Integrantes:

Claudio Vera, Ivan Kvaternik, Nicolás Nahuel Palazzo , Jesús Gianini, Lucas Demarré

TRABAJO INTEGRADOR GRUPAL

Computacion Aplicada

*1) Configuración del entorno*

1) La máquina virtual está dividida en partes y comprimidas en formato “**.rar”**. Descargar y ensamblar los archivos utilizando herramientas como WinRar.

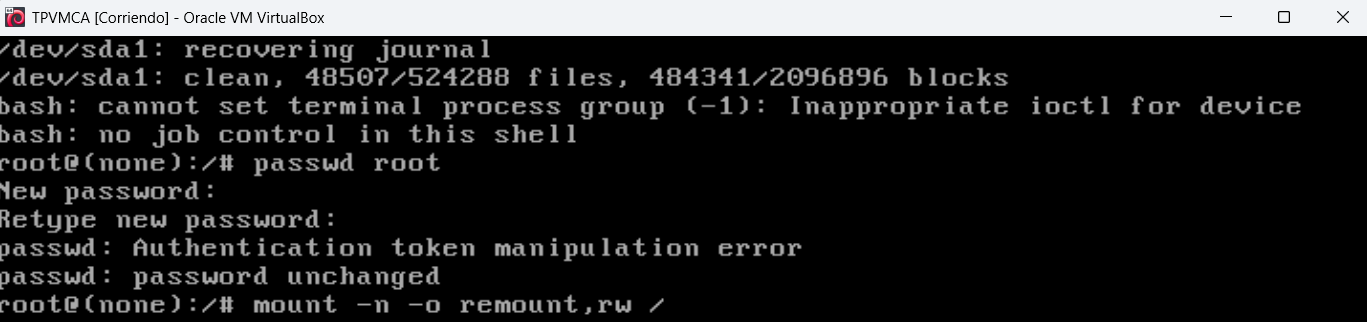


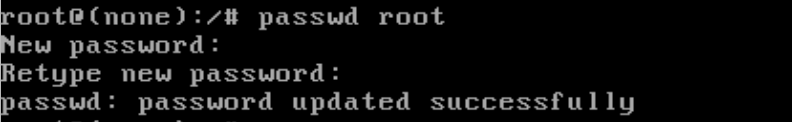
2) La clave de ***root*** es desconocida inicialmente, por lo que deberá ser blanqueada antes de comenzar. Una vez dentro del sistema operativo, la misma debe ser cambiada por “***palermo***” (sin comillas).

**Accedemos a la configuración de grub en el boteo editamos la línea para iniciar como root**

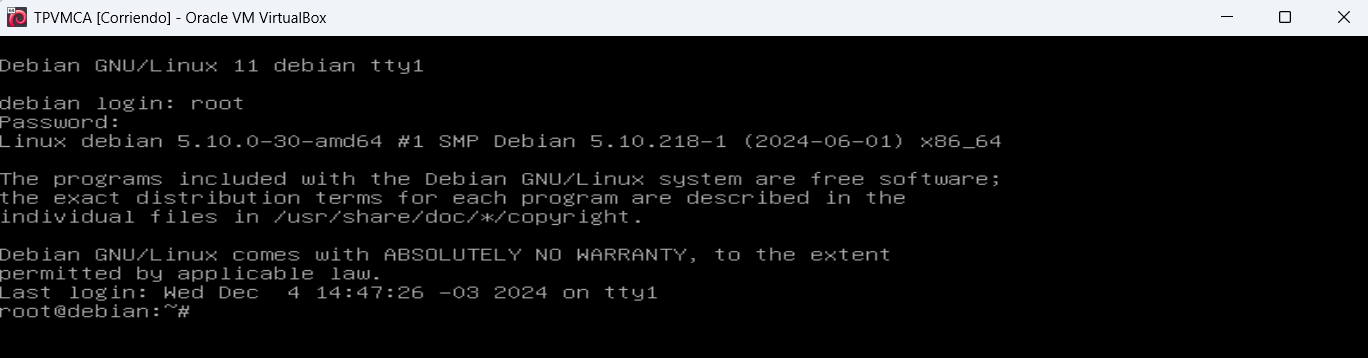


**Inicia en modo root y cambiamos la password**





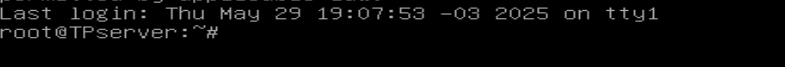
**Una vez reiniciada la maquina ya podemos loguearnos con password palermo**



