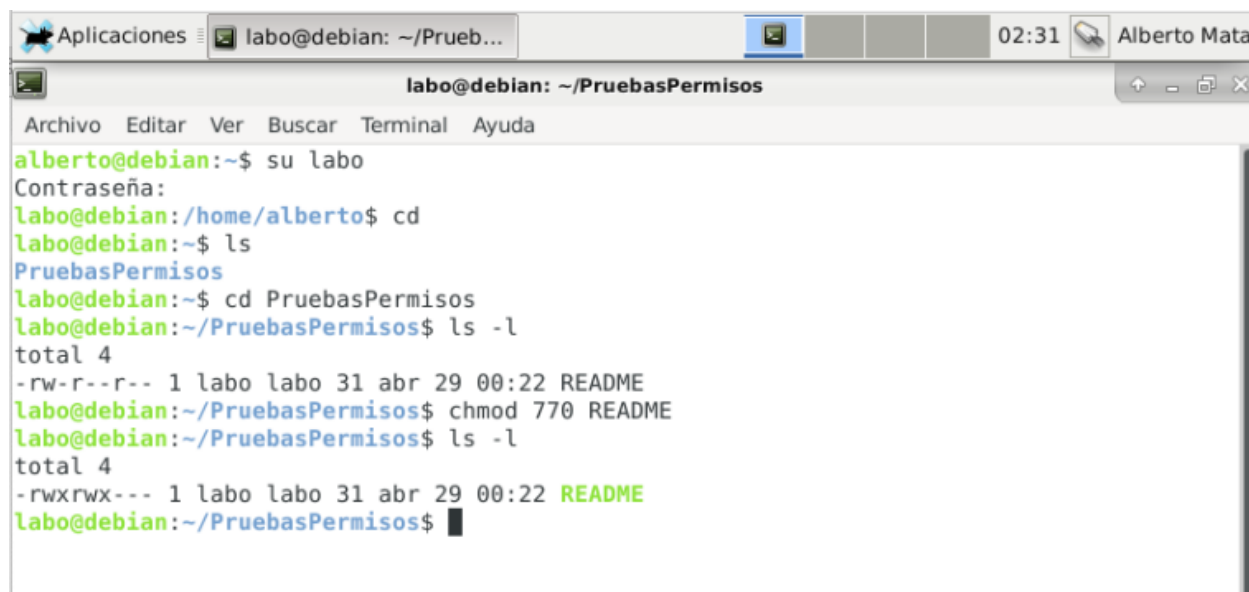
Dentro del directorio nuevo, se crea un archivo llamado README con el comando touch, que su contenido es: “Realizando pruebas de permisos”. El contenido se modifica con el comando nano. Tal como se ve en la figura 9.



```
labo@debian: ~/PruebasPermisos  
Archivo Editar Ver Buscar Terminal Pestañas Ayuda  
labo@debian: /home/alberto x labo@debian: ~/PruebasPermisos x  
labo@debian:~$ cd PruebasPermisos  
labo@debian:~/PruebasPermisos$ touch README  
labo@debian:~/PruebasPermisos$ ls  
README  
labo@debian:~/PruebasPermisos$ nano README  
labo@debian:~/PruebasPermisos$ cat README  
Realizando pruebas de permisos
```

**Figura 9:** Usuarios, grupos y permisos 2.4

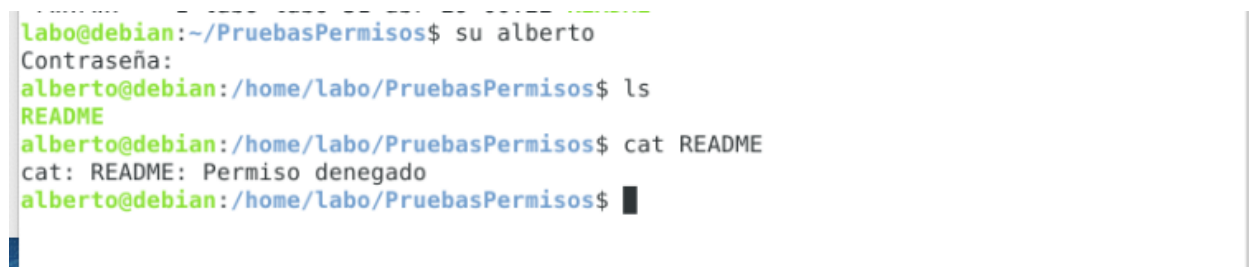
Con un solo comando, se cambian los permisos del directorio PruebasPermisos y todos los archivos que contiene, de manera tal que solamente el usuario dueño del directorio y el grupo al que pertenece, tienen permisos de lectura, escritura y ejecución. Para esto se utilizó el comando `chmod`. Como se ve en la figura 10.



```
labo@debian: ~/Prueb... 02:31 Alberto Mata
labo@debian: ~/PruebasPermisos
Archivo Editar Ver Buscar Terminal Ayuda
alberto@debian:~$ su labo
Contraseña:
labo@debian:/home/alberto$ cd
labo@debian:~$ ls
PruebasPermisos
labo@debian:~$ cd PruebasPermisos
labo@debian:~/PruebasPermisos$ ls -l
total 4
-rw-r--r-- 1 labo labo 31 abr 29 00:22 README
labo@debian:~/PruebasPermisos$ chmod 770 README
labo@debian:~/PruebasPermisos$ ls -l
total 4
-rwxrwx--- 1 labo labo 31 abr 29 00:22 README
labo@debian:~/PruebasPermisos$
```

**Figura 10:** Usuarios, grupos y permisos 2.5

Ahora se cambia al usuario original mediante su usuario. Se trata de leer el archivo README con el comando cat. Sucede lo que la figura 11 muestra. Donde se observa que el permiso esta denegado.



```
labo@debian:~/PruebasPermisos$ su alberto
Contraseña:
alberto@debian:/home/labo/PruebasPermisos$ ls
README
alberto@debian:/home/labo/PruebasPermisos$ cat README
cat: README: Permiso denegado
alberto@debian:/home/labo/PruebasPermisos$
```

