
TRABAJO PRÁCTICO

COMPUTACIÓN APLICADA

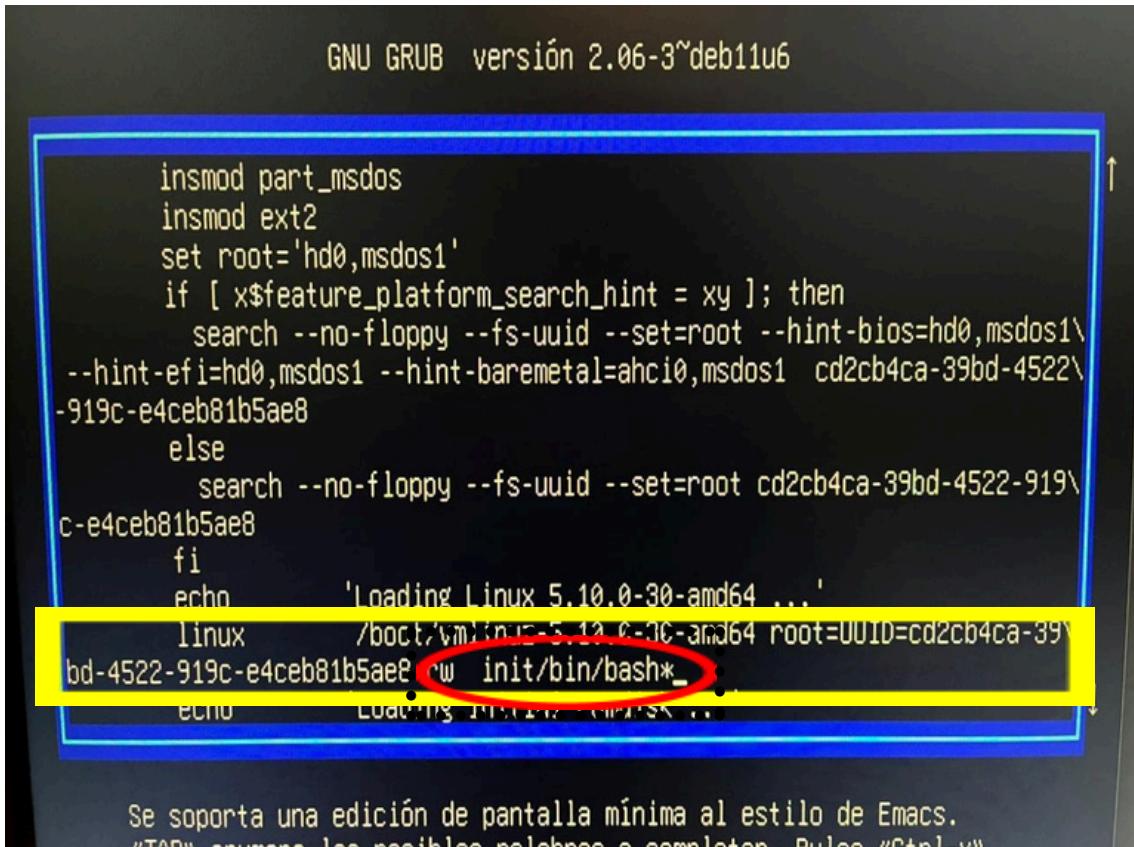


Índice:

1) Configuración del entorno.....	2
2) Servicios.....	6
Conexión SSH:.....	6
Web:.....	9
Base de Datos.....	11
3) Redes:.....	15
4) Almacenamiento.....	17
5) Back Up.....	24 6)
Diagrama Topográfico.....	30

1) Configuración del entorno

Al iniciar la máquina virtual por primera vez, accedimos al menú del GRUB y presionamos la tecla 'e' para editar las opciones de arranque. Dentro de la línea que comenzaba con 'linux', agregamos al final el texto '**rw init=/bin/bash**'. Esto nos permitió iniciar el sistema directamente en una terminal con acceso de root.

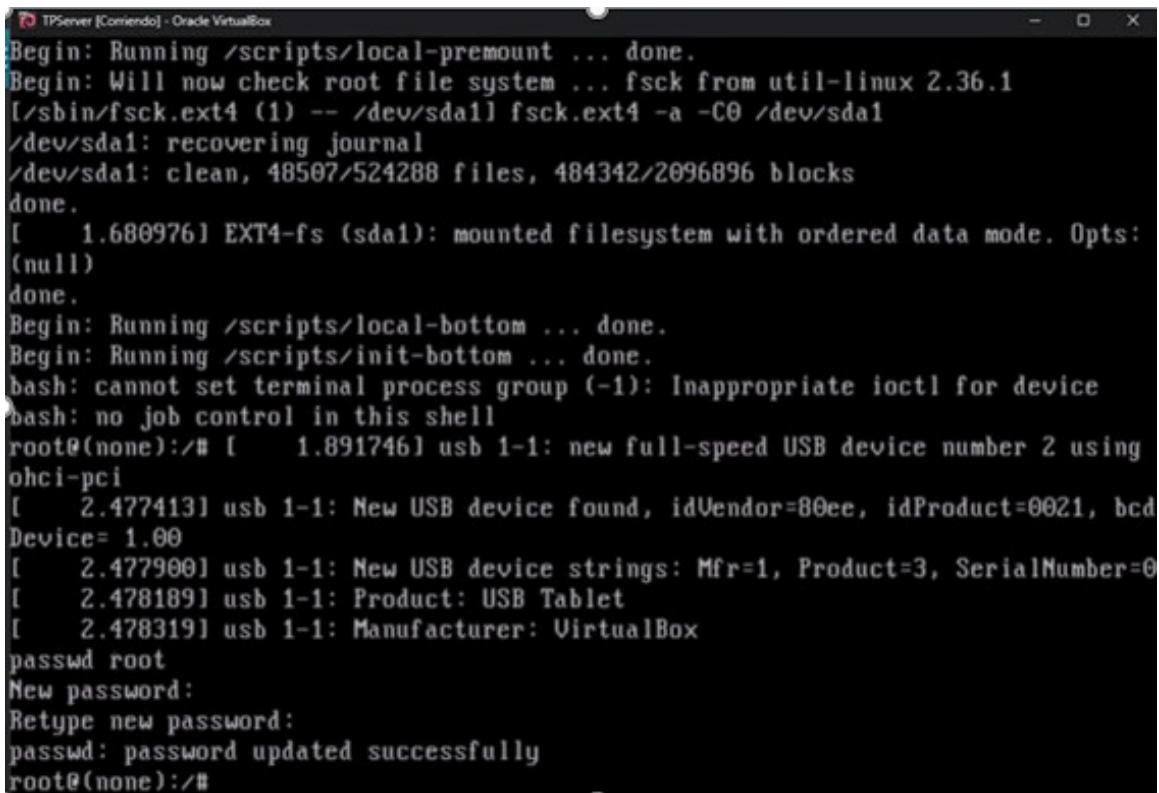


The screenshot shows the GRUB 2.06-3^deb11u6 menu editor. A yellow box highlights the 'linux' line of the configuration file. The line contains the parameter **rw init=/bin/bash**, which is circled in red. The text in the yellow box is as follows:

```
insmod part_msdos
insmod ext2
set root='hd0,msdos1'
if [ $feature_platform_search_hint = xy ]; then
    search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1\
--hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 cd2cb4ca-39bd-4522\
-919c-e4ceb81b5ae8
else
    search --no-floppy --fs-uuid --set=root cd2cb4ca-39bd-4522-919\
c-e4ceb81b5ae8
fi
echo      'Loading Linux 5.10.0-30-amd64 ...'
linux      /boot/vmlinuz-5.10.0-30-amd64 root=UUID=cd2cb4ca-39\
bd-4522-919c-e4ceb81b5ae8 rw init=/bin/bash*
echo      'Loading initial ramdisk ...'
```