**3) Establecer el nombre de *hostname* como TPServer.**



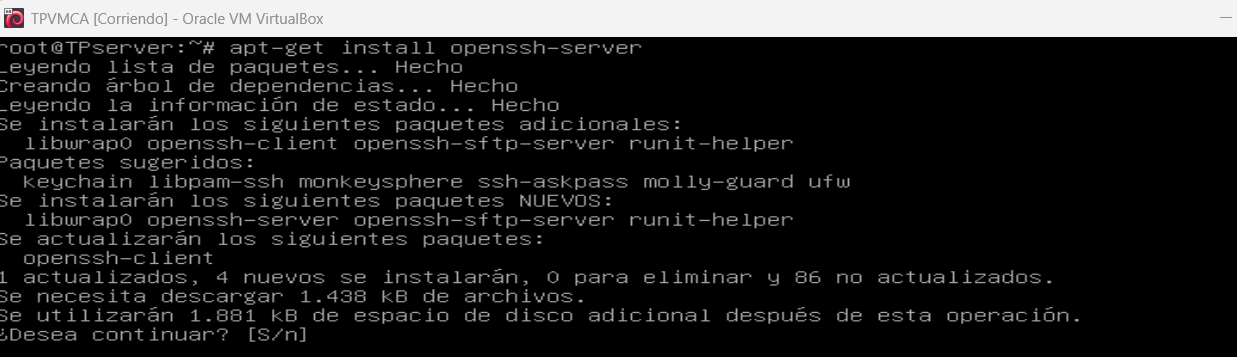
**Reiniciamos y ya vemos el nombre**

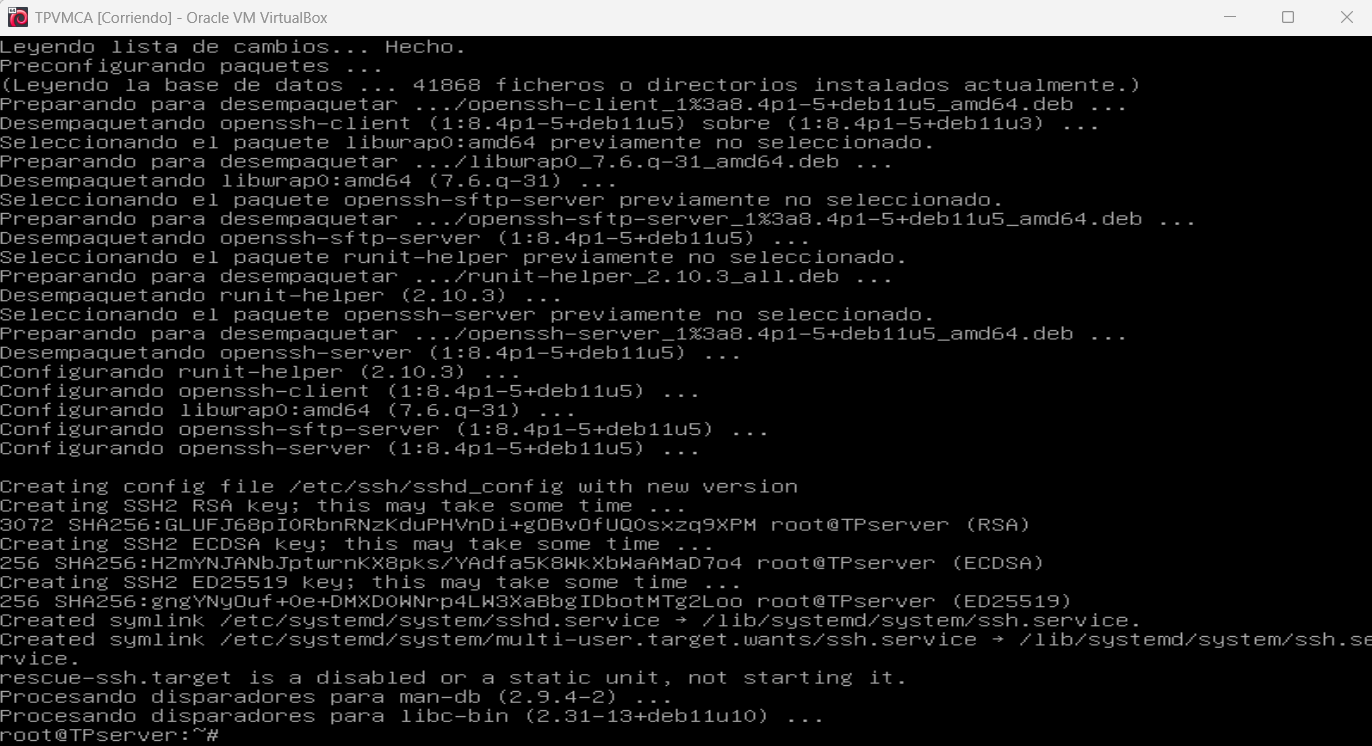


*2) Servicios*

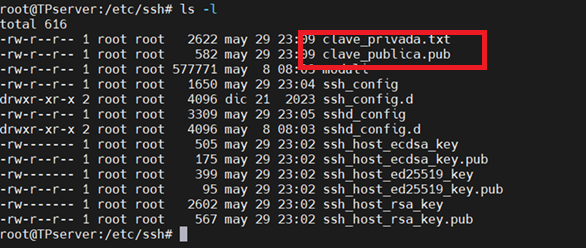
1) **SSH:** instalar y configurar el servicio de **SSH**. El servidor debe