**Figura 11:** Usuarios, grupos y permisos 2.6

Posteriormente se cambia de usuario a root nuevamente con su . Además se crea un grupo nuevo llamado grupolabo, donde se agregan al usuario labo y al usuario propio, este con el comando addgroup y se agregan con adduser. Tal como se observa en la figura 12.



```

alberto@debian:/home/labob$ su
Contraseña:
root@debian:/home/labob# cd .. ..
bash: cd: demasiados argumentos
root@debian:/home/labob# cd ..
root@debian:/home# cd ..
root@debian:/# sudo addgroup grupolabo
Añadiendo el grupo `grupolabo' (GID 1003) ...
Hecho.
root@debian:/# sudo adduser alberto grupolabo
Añadiendo al usuario `alberto' al grupo `grupolabo' ...
Añadiendo al usuario alberto al grupo grupolabo
Hecho.
root@debian:/# sudo adduser labo grupolabo
Añadiendo al usuario `labo' al grupo `grupolabo' ...
Añadiendo al usuario labo al grupo grupolabo
Hecho.
root@debian:/# █

```

**Figura 12:** Usuarios, grupos y permisos 2.7 y 2.8

Posteriormente se cambia recursivamente(hace que los elementos dentro del directorio también cambien de grupo) el grupo de pertenencia del directorio PruebasPermisos al grupo grupolabo. Para realizarlo recursivamente se utiliza el comando chgrp y además se complementa con -R. Se confirma ya que se vuelve a ver los permisos del archivo y esta vez en la columna de grupos dice grupolabo en vez de labo. Como se muestra en la figura 13.

```

alberto@debian:/home/labob/PruebasPermisos$ su
Contraseña:
root@debian:/home/labob/PruebasPermisos# cd ..
root@debian:/home/labob# cd ..
root@debian:/home# cd ..
root@debian:/# sudo chgrp -R grupolabo home/labob/PruebasPermisos
root@debian:/# cd home/labob/PruebasPermisos
root@debian:/home/labob/PruebasPermisos# ls -l
total 4
-rwxrwx--- 1 labo grupolabo 31 abr 29 00:22 README

```

**Figura 13:** Usuarios, grupos y permisos 2.9

Ahora se cierra la sesión del root mediante exit y se vuelve al usuario propio con su usuario. Se ejecuta el siguiente comando: `exec su -l $USER`. Como se ve en la figura 14. Dicho comando lo que hace es que utiliza el proceso actual de la terminal y lo reemplaza de manera que cambia la sesión y cierra la anterior.



```
alberto@debian:~$ su
Contraseña:
root@debian:/home/alberto# exit
exit
alberto@debian:~$ exec su -l $USER
Contraseña:
alberto@debian:~$
```

**Figura 14:** Usuarios, grupos y permisos 2.10

Finalmente se accesa nuevamente al archivo README y en esta ocasión si se logra acceder al archivo, no como en la figura 11 donde el permiso fue negado. Dicha resolución se ve en la figura 15.

```
alberto@debian:/$ cd home
alberto@debian:/home$ cd labo
alberto@debian:/home/labo$ ls
PruebasPermisos
alberto@debian:/home/labo$ cd PruebasPermisos
alberto@debian:/home/labo/PruebasPermisos$ cat README
Realizando pruebas de permisos
alberto@debian:/home/labo/PruebasPermisos$
```

**Figura 15:** Usuarios, grupos y permisos 2.11

### 3. Instalacion de paquetes desde otros repositorios

Primeramente se copia el código brindado por el link <https://www.spotify.com/cr/download/linux/>. Como se muestra en la figura 16 y 17

```

alberto@debian:~$ curl -sS https://download.spotify.com/debian/pubkey.gpg | sudo apt-key
y add-
bash: curl: orden no encontrada
[sudo] password for alberto:
Usage: apt-key [--keyring file] [command] [arguments]

Manage apt's list of trusted keys

    apt-key add <file>          - add the key contained in <file> ('-' for stdin)
    apt-key del <keyid>         - remove the key <keyid>
    apt-key export <keyid>      - output the key <keyid>
    apt-key exportall           - output all trusted keys
    apt-key update              - update keys using the keyring package
    apt-key net-update          - update keys using the network
    apt-key list                - list keys
    apt-key finger              - list fingerprints
    apt-key adv                 - pass advanced options to gpg (download key)

If no specific keyring file is given the command applies to all keyring files.
alberto@debian:~$ echo "deb http://repository.spotify.com stable non-free" | sudo tee /
etc/apt/sources.list.d/spotify.list
deb http://repository.spotify.com stable non-free
alberto@debian:~$ sudo apt-get update && apt-get install spotify-client
Obj:1 http://deb.debian.org/debian buster InRelease
Des:2 http://security.debian.org/debian-security buster/updates InRelease [65,4 kB]
Des:3 http://repository.spotify.com stable InRelease [3 316 B]