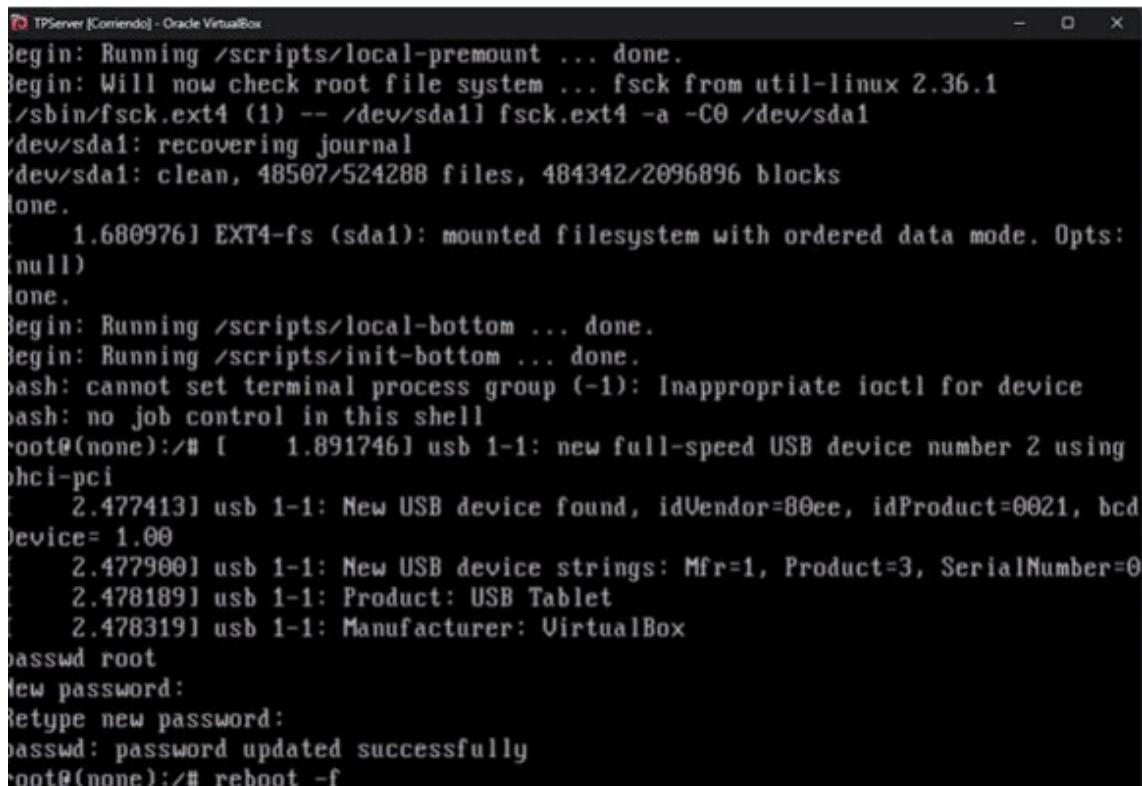
Below the highlighted line, a message reads: "Se soporta una edición de pantalla mínima al estilo de Emacs. «TAB» enumera las posibles palabras a completar. Pulse «Ctrl-X»".

Luego, presionamos F10 para continuar con el arranque usando esa configuración temporal.



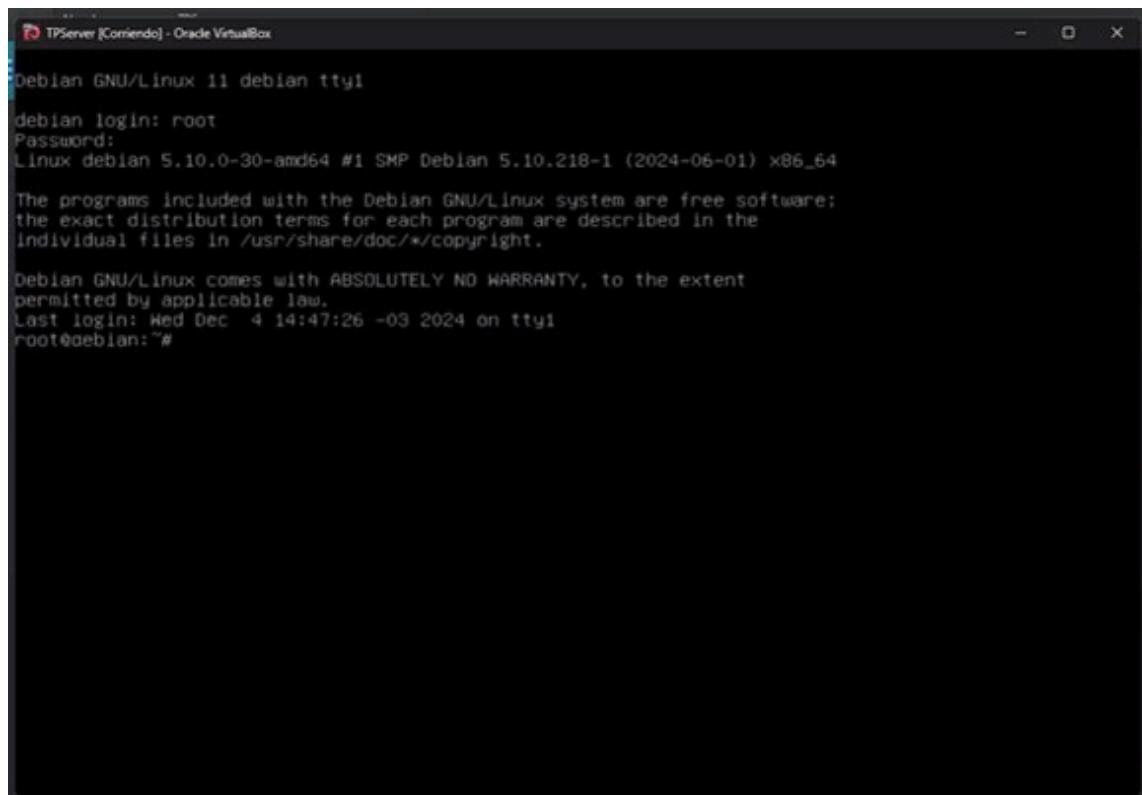
```
TPServer [Corriendo] - Oracle VirtualBox
Begin: Running /scripts/local-premount ... done.
Begin: Will now check root file system ... fsck from util-linux 2.36.1
[/sbin/fsck.ext4 (1) -- /dev/sda1] fsck.ext4 -a -C0 /dev/sda1
/dev/sda1: recovering journal
/dev/sda1: clean, 48507/524288 files, 484342/2096896 blocks
done.
[ 1.680976] EXT4-fs (sda1): mounted filesystem with ordered data mode. Opts:
(null)
done.
Begin: Running /scripts/local-bottom ... done.
Begin: Running /scripts/init-bottom ... done.
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
root@none:~# [ 1.891746] usb 1-1: new full-speed USB device number 2 using
ohci-pci
[ 2.477413] usb 1-1: New USB device found, idVendor=80ee, idProduct=0021, bcd
Device= 1.00
[ 2.477900] usb 1-1: New USB device strings: Mfr=1, Product=3, SerialNumber=0
[ 2.478189] usb 1-1: Product: USB Tablet
[ 2.478319] usb 1-1: Manufacturer: VirtualBox
passwd root
New password:
Retype new password:
passwd: password updated successfully
root@none:~#
```

Una vez iniciado el sistema con acceso directo como root, ejecutamos el comando 'mount -o remount rw /' para montar el sistema en modo escritura. Después usamos '**passwd root**' para establecer una nueva contraseña. En este caso, la cambiamos por *********. El sistema nos pidió ingresar la nueva clave dos veces y luego confirmó que la contraseña fue actualizada correctamente.



```
TPServer [Corriendo] - Oracle VirtualBox
begin: Running /scripts/local-premount ... done.
begin: Will now check root file system ... fsck from util-linux 2.36.1
/sbin/fsck.ext4 (1) -- /dev/sda1] fsck.ext4 -a -O0 /dev/sda1
/dev/sda1: recovering journal
/dev/sda1: clean, 48507/524288 files, 484342/2096896 blocks
done.
[ 1.680976] EXT4-fs (sda1): mounted filesystem with ordered data mode. Opts:
(null)
done.
begin: Running /scripts/local-bottom ... done.
begin: Running /scripts/init-bottom ... done.
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
root@(none):/# [ 1.891746] usb 1-1: new full-speed USB device number 2 using
uhci-pci
[ 2.477413] usb 1-1: New USB device found, idVendor=80ee, idProduct=0021, bcd
Device= 1.00
[ 2.477900] usb 1-1: New USB device strings: Mfr=1, Product=3, SerialNumber=0
[ 2.478189] usb 1-1: Product: USB Tablet
[ 2.478319] usb 1-1: Manufacturer: VirtualBox
passwd root
new password:
Re-type new password:
passwd: password updated successfully
root@(none):/# reboot -f
```

Con la contraseña de root ya modificada, ejecutamos '**reboot -f**' para reiniciar el sistema de forma inmediata. Esto nos permitió continuar con el inicio normal del sistema utilizando las credenciales nuevas.



```
TPServer [Corriendo] - Oracle VirtualBox
Debian GNU/Linux 11 debian tty1
debian login: root
Password:
Linux debian 5.10.0-30-amd64 #1 SMP Debian 5.10.218-1 (2024-06-01) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Dec  4 14:47:26 -03 2024 on tty1
root@debian:~#
```

Luego del reinicio, ya pudimos iniciar sesión normalmente en Debian utilizando el usuario 'root' y la nueva contraseña '*****'.