```

Figura 16: Instalación de paquetes desde otros repositorio

```

etc/apt/sources.list.d/spotify.list
deb http://repository.spotify.com stable non-free
alberto@debian:~$ sudo apt-get update && apt-get install spotify-client
Obj:1 http://deb.debian.org/debian buster InRelease
Des:2 http://security.debian.org/debian-security buster/updates InRelease [65,4 kB]
Des:3 http://repository.spotify.com stable InRelease [3 316 B]
Des:4 http://deb.debian.org/debian buster-updates InRelease [49,3 kB]
Err:3 http://repository.spotify.com stable InRelease
  Las firmas siguientes no se pudieron verificar porque su clave pública no está dispon
ible: NO_PUBKEY 4773BD5E130D1D45
Des:5 http://security.debian.org/debian-security buster/updates/main Sources [116 kB]
Des:6 http://security.debian.org/debian-security buster/updates/main amd64 Packages [19
3 kB]
Des:7 http://security.debian.org/debian-security buster/updates/main Translation-en [10
3 kB]
Leyendo lista de paquetes... Hecho
W: Error de GPG: http://repository.spotify.com stable InRelease: Las firmas siguientes
no se pudieron verificar porque su clave pública no está disponible: NO_PUBKEY 4773BD5E
130D1D45
E: El repositorio «http://repository.spotify.com stable InRelease» no está firmado.
N: No se puede actualizar de un repositorio como este de forma segura y por tanto está
deshabilitado por omisión.
N: Vea la página de manual apt-secure(8) para los detalles sobre la creación de reposi
torios y la configuración de usuarios.
alberto@debian:~$

```

Figura 17: Instalación de paquetes desde otros repositorio

Sin embargo en la figura 17 se presenta un error en el gpg. Por ende mediante investigación en internet se logra corregir el error, para hacerlo se copia el numero de 16 dígitos que aparece

en el error pgp y se introduce con el código que se muestra en la figura 18.

```
E: El repositorio «http://repository.spotify.com stable InRelease» no está firmado.
N: No se puede actualizar de un repositorio como este de forma segura y por tanto está
deshabilitado por omisión.
N: Vea la página de manual apt-secure(8) para los detalles sobre la creación de reposi
torios y la configuración de usuarios.
alberto@debian:~$ gpg --keyserver keyserver.ubuntu.com --recv 4773BD5E130D1D45
gpg: caja de claves '/home/alberto/.gnupg/pubring.kbx' creada
gpg: gpg: /home/alberto/.gnupg/trustdb.gpg: se ha creado base de datos de confianza
gpg: clave 4773BD5E130D1D45: clave pública "Spotify Public Repository Signing Key <tux@
spotify.com>" importada
gpg: Cantidad total procesada: 1
gpg: importadas: 1
alberto@debian:~$ gpg --export --armor 4773BD5E130D1D45 |sudo apt-key add -
OK
alberto@debian:~$ sudo apt-get update
Des:1 http://repository.spotify.com stable InRelease [3 316 B]
Obj:2 http://security.debian.org/debian-security buster/updates InRelease
Obj:3 http://deb.debian.org/debian buster InRelease
Obj:4 http://deb.debian.org/debian buster-updates InRelease
Des:5 http://repository.spotify.com stable/non-free amd64 Packages [1 644 B]
Descargados 4 960 B en 2s (2 655 B/s)
Leyendo lista de paquetes... Hecho
alberto@debian:~$
```

**Figura 18:** Instalación de paquetes desde otros repositorio

Finalmente con el error corregido, se vuelve a introducir el código del link y esta vez si funciona de manera correcta. Como se muestra en la figura 19 y en la figura 20 se observa el programa Spotify en ejecución.

```
alberto@debian:~$ su
Contraseña:
root@debian:/home/alberto# sudo apt-get update && apt-get install spotify-client
Des:1 http://repository.spotify.com stable InRelease [3 316 B]
Obj:2 http://deb.debian.org/debian buster InRelease
Obj:3 http://deb.debian.org/debian buster-updates InRelease
Obj:4 http://security.debian.org/debian-security buster/updates InRelease
Descargados 3 316 B en 1s (2 451 B/s)
Leyendo lista de paquetes... Hecho
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
```

**Figura 19:** Instalación de paquetes desde otros repositorio

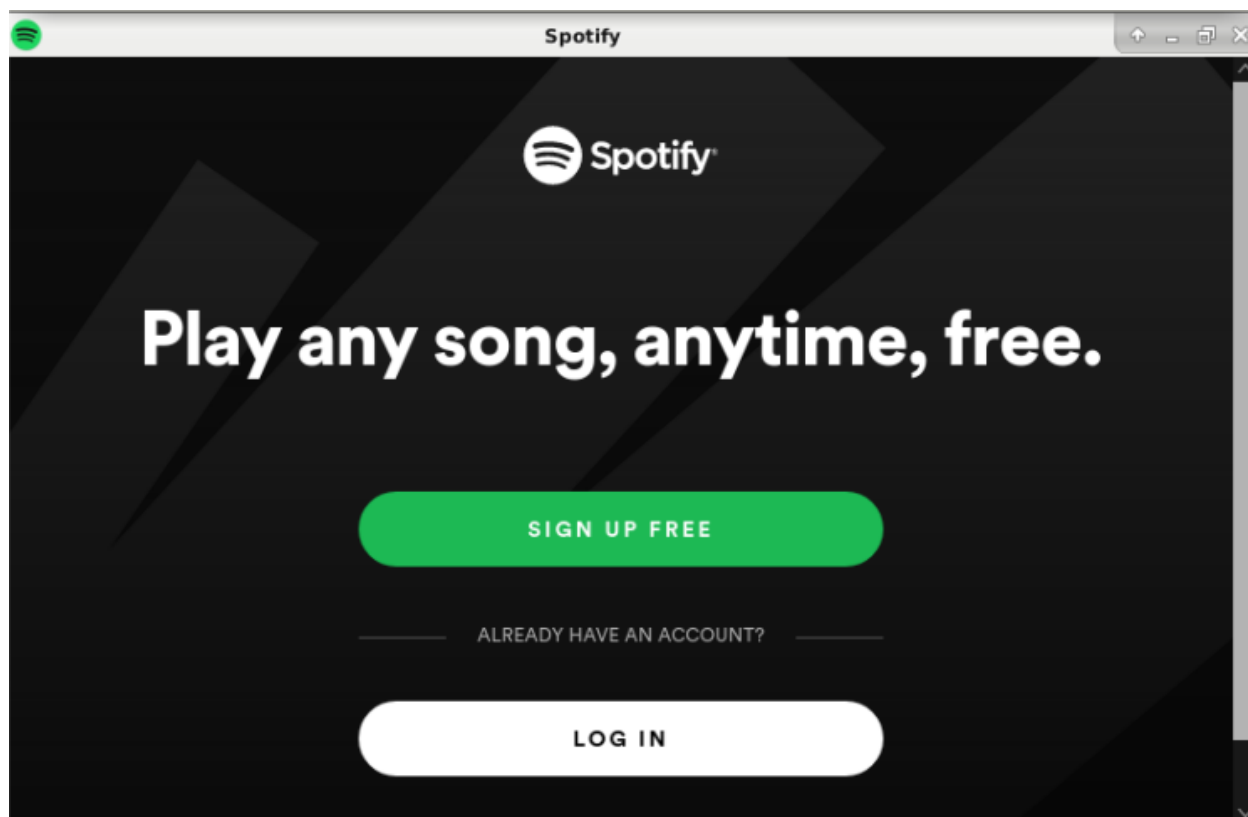


Figura 20: Spotify en ejecución

Explicación del código:

- **curl -sS https://download.spotify.com/debian/pubkey.gpg | sudo apt-key add -:** Donde se empieza con el comando curl, el cual es una abreviatura de Client URL y se utiliza como una herramienta para transferir datos. Donde se utiliza curl -sS para la verificar el URL y se une al comando | ya que este toma el stdout de un programa, el cual vendría siendo el de curl y lo redirige al stdin de key add y así logra gestionar la lista de claves que APT usa para autenticar paquetes. Esto es de suma importancia ya que los paquetes autenticados mediante estas claves se consideran de confianza.
- **echo "deb http://repository.spotify.com stable non-free" | sudo tee /etc/apt/sources.list.d/spotify.list:** Donde se empieza usando el comando echo que lo único que va a realizar es que pone a su salida el parametro "deb http://repository.spotify.com stable non-free". Sin embargo es de suma importancia ya que así se logra por medio del comando | que dicho parametro sea la entrada estándar en el comando tee y este la escribe en la salida estándar en el archivo */etc/apt/sources.list.d/spotify.list*
- **sudo apt-get update sudo apt-get install spotify-client:** Por último, con todos los procedimientos anteriores bien ejecutados, lo que falta seria instalar el programa como tal. Para ello se utiliza el comando apt-get update && sudo apt-get install el cual garantiza que primero se actualice la base de datos local y posteriormente se instala esa versión en el sistema.

## 4. Instalacion de programas desde codigo fuente

Primeramente se descarga el fichero comprimido del siguiente enlace: <https://ffmpeg.org/releases/ffmpeg-4.2.1.tar.bz2>. Posteriormente se descomprima el fichero utilizando el comando tar, ya que este comando en Linux es uno de los más usados para comprimir y descomprimir archivos. Como se puede observar en la figura 21. Después de llamar al comando tar se pone -xjvf, donde la x indica a tar que descomprima el fichero.tar., la y creará un archivo tar basado en Bzip2, la v indica a tar que muestre lo que va desempaquetando y la f indica a tar que el siguiente argumento es el nombre del fichero a desempaquetar.

```
alberto@debian:~$ ls
alberto  Documentos  Imágenes  pcinfo      Público
Descargas Escritorio  Música    Plantillas  Vídeos
alberto@debian:~$ cd Descargas
alberto@debian:~/Descargas$ ls
ffmpeg-4.2.1.tar.bz2  Labo2_Linux.pdf
alberto@debian:~/Descargas$ tar -xjvf ffmpeg-4.2.1.tar.bz2
ffmpeg-4.2.1/
ffmpeg-4.2.1/libswscale/
ffmpeg-4.2.1/libswscale/x86/
ffmpeg-4.2.1/libswscale/x86/yuv2rgb.c
ffmpeg-4.2.1/libswscale/x86/w64xmmtest.c
ffmpeg-4.2.1/libswscale/x86/rgb_2_rgb.asm
```

**Figura 21:** Instalación de programas desde código fuente 4.1,4.2 y 4.3

Posteriormente se instalan todas las dependencias. Cabe mencionar que ciertos paquetes no se localizaban como el checksintall y libfaac-dev como se muestra en la figura 22.

```
alberto@debian:~/Descargas$ sudo apt-get install build-essential git-core checkinstall
yasm texi2html libfaac-dev libjack-jackd2-dev libmp3lame-dev libopencore-amrnb-dev libo
pencore-amrwb-dev libSDL1.2-dev libtheora-dev libva-dev libvdpau-dev libvorbis-dev libx
11-dev libxfixes-dev libxvidcore-dev zlib1g-dev
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Nota, seleccionando «git» en lugar de «git-core»
E: No se ha podido localizar el paquete checkinstall
E: No se ha podido localizar el paquete libfaac-dev
```

**Figura 22:** Instalación de programas desde código fuente 4.4a

Debido que no se logran localizar esos paquetes primeramente se accede al archivo de source.list donde se agrega la última línea, que sería la que está subrayada en la figura 23.



```

alberto@debian:~/Descargas$ nano /etc/apt/sources.list
alberto@debian:~/Descargas$ su
Contraseña:
root@debian:/home/alberto/Descargas# nano /etc/apt/sources.list
root@debian:/home/alberto/Descargas# cat /etc/apt/sources.list

#
# deb cdrom:[Debian GNU/Linux 10.3.0 _Buster_ - Official amd64 NETINST 20200208-12:07]/
# buster main

#deb cdrom:[Debian GNU/Linux 10.3.0 _Buster_ - Official amd64 NETINST 20200208-12:07]/
# buster main

deb http://deb.debian.org/debian/ buster main
deb-src http://deb.debian.org/debian/ buster main

deb http://security.debian.org/debian-security buster/updates main
deb-src http://security.debian.org/debian-security buster/updates main

# buster-updates, previously known as 'volatile'
deb http://deb.debian.org/debian/ buster-updates main
deb-src http://deb.debian.org/debian/ buster-updates main

deb http://security.debian.org/debian-security buster/updates main
deb-src http://security.debian.org/debian-security buster/updates main

# buster-updates, previously known as 'volatile'
deb http://deb.debian.org/debian/ buster-updates main
deb-src http://deb.debian.org/debian/ buster-updates main

# This system was installed using small removable media
# (e.g. netinst, live or single CD). The matching "deb cdrom"
# entries were disabled at the end of the installation process.
# For information about how to configure apt package sources,
# see the sources.list(5) manual.
deb http://http.kali.org/kali kali-rolling main non-free contrib