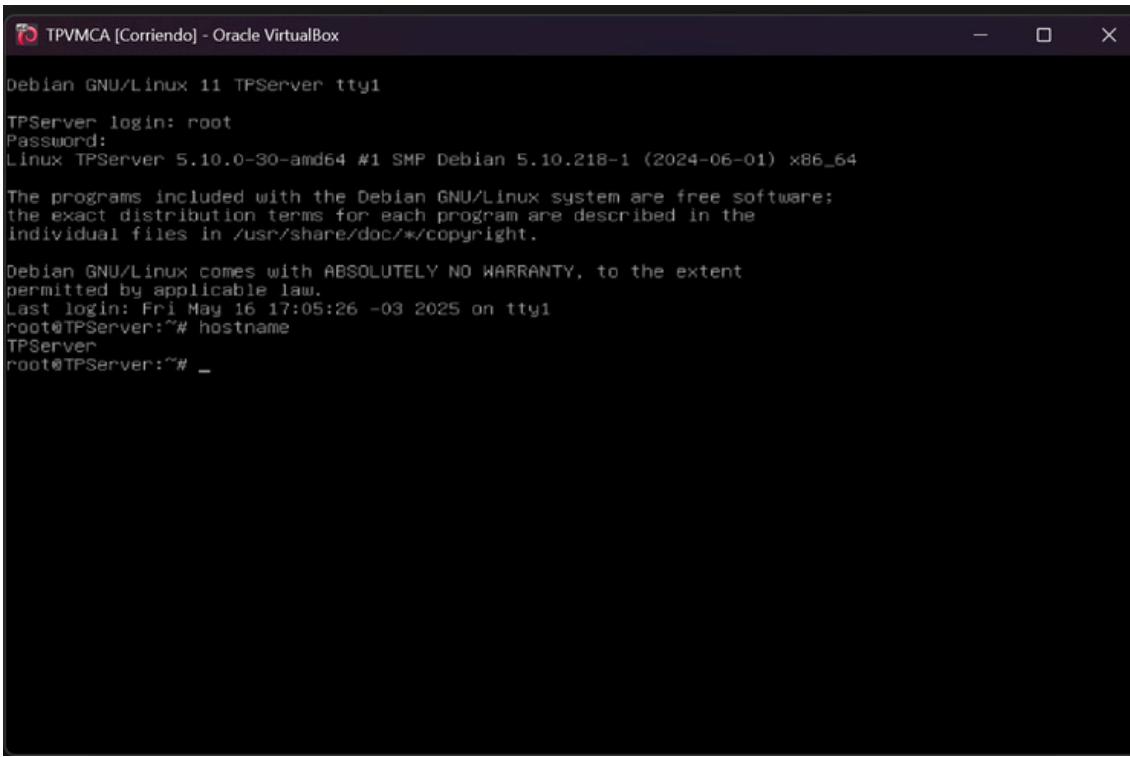
Para lograr establecer el nombre de hostname como **TPServer**, modificamos el archivo hostname (/etc/hostname) mediante la herramienta nano:

```
TPVMCA [Corriendo] - Oracle VirtualBox
GNU nano 5.4                               /etc/hostname *
TPServer

[ 1 línea leída ]
[G] Ayuda      [D] Guardar      [W] Buscar      [K] Cortar      [E] Ejecutar      [C] Ubicación M-U Deshacer
[X] Salir      [R] Leer fich.  [A] Reemplazar  [U] Pegar       [J] Justificar  [I] Ir a linea M-E Rehacer
```

Una vez concluido este paso, reiniciamos la máquina virtual para verificar que el cambio haya sido efectivo, esto se puede hacer con:

- Shutdown -r
- Apagando y prendiendo la MV
- Reboot -f



```
TPVMCA [Corriendo] - Oracle VirtualBox
Debian GNU/Linux 11 TPServer tty1
TPServer login: root
Password:
Linux TPServer 5.10.0-30-amd64 #1 SMP Debian 5.10.218-1 (2024-06-01) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri May 16 17:05:26 -03 2025 on tty1
root@TPServer:~# hostname
TPServer
root@TPServer:~# _
```

2) Servicios

Conexión SSH:

Para nuestra conexión mediante SSH, utilizaremos los archivos brindados en BlackBoard, estos se llaman “**clave publica.pub**” (para la clave publica) y “**clave_privada**” (para la clave privada).

A estos dos archivos se los descargó y se los incorporó a una nueva carpeta creada en el escritorio de la máquina, esta carpeta se llamó “**llaves**”. Para este proceso de envío de clave pública y privada se utilizó el método SSH mediante nuestra **cmd de Windows**.

Primer paso que debemos hacer es instalar o en el caso que ya lo tengamos instalado en la M.V el servicio SSH y modificar su archivo para que root pueda recibir claves, además debemos saber cuál es nuestra IP de la Máquina Virtual

SSH:

- Para instalarlo utilizaremos el comando “**apt-get install openssh-server**”

- Si ya está instalado, solo utilizaremos el comando **systemctl start openssh-server** y **systemctl status openssh-server** para ver si está corriendo.
- Para modificar el archivo de ssh para que root acepte claves públicas, debemos escribir “**nano /etc/ssh/sshd_config**”, este es el archivo de configuración de root **como servidor** y debemos modificarlo de la siguiente forma:

```
GNU nano 5.4                               /etc/ssh/sshd_config
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

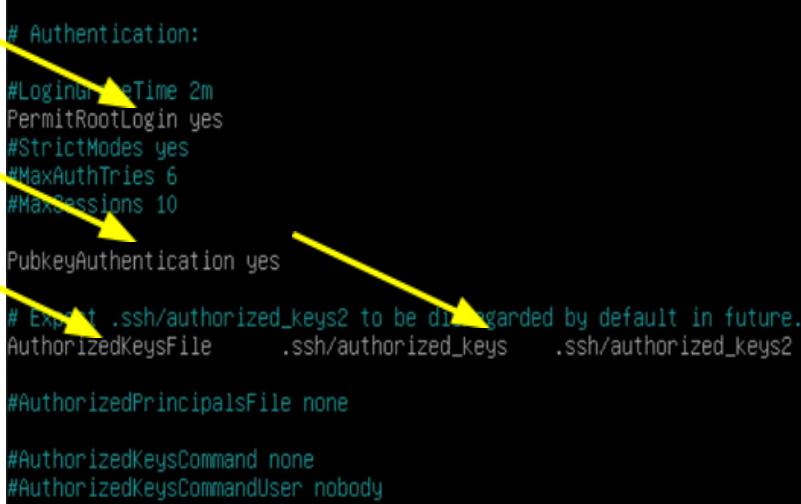
# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# Explicit .ssh/authorized_keys2 to be disregarded by default in future.
AuthorizedKeysFile      .ssh/authorized_keys      .ssh/authorized_keys2

#AuthorizedPrincipalsFile none
#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody
```



The image shows a terminal window displaying the contents of the /etc/ssh/sshd_config file. Three yellow arrows point to specific lines: one to the 'PermitRootLogin yes' line, another to the 'PubkeyAuthentication yes' line, and a third to the '# Explicit .ssh/authorized_keys2 to be disregarded by default in future.' comment.

Las primeras dos opciones que están desmarcadas del #, son específicamente para que el root se pueda loguear y acepte las claves que le enviaremos. Las ultima opcion serian los archivos/carpetas donde podremos guardar las llaves

IP DEL ROOT :

- Debemos aplicar el comando “ip a” y enfocarnos en la especificada como “**enp0s3**” (dirección IP que EL sistema asigna para comunicarse en la red) la cual se vera asi: **XXX.XXX.XXX.XXX/24**

Luego, debemos abrir nuestra cmd de windows(recomiendo entrar a la carpeta llaves creada con anterioridad y hacer click derecho para poder seleccionar la solapa de “Abrir en Terminal”, ¿porque se hace esto?, para que al momento de entrar a la cmd, al costado de la TTY ya tendríamos la ruta completa y solo nos haga falta agregar el archivo que deseamos enviar) y debemos escribir este comando para que la copia del archivo se la clave pública sea exitosa.

```
PS C:\Users\PC\Desktop\llaves> scp "C:\Users\PC\Desktop\llaves\clave_publica" root@192.168.0.40:/root/.ssh/authorized_keys
```

En este caso, el comando scp copia el archivo que es “**clave_publica**” y se envía a root, más específicamente a la carpeta de sshd como una de las carpetas autorizadas, Luego de esto nos pedirá la clave del root del debian y nos dirá que el archivo fue cargado al 100%