```

**Figura 23:** Instalación de programas desde código fuente 4.4b

Una vez añadido eso al source.list ya se pueden instalar los paquetes que estaban haciendo falta. Como se ven en las figuras 24 y 25.

```

root@debian:/home/alberto/Descargas# sudo apt-get update && get install checkinstall
Des:1 http://repository.spotify.com stable InRelease [3 316 B]
Obj:2 http://deb.debian.org/debian buster InRelease
Des:3 http://deb.debian.org/debian buster-updates InRelease [49,3 kB]
Obj:4 http://security.debian.org/debian-security buster/updates InRelease
Des:5 http://kali.download/kali kali-rolling InRelease [30,5 kB]
Des:6 http://kali.download/kali kali-rolling/main amd64 Packages [16,5 MB]
Des:7 http://kali.download/kali kali-rolling/non-free amd64 Packages [195 kB]
Des:8 http://kali.download/kali kali-rolling/contrib amd64 Packages [98,2 kB]
Descargados 16,8 MB en 10s (1 633 kB/s)
Leyendo lista de paquetes... Hecho

```

**Figura 24:** Instalación de programas desde código fuente 4.4c



```
root@debian:/home/alberto/Descargas# sudo apt-get update && get install checkinstall
Des:1 http://repository.spotify.com stable InRelease [3 316 B]
Obj:2 http://deb.debian.org/debian buster InRelease
Des:3 http://deb.debian.org/debian buster-updates InRelease [49,3 kB]
Obj:4 http://security.debian.org/debian-security buster/updates InRelease
Des:5 http://kali.download/kali kali-rolling InRelease [30,5 kB]
Des:6 http://kali.download/kali kali-rolling/main amd64 Packages [16,5 MB]
Des:7 http://kali.download/kali kali-rolling/non-free amd64 Packages [195 kB]
Des:8 http://kali.download/kali kali-rolling/contrib amd64 Packages [98,2 kB]
Descargados 16,8 MB en 10s (1 633 kB/s)
Leyendo lista de paquetes... Hecho
```

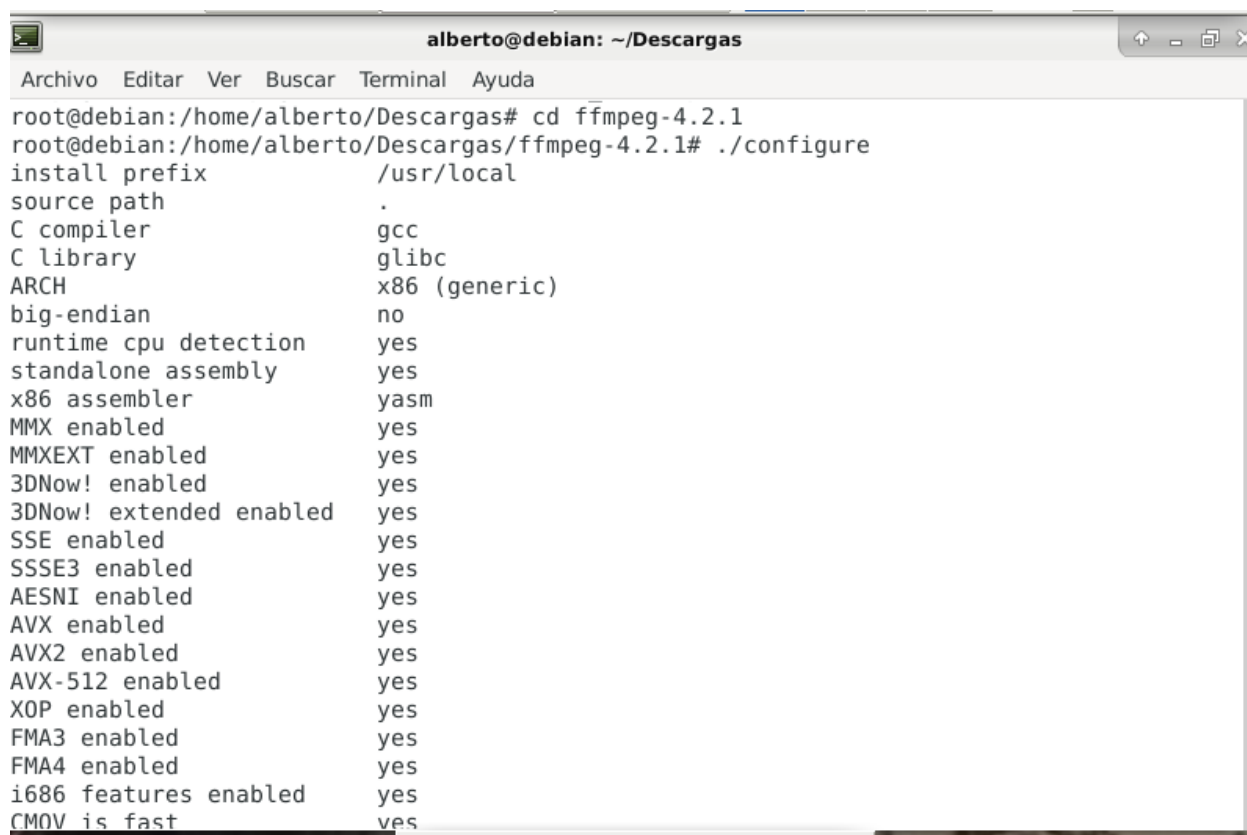
**Figura 25:** Instalación de programas desde código fuente 4.4c

Ahora se repite el código de la figura 22. En vez de los resultados anteriores, en este caso si se logran descargar todas las dependencias como se ve en la figura 26. Para ello se utiliza el comando `apt-get install` que como su nombre lo intuye, se utiliza para instalar las dependencias necesarias para la instalación del programa.

```
root@debian:/home/alberto/Descargas# sudo apt-get install build-essential git-core chec
kinstall yasm texi2html libfaac-dev libjack-jackd2-dev libmp3lame-dev libopencore-amrnb
-dev libopencore-amrwb-dev libsdl1.2-dev libtheora-dev libva-dev libvdpau-dev libvorbis
-dev libx11-dev libxfixes-dev libxvidcore-dev zlib1g-dev
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Nota. seleccionando «git» en lugar de «git-core»
```

**Figura 26:** Instalación de programas desde código fuente 4.4e

Posteriormente se procede a la instalación que consiste en la figura 27, 28, 29. Cabe mencionar que entre tanto código no se logró encontrar la captura del `make`, sin embargo se volvió a poner y como ya se había desenlazado en la figura 28 por eso después del `make` no pasa nada.



```

alberto@debian: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda
root@debian:/home/alberto/Descargas# cd ffmpeg-4.2.1
root@debian:/home/alberto/Descargas/ffmpeg-4.2.1# ./configure
install prefix                /usr/local
source path                   .
C compiler                    gcc
C library                     glibc
ARCH                          x86 (generic)
big-endian                    no
runtime cpu detection         yes
standalone assembly          yes
x86 assembler                 yasm
MMX enabled                   yes
MMXEXT enabled                yes
3DNow! enabled                yes
3DNow! extended enabled      yes
SSE enabled                   yes
SSSE3 enabled                 yes
AESNI enabled                 yes
AVX enabled                   yes
AVX2 enabled                  yes
AVX-512 enabled               yes
XOP enabled                   yes
FMA3 enabled                  yes
FMA4 enabled                  yes
i686 features enabled         yes
CMOV is fast                  yes

```

Figura 27: Instalación de programas desde código fuente 4.5 a y b

```

In file included from fftools/ffprobe.c:31:
./libavformat/avformat.h:885:21: note: declared here
  885 |     AVCodecContext *codec;
      |     ~~~~~
LD      ffprobe_g
STRIP   ffprobe
root@debian:/home/alberto/Descargas/ffmpeg-4.2.1# make
root@debian:/home/alberto/Descargas/ffmpeg-4.2.1#

```

Figura 28: Instalación de programas desde código fuente 4.5c

```

root@debian:/home/alberto/Descargas/ffmpeg-4.2.1# sudo make install
INSTALL doc/ffmpeg.html
INSTALL doc/ffprobe.html
INSTALL doc/ffmpeg-all.html
INSTALL doc/ffprobe-all.html
INSTALL doc/ffmpeg-utils.html
INSTALL doc/ffmpeg-scaler.html

```

Figura 29: Instalación de programas desde código fuente 4.5d

Finalmente se ejecuta el comando `ffmpeg` como se ve en las figuras 30 y 31. En la figura 31 se introdujo `man ffmpeg`.

```

root@debian:/home/alberto/Descargas/ffmpeg-4.2.1# ffmpeg
ffmpeg version 4.2.1 Copyright (c) 2000-2019 the FFmpeg developers
  built with gcc 9 (Debian 9.3.0-10)
  configuration:
    libavutil      56. 31.100 / 56. 31.100
    libavcodec      58. 54.100 / 58. 54.100
    libavformat      58. 29.100 / 58. 29.100
    libavdevice      58.  8.100 / 58.  8.100
    libavfilter       7. 57.100 /  7. 57.100
    libswscale       5.  5.100 /  5.  5.100
    libswresample    3.  5.100 /  3.  5.100
Hyper fast Audio and Video encoder
usage: ffmpeg [options] [[infile options] -i infile]... {[outfile options] outfile}...

Use -h to get full help or, even better, run 'man ffmpeg'
root@debian:/home/alberto/Descargas/ffmpeg-4.2.1#

```

Captura de Pantalla 2020-04-30 a la(s) 2.03.40 p

Figura 30: Instalación de programas desde código fuente 4.6a

```

alberto@debian: ~/Descargas
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
FFMPEG(1) FFMPEG(1)

NAME
  ffmpeg - ffmpeg video converter

SYNOPSIS
  ffmpeg [global_options] {[input_file_options] -i input_url} ...
  {[output_file_options] output_url} ...

DESCRIPTION
  ffmpeg is a very fast video and audio converter that can also grab from a
  live audio/video source. It can also convert between arbitrary sample rates
  and resize video on the fly with a high quality polyphase filter.

  ffmpeg reads from an arbitrary number of input "files" (which can be regular
  files, pipes, network streams, grabbing devices, etc.), specified by the "-i"
  option, and writes to an arbitrary number of output "files", which are
  specified by a plain output url. Anything found on the command line which
  cannot be interpreted as an option is considered to be an output url.

  Each input or output url can, in principle, contain any number of streams of
  different types (video/audio/subtitle/attachment/data). The allowed number
  and/or types of streams may be limited by the container format. Selecting
  which streams from which inputs will go into which output is either done
  Manual page ffmpeg(1) line 1 (press h for help or q to quit)