Luego verificamos que la clave esté ahí:

```
root@TPServer:~# cat /root/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAAQDom0g03Rr2B2IU1yp3HgGMohZzNFfXxgugYcUrpXjcCQHEJ/x2vmY05M2FF5cN
3fXTDYadWNdtIC3IpCHYD9ds3RCVrjztayH5CsdE5oBImrVea85bYsp2fK1N2/328r4uQh/8+LcpqpGDvT1j2f8GnnPhABQq5lw0
V204D920sbwBQM3wVg9P0M2o8WXjtPpYvVxfQb6R2NB2oY17EUf73rW1m63Qz6kI5tq6yMy2yaqZGr7G4x+yAV7N/YWxqmNz+fE4
Awv3QNyH6xhKIkJRTnajEW68aeQuFX+oYOKBSDhyvYeU2vkj2s7jVWNXXMu07xYQv7G7VKL14N9T13t0i11tsalhyWxBspyzSQmjg
QCrxvCgNUbsaP5/wHATty64QQHLL+r0PHxtds/nJcnwH05fGm3E6iIG4t3ODUrWfd+FkLmlEhzk8+cm8U1G9x58u/NXH4hI264n
WR0K61rnAfSWz1P38B39uFgrjLqz1ja8HmA0BxYLIHmU6nnfTk= sonda@ttys9-HP-ENVY-Notebook
```

Ahora haremos la conexión ssh por medio de la clave privada, en este caso utilizaremos la cmd de windows al igual que con la llave pública, solo que ahora haremos unos cambios:

```
PS C:\Users\PC\Desktop\llaves> ssh -i "C:\Users\PC\Desktop\llaves\clave_privada" root@192.168.0.5 -P 22
Warning: Identity file C:\Users\PC\Desktop\llaves\clave_privada not accessible: No such file or directory.
root@192.168.0.5's password:
```

```
Linux TPServer 5.10.0-30-amd64 #1 SMP Debian 5.10.218-1 (2024-06-01) x86_64
```

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
```

```
Last login: Mon May 26 14:20:59 2025
```

```
root@TPServer:~# |
```

este comando se explica de la siguiente forma:

- **ssh**: le dice al sistema que la conexión será por medio de SSH
 - **-i “RUTA_DEL_ARCHIVO_CLAVE-PRIVADA”**: Le indicas a SSH que use una clave **privada** específica para conectarse.
 - **-P 22**: Especifica el puerto que utilizará ssh (ya por defecto utiliza el 22)

Al igual que con la pública, el root nos pedirá la contraseña para verificar que somos nosotros, luego de colocarla correctamente podemos ver que ahora podremos manejar el Debian desde nuestra cmd de windows, por lo que la conexión SSH se realizó con éxito.

Web:

Como primer paso debemos instalar los archivos “**index.php**” y “**logo.png**” que fueron otorgados en la plataforma de BlackBoard, estos archivos serán guardados en la carpeta “Descargas” que viene incorporada a la computadora personal.

En nuestro Debian, procedemos a instalar los siguientes paquetes:

- **Apache2**= este servidor lo que hace es mostrar páginas web en internet, en este caso muestra la página web con el contenido que tenga guardado en su directorio **/var/www/html/**.

Para instalarlo utilizaremos el comando “`apt-get install apache2`” y lo prenderemos utilizando “`systemctl start apache2`” y verificaremos si está corriendo utilizando “`systemctl status apache2`”

```
[root@TPServer:~# apt-get install apache2
```

```
may 19 15:18:01 TPServer systemd[1]: Starting The Apache HTTP Server...
may 19 15:18:01 TPServer apachectl[483]: AH00557: apache2: apr_sockaddr
```

- **Lenguaje PHP**= este lenguaje de programación tiene la función de crear páginas web “dinámicas”, esto quiere decir que, si el usuario provoca un cambio en el contenido de esa página, PHP actualizara la pagina con esos respectivos cambios

Para instalarlo utilizamos el comando “*apt-get install php7.3*”

```
root@TPServer:~# apt-get install php7.3
```

- **libapache2-mod-php**= su función es la conexión entre apache2 y PHP, permite que apache pueda entender y ejecutar archivos “.php”

Para instalarlo utilizamos el comando “`apt-get install libapache2-mod-php`”

En nuestra Terminal de Windows, procedemos a hacer la conexión ssh para así poder transferir los archivos “index.php” y “logo.png” de nuestro Windows a nuestro Debian.

1. Primero debemos abrir la terminal de Windows (recomiendo ir a nuestra carpeta de descargas de nuestra computadora, Desde allí, hacer a un costado click derecho y apretar la solapa que diga “Abrir en Terminal”, esto se hace especialmente para no gastar tiempo en buscar la ubicación de los archivos)

2. Una vez dentro de la terminal procedemos a escribir estos comandos:

Aquí lo que hacemos es copiar (comando “scp”) los archivos index.php y logo.png, ubicados en la carpeta Downloads, enviarlos a nuestro Debian (root@IP_DEL_DEBIAN), específicamente al directorio “/var/www/html”

De vuelta en nuestro Debian, procedemos a verificar que nuestros archivos estén en el directorio “/var/www/html”, utilizaremos el comando “ls -l /var/www/html”

```
root@TPServer:~# ls -l /var/www/html
total 8
-rw-r--r-- 1 root root 2325 may 19 15:52 index.php
-rw-r--r-- 1 root root 1719 may 19 15:52 logo.png
```

Base de Datos

Ahora lo que necesitamos es una base de datos donde se pueda guardar los datos que ingresamos, en este caso esos datos es el contenido de los archivos “index.php”.

- Según lo requerido utilizaremos la base de datos “**MariaDB**”, esta base de

datos fue creada por los mismos creadores que MySQL, donde se guarda, busca y modifica los datos

Para descargarlo utilizamos el comando “apt-get install mariadb-server” y al igual que como hicimos con apache2 lo iniciaremos y verificamos que este corriendo

```
root@TPServer:~# apt-get install mariadb-server
```

```
root@TPServer:~# systemctl start mariadb
root@TPServer:~# systemctl status mariadb
● mariadb.service - MariaDB 10.5.28 database server
   Loaded: loaded (/lib/systemd/mariadb.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2025-05-19 15:18:01 -03; 1h 22min ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 461 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysqld (code=exi>
   Process: 469 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=ex>
   Process: 475 ExecStartPre=/bin/sh -c [ -e /usr/bin/galera_recovery ] && VAR= || VAR=/usr/b>
   Process: 548 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=e>
   Process: 550 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
 Main PID: 526 (mariadbd)
   Status: "Taking your SQL requests now..."
    Tasks: 9 (limit: 15332)
   Memory: 103.6M
      CPU: 777ms
     CGroup: /system.slice/mariadb.service
             └─526 /usr/sbin/mariadbd

may 19 15:18:01 TPServer mariadb[526]: 2025-05-19 15:18:01 0 [Note] InnoDB: Loading buffer pool(s)>
may 19 15:18:01 TPServer mariadb[526]: 2025-05-19 15:18:01 0 [Note] InnoDB: Buffer pool(s) load co>
may 19 15:18:01 TPServer mariadb[526]: 2025-05-19 15:18:01 0 [Note] Server socket created on IP: '>
may 19 15:18:01 TPServer mariadb[526]: 2025-05-19 15:18:01 0 [Note] Reading of all Master_info ent>
may 19 15:18:01 TPServer mariadb[526]: 2025-05-19 15:18:01 0 [Note] Added new Master_info '' to ha>
may 19 15:18:01 TPServer mariadb[526]: 2025-05-19 15:18:01 0 [Note] /usr/sbin/mariadbd: ready for >
may 19 15:18:01 TPServer mariadb[526]: Version: '10.5.28-MariaDB-0+deb11u2' socket: '/run/mysqld/>
may 19 15:18:01 TPServer systemd[1]: Started MariaDB 10.5.28 database server.
may 19 15:18:01 TPServer /etc/mysql/debian-start[553]: Upgrading MySQL tables if necessary.
may 19 15:18:01 TPServer /etc/mysql/debian-start[568]: Triggering muisam-recover for all MuISAM tab>
```

**ESTA PRENDIDO Y
CORRIENDO**

- Ahora necesitamos que aquellos datos guardados en "index.php" que están en formato ".php" se puedan conectar/comunicar con la base de datos Mariadb, para esto utilizaremos el paquete "**php-mysql**"
Lo instalaremos con el comando "*apt-get install php-mysql*"

```
root@TPServer:~# apt-get install php-mysql
```

- También necesitaremos un cliente con el cual trabajar en la base de datos, para eso utilizaremos el cliente default de MariaDB que es "**mariadb-client**"
Para instalarlo utilizamos el comando "*apt-get install mariadb-client*"

```
root@TPServer:~# apt-get install mariadb-client
```

El paso siguiente es instalar desde el BlackBoard el archivo "**db.sql**" y procedemos a transferirlo a nuestro Debian mediante SSH como lo hicimos anteriormente con nuestros archivos "index.php" y "logo.png". En este caso lo enviaremos al directorio home de root (/root) y verificamos que esté ahí

```
PS C:\Users\PC\Downloads> scp "C:\Users\PC\Downloads\db.sql" root@192.168.0.236:/root
root@192.168.0.236's password:                                         100% 1786   872.0KB/s   00:00
db.sql
```

```
root@TPServer:~# ls -l /root
total 4
-rw-r--r-- 1 root root 1786 may 19 17:05 db.sql
```

Con el archivo ya introducido en el Debian, procedemos a importarlo/introducirlo a la base de datos, para eso utilizaremos el comando "**maradb -u root -p </root/db.sql**", aquí lo que se está haciendo es conectar MariaDB (*mariadb*) con root (-u root) pidiéndole previamente la contraseña a root (-p) y enviándole el contenido de /root/db.sql a MariaDB. Luego procedemos a entrar dentro de MariaDB

```
root@TPServer:~# mariadb -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 32
Server version: 10.5.28-MariaDB-0+deb11u2 Debian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> _
```

agregamos esta configuración:

```
MariaDB [(none)]> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| ingenieria      |
| mysql          |
| performance_schema |
+-----+
4 rows in set (0,001 sec)

MariaDB [(none)]>
MariaDB [(none)]> use ingenieria;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
MariaDB [ingenieria]> show tables;
+-----+
| Tables_in_ingenieria |
+-----+
| alumnos           |
| modulos          |
| notas            |
+-----+
3 rows in set (0,000 sec)

MariaDB [ingenieria]> SELECT * FROM alumnos EXIT;
```

Le decimos a MariaDB que muestre su base de datos

Le decimos que utilice "ingeniería"

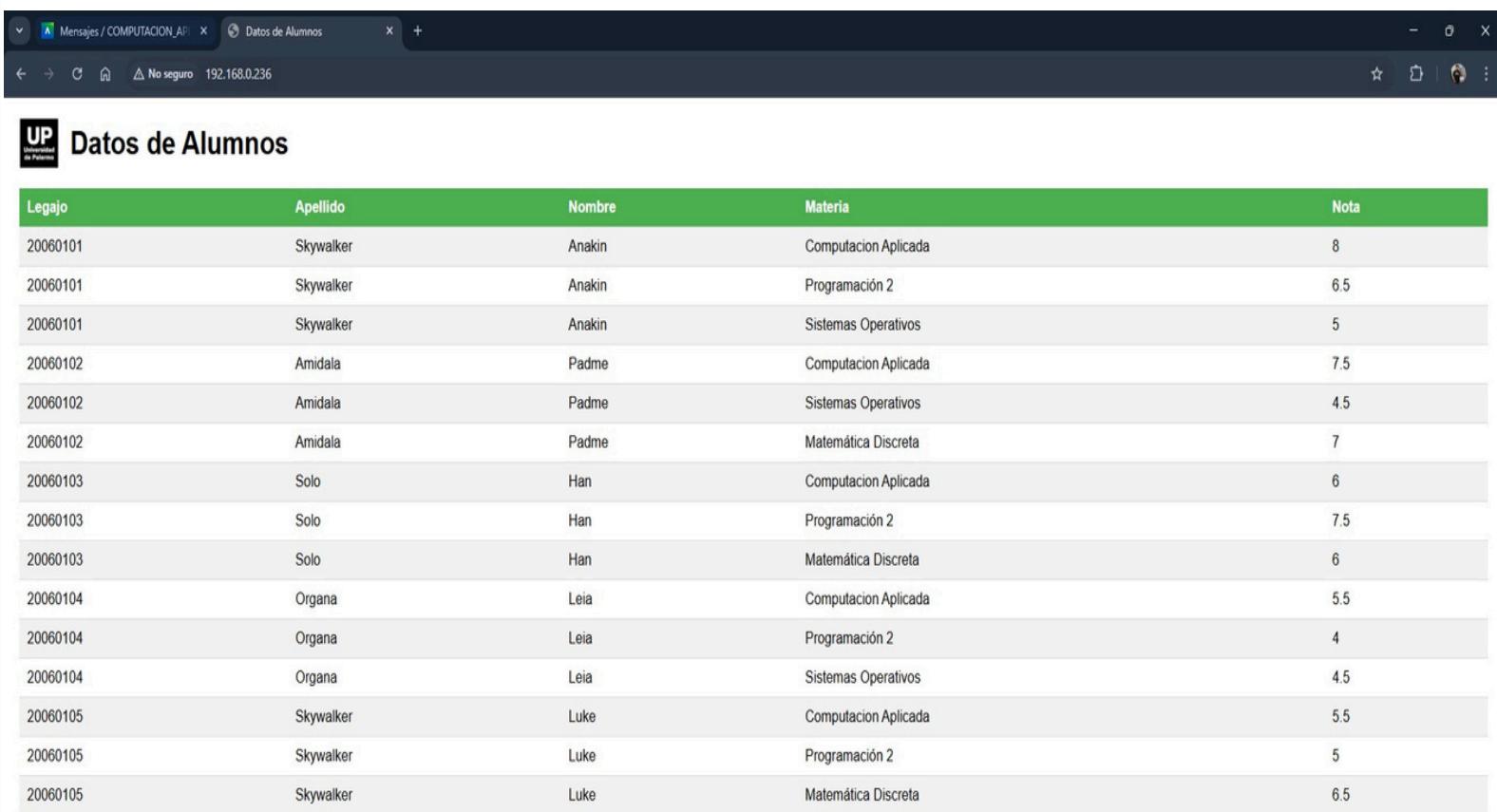
Le decimos que utilice "ingeniería"

```
MariaDB [ingenieria]> SELECT * FROM alumnos
-> exit;
```

Salimos

Muestre el contenido de "alumnos"

Luego de eso, procedemos a ir a nuestro navegador de confianza para poner lo siguiente en el buscador “http://@IP_DEL_DEBIAN/” y nos mostrará una tabla de alumnos ficticia.



A screenshot of a web browser window titled "Datos de Alumnos". The URL bar shows "192.168.0.236". The page content is a table with the following data:

Legajo	Apellido	Nombre	Materia	Nota
20060101	Skywalker	Anakin	Computacion Aplicada	8
20060101	Skywalker	Anakin	Programación 2	6.5
20060101	Skywalker	Anakin	Sistemas Operativos	5
20060102	Amidala	Padme	Computacion Aplicada	7.5
20060102	Amidala	Padme	Sistemas Operativos	4.5
20060102	Amidala	Padme	Matemática Discreta	7
20060103	Solo	Han	Computacion Aplicada	6
20060103	Solo	Han	Programación 2	7.5
20060103	Solo	Han	Matemática Discreta	6
20060104	Organa	Leia	Computacion Aplicada	5.5
20060104	Organa	Leia	Programación 2	4
20060104	Organa	Leia	Sistemas Operativos	4.5
20060105	Skywalker	Luke	Computacion Aplicada	5.5
20060105	Skywalker	Luke	Programación 2	5
20060105	Skywalker	Luke	Matemática Discreta	6.5

3) Redes:

```
Sufijo DNS específico para la conexión. . . :  
Vínculo: dirección IPv6 local. . . : fe80::d47d:a812:484f:72%18  
Dirección IPv4. . . . . : 192.168.56.1  
Máscara de subred . . . . . : 255.255.255.0  
Puerta de enlace predeterminada . . . . . :  
  
Adaptador de LAN inalámbrica Conexión de área local* 9:  
  
Estado de los medios. . . . . : medios desconectados  
Sufijo DNS específico para la conexión. . . :  
  
Adaptador de LAN inalámbrica Conexión de área local* 10:  
  
Estado de los medios. . . . . : medios desconectados  
Sufijo DNS específico para la conexión. . . :  
  
Adaptador de LAN inalámbrica Wi-Fi:  
  
Sufijo DNS específico para la conexión. . . : fibertel.com.ar  
Vínculo: dirección IPv6 local. . . : fe80::4ec7:fece:547d:4ca9%19  
Dirección IPv4. . . . . : 192.168.0.30  
Máscara de subred . . . . . : 255.255.255.0  
Puerta de enlace predeterminada . . . . . : 192.168.0.1  
  
Adaptador de Ethernet Conexión de red Bluetooth:  
  
Estado de los medios. . . . . : medios desconectados  
Sufijo DNS específico para la conexión. . . :  
PS C:\Users\PC\Desktop>
```

Esta imagen muestra la salida del comando '**ipconfig**', utilizado en sistemas Windows para consultar la configuración de red de las interfaces activas o inactivas. Se identifican varias interfaces, pero nos centraremos en la activa: el adaptador de LAN inalámbrica Wi-Fi.

- **Dirección IPv4: 192.168.0.30** – Esta es la dirección IP local asignada a la computadora dentro de la red del hogar.
- **Máscara de subred: 255.255.255.0** – Indica que la red está compuesta por 254 posibles direcciones IP utilizables.
- **Puerta de enlace predeterminada: 192.168.0.1** – Es la dirección del router, que permite la salida a Internet.

Las demás interfaces están desconectadas (Conexión de área local 9 y 10, y la interfaz Bluetooth). Esto es muy común y no indica ningún problema.