```

Figura 31: Instalación de programas desde código fuente 4.6b

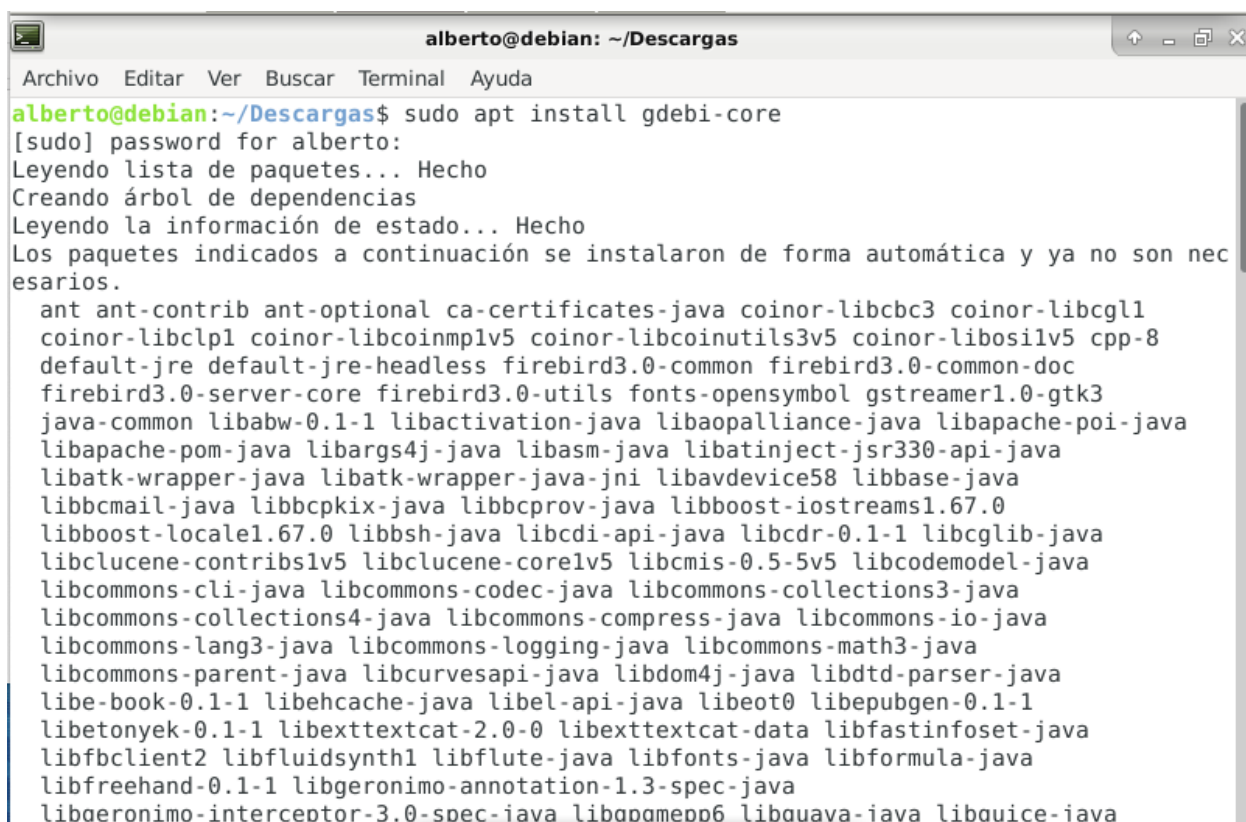
## 5. Instalacion de programas desde un .deb

Cuando se selecciona el link <https://go.microsoft.com/fwlink/?LinkID=760868> , sale un error que falta `/bin/sh`. Por ende se introduce eso en la terminal y se abre desde ahí firefox y luego se copia el link para poder descargarlo. Como se muestra en la figura 32.

```
alberto@debian:~$ </bin/sh>
bash: </bin/sh>: No existe el fichero o el directorio
alberto@debian:~$ /bin/sh
$ firefox
Sandbox: seccomp sandbox violation: pid 1662, tid 1662, syscall 315, args 1662 14023744
2078912 56 0 140237441741440 140237442078912.
Sandbox: seccomp sandbox violation: pid 1703, tid 1703, syscall 315, args 1703 13982551
1036672 56 0 139825511232608 139825511036672.
Sandbox: seccomp sandbox violation: pid 1729, tid 1729, syscall 315, args 1729 13980006
1126976 56 0 139800061270048 139800061126976.
Sandbox: seccomp sandbox violation: pid 1776, tid 1776, syscall 315, args 1776 13997544
0660672 56 0 139975442541440 139975440660672.
```

**Figura 32:** Instalación de programas desde código fuente 5.1

Posteriormente se instala desde repositorios gdebi como en la figura 33.



```
alberto@debian:~/Descargas$ sudo apt install gdebi-core
[sudo] password for alberto:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son nec
esarios.
ant ant-contrib ant-optional ca-certificates-java coinor-libcbc3 coinor-libcgl1
coinor-libclp1 coinor-libcoinmplv5 coinor-libcoinutils3v5 coinor-libosilv5 cpp-8
default-jre default-jre-headless firebird3.0-common firebird3.0-common-doc
firebird3.0-server-core firebird3.0-utils fonts-opensymbol gstreamer1.0-gtk3
java-common libabw-0.1-1 libactivation-java libaopalliance-java libapache-poi-java
libapache-pom-java libargs4j-java libasm-java libatinject-jsr330-api-java
libatk-wrapper-java libatk-wrapper-java-jni libavdevice58 libbase-java
libbcmail-java libbcprov-java libboost-iostreams1.67.0
libboost-locale1.67.0 libbsh-java libcdi-api-java libcdr-0.1-1 libcglib-java
libclucene-contribslv5 libclucene-corelv5 libcmis-0.5-5v5 libcodemodel-java
libcommons-cli-java libcommons-codec-java libcommons-collections3-java
libcommons-collections4-java libcommons-compress-java libcommons-io-java
libcommons-lang3-java libcommons-logging-java libcommons-math3-java
libcommons-parent-java libcurvesapi-java libdom4j-java libdtd-parser-java
libe-book-0.1-1 libehcache-java libel-api-java libeot0 libepubgen-0.1-1
libetonyek-0.1-1 libexttextcat-2.0-0 libexttextcat-data libfastinfoset-java
libfbclient2 libfluidsynth1 libflute-java libfonts-java libformula-java
libfreehand-0.1-1 libgeronimo-annotation-1.3-spec-java
libgeronimo-interceptor-3.0-spec-iaa libgamedb6 libguava-java libguice-iaa
```