```
GNU nano 5.4                               /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s3
allow-hotplug enp0s3
iface enp0s3 inet static
    address 192.168.0.40
    netmask 255.255.255.0
    gateway 192.168.0.1
```

En esta imagen mostramos el archivo '**/etc/network/interfaces**' que define cómo se configuran las interfaces de red en sistemas Debian o derivados.

Se especifican dos interfaces:

- **'lo'**: interfaz loopback, usada para el tráfico interno (localhost).
- **allow-hotplug**: la interfaz se debe activar automáticamente cuando sea detectada
- **'enp0s3'**: interfaz física de red con configuración IP estática.
- **iface**: Sirve para **declarar y configurar una interfaz de red** específica
 - inet:usa el **protocolo IPv4**.

Configuración estática

- Dirección IP: 192.168.0.40 (La dirección IP es distinta a la cmd de windows para que no haya un conflicto de red)
- Máscara de subred: 255.255.255.0
- Puerta de enlace: 192.168.0.1

Esta configuración permite que la máquina tenga una dirección IP fija dentro de la red local.

Es importante reiniciar la red después de editar este archivo usando comandos como 'systemctl restart networking' o reiniciando el equipo.

```
root@TPServer:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=116 time=18.6 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=116 time=17.0 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=116 time=15.7 ms
```

Utilizamos el comando '**ping 8.8.8.8**' para comprobar la conectividad con Internet. Y 8.8.8.8 es un servidor DNS público de Google.

Por lo que podemos ver:

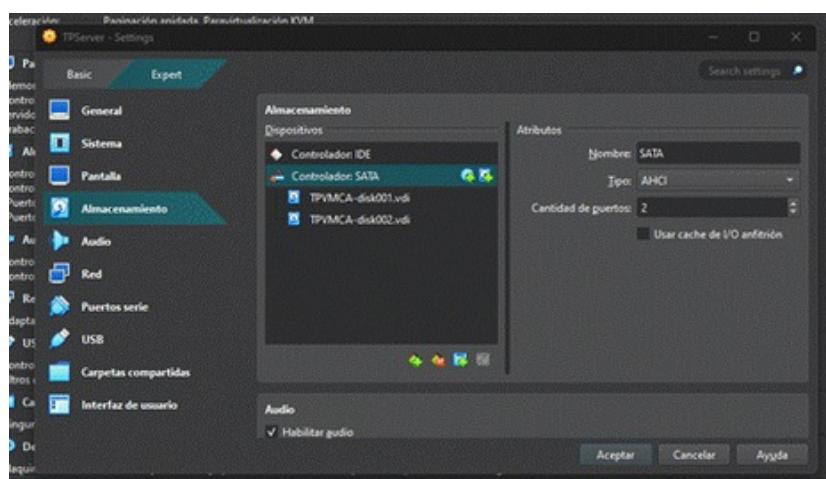
- Qué hay conexión a Internet (los paquetes fueron enviados y recibidos).
- El tiempo de respuesta (latencia) es bajo (15 a 18 ms), lo que **significa una conexión estable**.
- El valor TTL=116 indica cuántos saltos de red están permitidos antes de que el paquete sea descartado. No requiere acción.

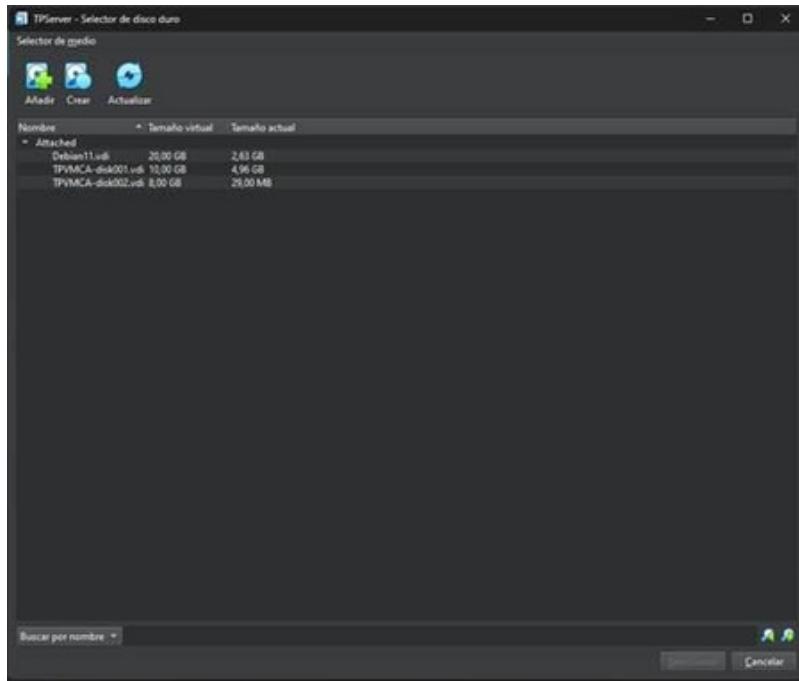
4) Almacenamiento

Paso 1: Agregar discos virtuales

Se añaden dos discos virtuales en la máquina virtual a través de VirtualBox:

- TPVMCA-disk001.vdi de 10 GB
- TPVMCA-disk002.vdi de 8 GB





Estos discos se agregan al controlador SATA, configurado en modo AHCI con 2 puertos disponibles. Esto se realiza desde el menú de almacenamiento de la configuración de la máquina virtual.

Paso 2: Crear particiones en los discos

Se ejecuta el comando **fdisk** para crear las siguientes particiones en el “directorio de almacenamiento”:

```
root@TPServer:~# fdisk /dev/sdc
Bienvenido a fdisk (util-linux 2.36.1).
Los cambios solo permanecerán en la memoria, hasta que decida escribirlos.
Tenga cuidado antes de utilizar la orden de escritura.

El dispositivo no contiene una tabla de particiones reconocida.
Se ha creado una nueva etiqueta de disco DOS con el identificador de disco 0x64a4da28.

Orden (m para obtener ayuda): n
Tipo de partición
  p  primaria (0 primaria(s), 0 extendida(s), 4 libre(s))
  e  extendida (contenedor para particiones lógicas)
Seleccionar (valor predeterminado p): p
Número de partición (1-4, valor predeterminado 1):
Primer sector (2048-20971519, valor predeterminado 2048):
Último sector, +/-sectores o +/-tamaño[K,M,G,T,P] (2048-20971519, valor predeterminado 20971519): +3
Crea una nueva partición 1 de tipo 'Linux' y de tamaño 3 GiB.

Orden (m para obtener ayuda): _
```

```
Orden (m para obtener ayuda): n
Tipo de partición
  p  primaria (1 primaria(s), 0 extendida(s), 3 libre(s))
  e  extendida (contenedor para particiones lógicas)
Seleccionar (valor predeterminado p): p
Número de partición (2-4, valor predeterminado 2):
Primer sector (6293504-20971519, valor predeterminado 6293504):
Último sector, +/-sectores o +/-tamaño[K,M,G,T,P] (6293504-20971519, valor predeterminado 20971519): +6G
Crea una nueva partición 2 de tipo 'Linux' y de tamaño 6 GiB.
```

- En /dev/sdc:

- **/dev/sdc1**: partición primaria de 3 GiB (para /www_dir)
- **/dev/sdc2**: partición primaria de 6 GiB (para /backup_dir)

Se confirma que se están usando sectores adecuados y se define el tipo como Linux.

Paso 3: Confirmar particiones

```
root@TPServer:~# lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda     8:0    0 10G  0 disk
└─sda1  8:1    0  8G  0 part /
└─sda2  8:2    0  1K  0 part
└─sda5  8:5    0  2G  0 part [SWAP]
sdb     8:16   0  8G  0 disk
└─sdb1  8:17   0  8G  0 part /home
sdc     8:32   0 10G  0 disk
└─sdc1  8:33   0  3G  0 part
└─sdc2  8:34   0  6G  0 part
```

El comando **lsblk** muestra la estructura de particionado de los discos. Se puede verificar que:

- sdc1 tiene 3G y será montado en /www_dir
 - sdc2 tiene 6G y será montado en /backup_dir
-

Paso 4: Formateo de particiones

Se formatean las nuevas particiones usando el sistema de archivos ext4:

con el comando “**mkfs -t ext4 “nombre de partición”** podremos formatear el disco

- **mkfs**: crea el filesystems
- **-t ext4**: especifica el tipo filesystem, en este caso es ext4

```
root@TPServer:~# mkfs -t ext4 /dev/sdc1
mke2fs 1.46.2 (28-Feb-2021)
Creating filesystem with 786432 4k blocks and 196608 inodes
Filesystem UUID: 02c84ca0-1b0a-4bca-8af3-67e67ca2d6f2
Superblock backups stored on blocks:
      32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

root@TPServer:~# mkfs -t ext4 /dev/sdc2
mke2fs 1.46.2 (28-Feb-2021)
Creating filesystem with 1572864 4k blocks and 393216 inodes
Filesystem UUID: 4bc6c128-03b5-49e8-b58b-fb22af40040b
Superblock backups stored on blocks:
      32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

Esto prepara los discos para ser usados con Linux, asignándoles inodos, bloques y UUID únicos.

Paso 5: Montaje manual de las particiones

Se montan temporalmente las particiones para comprobar que funcionan correctamente:

```
root@TPServer:~# mount /dev/sdc1 /www_dir/
root@TPServer:~# mount /dev/sdc2 /backup_dir/
```

Paso 6: Verificación con df -h

Se comprueba que los puntos de montaje están activos y disponibles. El sistema detecta:

```
root@TPServer:~# df -h
S.ficheros      Tamaño Usados  Disp Uso% Montado en
udev            968M     0  968M  0% /dev
tmpfs           198M   516K  197M  1% /run
/dev/sda1        7,8G  2,1G  5,3G  29% /
tmpfs           986M     0  986M  0% /dev/shm
tmpfs           5,0M     0  5,0M  0% /run/lock
/dev/sdb1        7,8G    32K  7,4G  1% /home
tmpfs           198M     0  198M  0% /run/user/0
/dev/sdc1        2,9G   24K  2,8G  1% /www_dir
/dev/sdc2        5,9G   24K  5,6G  1% /backup_dir
```

/dev/sdc1 montado en /www_dir con 2,9G disponibles

/dev/sdc2 montado en /backup_dir con 5,9G disponible

Paso 7: Configurar Apache para usar /www_dir,

Se edita el archivo de configuración de Apache (000-default.conf) para que el DocumentRoot apunte a /www_dir en lugar de su ruta predeterminada:

```
GNU nano 5.4                               /etc/apache2/sites-available/000-default.conf *
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    
        ServerAdmin webmaster@localhost
        DocumentRoot /www_dir
        <Directory /www_dir>
            Require all granted
        </Directory>
    

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

- **ServerAdmin webmaster@localhost** : Esta directiva indica el correo electrónico del administrador del sitio web.
- **DocumentRoot /www_dir**: Habilita a root pueda acceder a los “.html” o “.php” que están dentro de la carpeta /www_dir
- **<Directory /www_dir>**: indica que las reglas de Apache se apliquen al directorio /www_dir.
 - **Require all granted**: permite el acceso a todos los clientes sin ninguna restricción.
- **</directory>**: toda configuración que se haga solamente se aplicará en el directorio correspondiente

Paso 8: Copiar archivos web al nuevo directorio

Se copian index.php y logo.png al nuevo directorio web:

```
root@TPServer:/# cp /var/www/html/index.php /www_dir
root@TPServer:/# ls /www_dir
index.php  lost+found
root@TPServer:/# cp /var/www/html/logo.png /www_dir
root@TPServer:/# ls -l /www_dir
ls: no se puede acceder a '-': No existe el fichero o el directorio
ls: no se puede acceder a 'l': No existe el fichero o el directorio
/www_dir:
index.php  logo.png  lost+found
root@TPServer:/# ls -l /www_dir
total 24
-rw-r--r-- 1 root root  2325 may 20 19:33 index.php
-rw-r--r-- 1 root root  1719 may 20 19:33 logo.png
drwx----- 2 root root 16384 may 20 18:59 lost+found
root@TPServer:/#
```

Paso 9: Montaje automático – edición de /etc/fstab

Se edita el archivo con NANO. `/etc/fstab`, este archivo sirve para que las particiones se monten automáticamente al iniciar el sistema:

```
GNU nano 5.4                               /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options>      <dump> <pass>
# / was on /dev/sda1 during installation
UUID=cd2cb4ca-39bd-4522-919c-e4ceb81b5ae8 /          ext4    errors=remount-ro 0      1
# swap was on /dev/sda5 during installation
UUID=e550cf9e-dc94-442c-b9c4-9c12f05024b3 none        swap     sw      0      0
/dev/sr0       /media/cdrom0  udf,iso9660 user,noauto  0      0
/dev/sdb1      /home        ext4    rw,auto       0      0
/dev/sdc1      /www_dir    ext4    rw,auto       0      2
/dev/sdc2      /backup_dir ext4    rw,auto       0      2
```

Paso 10: Confirmar UUIDs y sistema de archivos

El comando `lsblk -f` permite ver el UUID de cada partición, útil en caso de que se desee usar UUID en lugar del path en fstab.

NAME	FSTYPE	FSVER	LABEL	UUID	FSAVAIL	FSUSE%	MOUNTPOINT
sda							
└─sda1	ext4	1.0		cd2cb4ca-39bd-4522-919c-e4ceb81b5ae8	5,3G	27%	/
└─sda2							
└─sda5	swap	1		e550cf9e-dc94-442c-b9c4-9c12f05024b3			[SWAP]
sdb							
└─sdb1	ext4	1.0		ba0d9c77-fb77-4d0f-9193-50d3788762cf	7,4G	0%	/home
sdc							
└─sdc1	ext4	1.0		02c84ca0-1b0a-4bca-8af3-67e67ca2d6f2	2,7G	0%	/www_dir
└─sdc2	ext4	1.0		4bc6c128-03b5-49e8-b58b-fb22af40040b	5,5G	0%	/backup_dir

Paso 11: Probar la configuración con mount -a