**Figura 33:** Instalación de programas desde código fuente 5.2

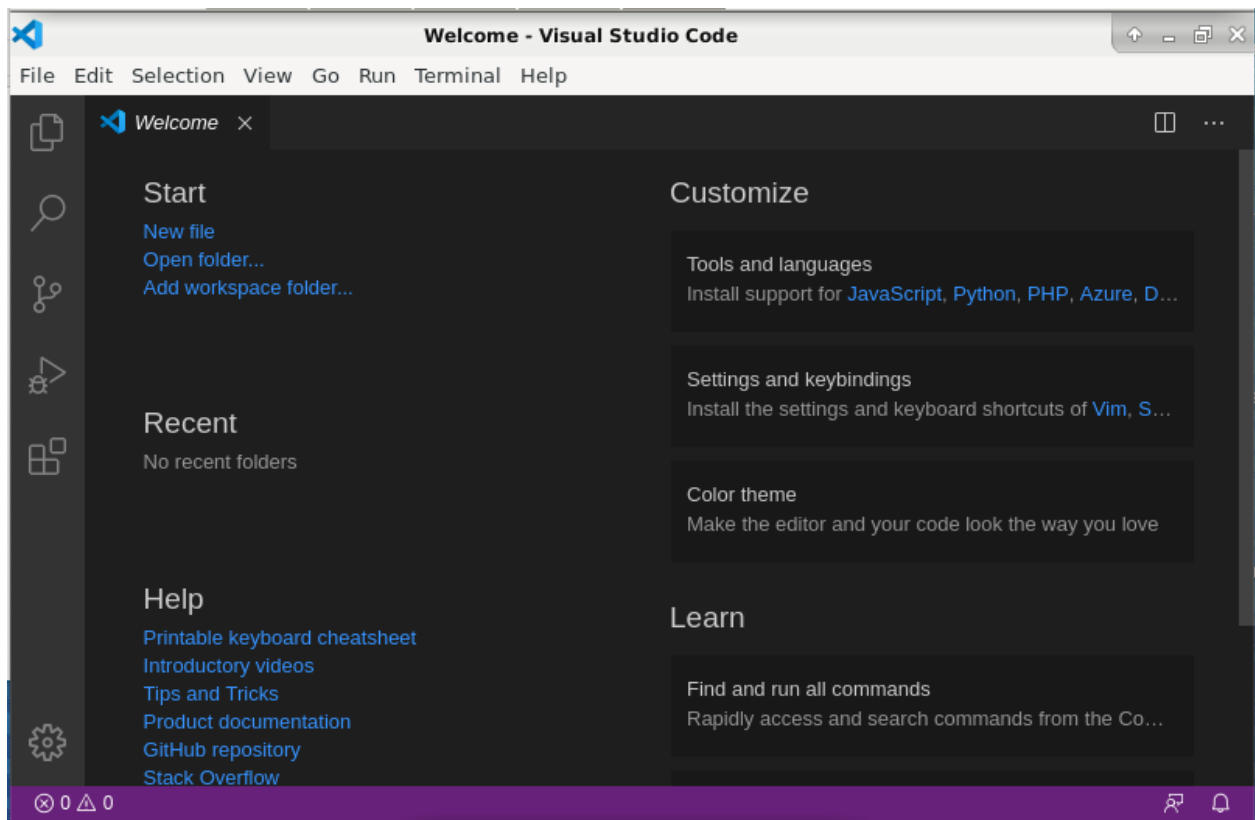
Con el comando gdebi instalado ya se puede instalar el archivo descargado de la figura 32. De la manera como lo muestra la figura 34

```
Procesando disparadores para man-db (2.8.5-2) ...
alberto@debian:~/Descargas$ ls
code_1.44.2-1587059832_amd64.deb  Labo2_Linux.pdf
alberto@debian:~/Descargas$ sudo gdebi code_1.44.2-1587059832_amd64.deb
Reading package lists... Done
Building dependency tree
Reading state information... Done
Reading state information... Done

Code editing. Redefined.
Visual Studio Code is a new choice of tool that combines the simplicity of a code edit
or with what developers need for the core edit-build-debug cycle. See https://code.visu
alstudio.com/docs/setup/linux for installation instructions and FAQ.
¿Quiere instalar el paquete de software? [s/N]:S
/usr/bin/gdebi:113: FutureWarning: Possible nested set at position 1
  c = findall("[{}()\\S+)/\\S+[]]", msg)[0].lower()
Seleccionando el paquete code previamente no seleccionado.
(Leyendo la base de datos ... 153405 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar code_1.44.2-1587059832_amd64.deb ...
Desempaquetando code (1.44.2-1587059832) ...
Configurando code (1.44.2-1587059832) ...
Procesando disparadores para gnome-menus (3.31.4-3) ...
Procesando disparadores para desktop-file-utils (0.23-4) ...
Procesando disparadores para mime-support (3.62) ...
alberto@debian:~/Descargas$
```

**Figura 34:** Instalación de programas desde código fuente 5.3

Finalmente se introduce code en la terminal y se abre el programa que se muestra en la figura 35.



**Figura 35:** Instalación de programas desde código fuente 5.4