```
root@TPServer:/# mount -a
root@TPServer:/# _
```

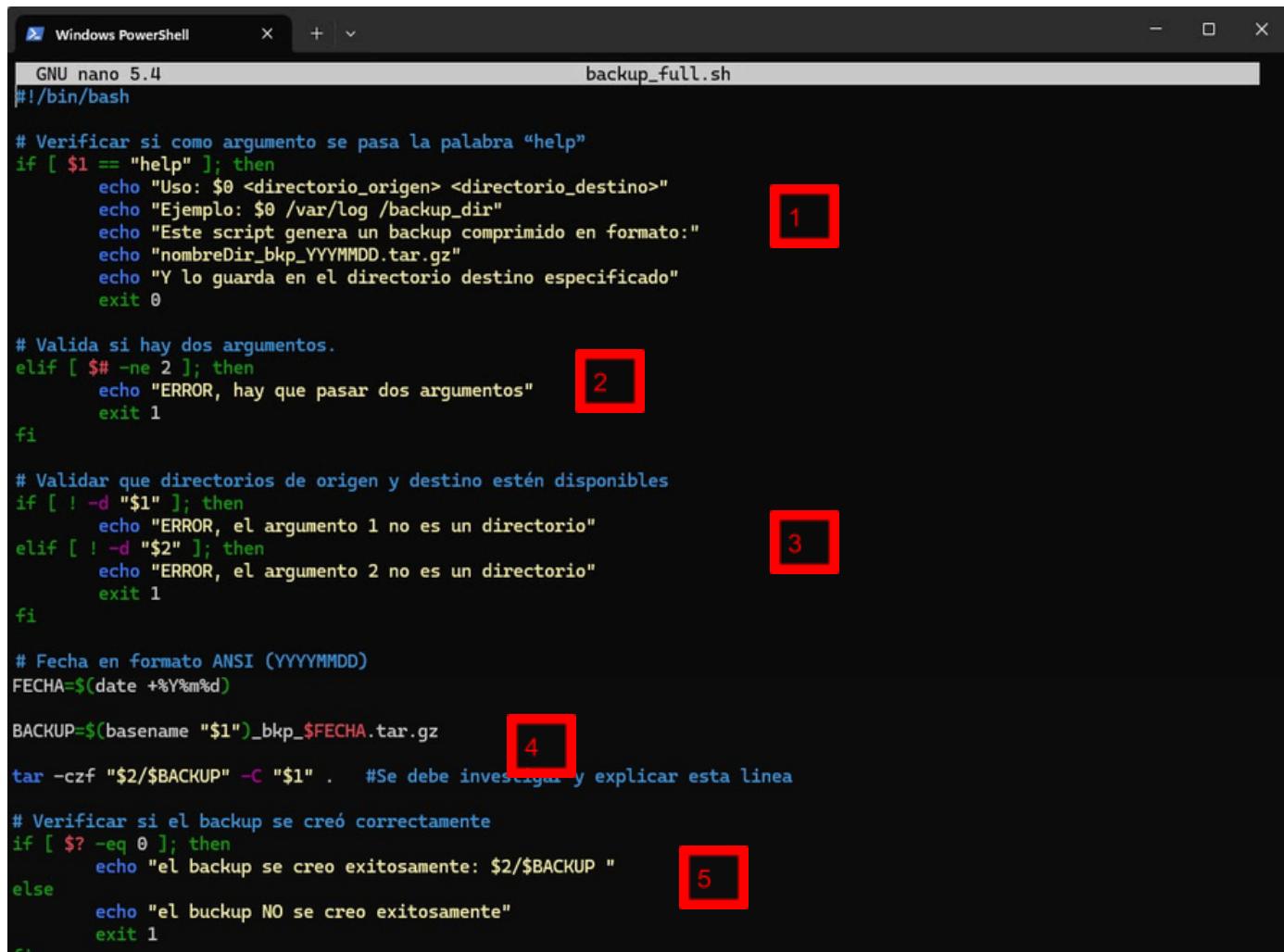
Este comando (`mount -a`) intenta montar todas las particiones que están en `/etc/fstab`. Si no hay errores, significa que está correctamente configurado.

5) Back Up

Para poder realizar este punto, como primer paso debemos crear el directorio /opt/scripts, para eso utilizaremos “**mkdir /opt/scripts**” para así poder introducir en ese directorio nuestro script. (“**opt**”—> directorio para instalar software adicional)

Una vez dentro de él procedemos a crear un editor de texto con la herramienta “nano” llamado “**backup_full.sh**” para así poder pegar nuestro script, en nuestro caso utilizamos la cmd de windows mediante conexión SSH ya que nos permite copiar y pegar más fácil.

```
root@TPServer:~# cd /opt/scripts
root@TPServer:/opt/scripts# nano backup_full.sh
```



```
GNU nano 5.4                                     backup_full.sh
#!/bin/bash

# Verificar si como argumento se pasa la palabra "help"
if [ $1 == "help" ]; then
    echo "Uso: $0 <directorio_origen> <directorio_destino>" 1
    echo "Ejemplo: $0 /var/log /backup_dir"
    echo "Este script genera un backup comprimido en formato:"
    echo "nombreDir_bkp_YYYYMMDD.tar.gz"
    echo "Y lo guarda en el directorio destino especificado"
    exit 0

# Valida si hay dos argumentos.
elif [ $# -ne 2 ]; then
    echo "ERROR, hay que pasar dos argumentos" 2
    exit 1
fi

# Validar que directorios de origen y destino estén disponibles
if [ ! -d "$1" ]; then
    echo "ERROR, el argumento 1 no es un directorio" 3
elif [ ! -d "$2" ]; then
    echo "ERROR, el argumento 2 no es un directorio"
    exit 1
fi

# Fecha en formato ANSI (YYYYMMDD)
FECHA=$(date +%Y%m%d)

BACKUP=$(basename "$1")_bkp_$FECHA.tar.gz 4
tar -czf "$2/$BACKUP" -C "$1" .    #Se debe investigar y explicar esta linea

# Verificar si el backup se creó correctamente
if [ $? -eq 0 ]; then
    echo "el backup se creo exitosamente: $2/$BACKUP" 5
else
    echo "el backup NO se creo exitosamente"
    exit 1
fi
```

Este es nuestro script, el mismo fue proporcionado gracias al profesor en las ayudas otorgadas.

las siguientes resoluciones e explican de esta forma:

- 1.** en la primera parte del script, básicamente decimos que si el argumento 1 (\$1) contiene la palabra “help”, el script nos mostrará el manual de ayuda para el usuario (exit 0= salida exitosa)
- 2.** luego se valida si hay dos argumentos, si el argumento ingresado (#\$) es NO igual (-ne) a 2 (osea son dos argumentos) entonces la salida es errónea ya que solo puede existir un solo argumento

3. se validan los directorios de origen, si el argumento 1 y 2 (\$1, \$2) no son (!) directorios (-d) entonces la salida es errónea (exit1)
4. “**BACKUP=\$(basename "\$1")_bkp_\${FECHA}.tar.gz**” se explica de la siguiente forma nos dice que el nombre (basename) del argumento 1(\$1) será guardado en tipo backup (_bkp_) con su fecha de ejecución (\$FECHA) en un tipo de formato comprimido tar.gz

tar -czf "\$2/\$BACKUP" -C "\$1", nos indica que el archivo donde se guarda el script será comprimido (tar) en un archivo .tar, utilizando gzip con la salida en el archivo file (-czf), \$2/\$BACKUP” es el archivo de salida junto a la ruta del script.

Nuestro último comando dice que si el último comando ejecutado (?) es igual

5. (-eq) a 0, entonces tiene una salida exitosa.

Luego de esto, procederemos a ver si el script colocado en nuestro editor de texto se ejecuta de manera correcta. Primero debemos establecer permisos de ejecución para este archivo:

```
root@TPServer:/opt/scripts# chmod u+x backup_full.sh
root@TPServer:/opt/scripts# ls -l
total 4
-rwxr--r-x 1 root root 1057 jun  3 19:42 backup_full.sh
```

Luego veremos si al escribir como argumento 1 “help” se ejecuta correctamente. Para esto escribimos la ruta absoluta de la ubicación del script junto al argumento “help”

```
root@TPServer:/opt/scripts# /opt/scripts/backup_full.sh help
Uso: /opt/scripts/backup_full.sh <directorio_origen> <directorio_destino>
Ejemplo: /opt/scripts/backup_full.sh /var/log /backup_dir
Este script genera un backup comprimido en formato:
nombreDir_bkp_YYYYMMDD.tar.gz
Y lo guarda en el directorio destino especificado
```

“Nos tendría que devolver el manual de help que pusimos con anterioridad en el script”

Ahora ejecutamos todo el script para saber si esta funcionando bien, para eso escribiremos la ruta absoluta de la ubicación del script agregado el directorio donde se va a backuppear (/var/log) y donde se van a guardar los archivos (/backup_dir).

```
root@TPServer:/opt/scripts# /opt/scripts/backup_full.sh /var/log /backup_dir
el backup se creo exitosamente: /backup_dir/log_bkp_20250603.tar.gz
```

Luego de verificar que el script funciona correctamente, procedemos a modificar los horarios de ejecución, para eso utilizamos el programa “**cron**” que se ejecuta con el comando “**crontab -e**”

Dentro del archivo modifiable de cron, procedemos a colocar que :

- el archivo de /opt/scripts/backup_full.sh se haga backup en /var/log y se guarde en el directorio backup_dir y su salida stdout y stderr se envíen a /var/log/backup_logs.log **TODOS LOS DÍAS a las 00:00 hs**

- Lo mismo pero ahora el script haga backup en /www_dir y su salida stdout y stderr se envíen a /var/log/backup_logs.log **LUNES, MIÉRCOLES, VIERNES a las 23:00 hs**

```
GNU nano 5.4                               /tmp/crontab.Yiig59/crontab *
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
@reboot /usr/sbin/iptables-restore /root/myfw.txt
00 00 * * * /opt/scripts/backup_full.sh /var/log /backup_dir >> /var/log/backup_logs.log 2>&1
00 23 * * mon,wed,fri /opt/scripts/backup_full.sh /www_dir /backup_dir >> /var/log/backup_logs.log >
```

PLUS: Para poder verificar que esto funciona, lo que hicimos fue cambiar el horario de los dos archivos para que se ejecuten todos los minutos de todas las horas, luego hicimos un cat de "/var/log/backup_logs.log" para ver si los resultados se enviaban, este fue el resultado:

```
root@TPServer:/# cat /var/log/backup_logs.log
el backup se creo exitosamente: /backup_dir/log_bkp_20250603.tar.gz
el backup se creo exitosamente: /backup_dir/log_bkp_20250603.tar.gz
```

Como se ve, el resultado del script fue redireccionado a /var/log/backup_logs.log durante todos los minutos.

Diagrama Topográfico