permitir el acceso al usuario *root* mediante una clave privada/pública, proporcionada junto con la máquina virtual en **Blackboard.**

**Instalamos el servidor ssh**

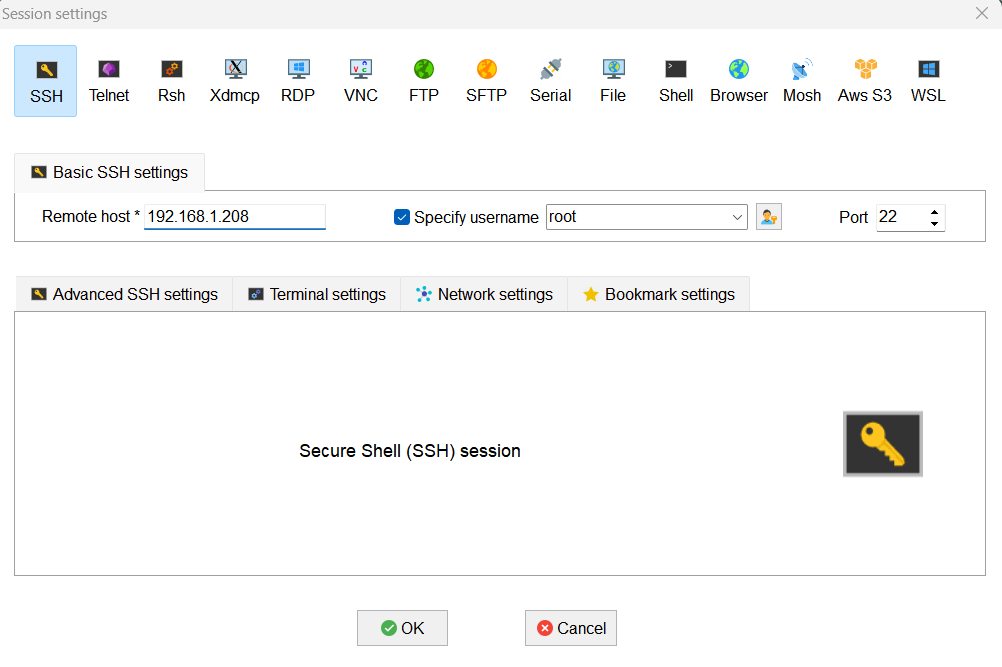


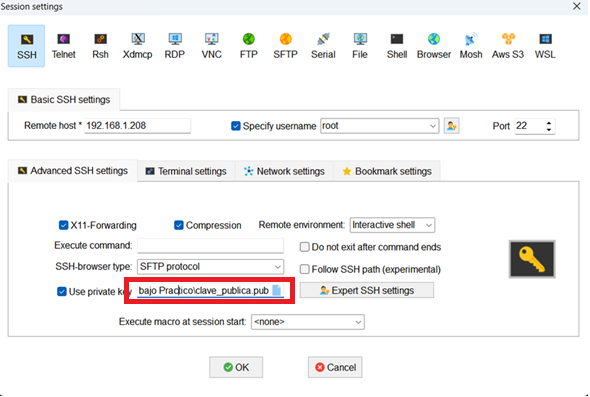


**Copiamos las claves proporcionadas para el trabajo practico**

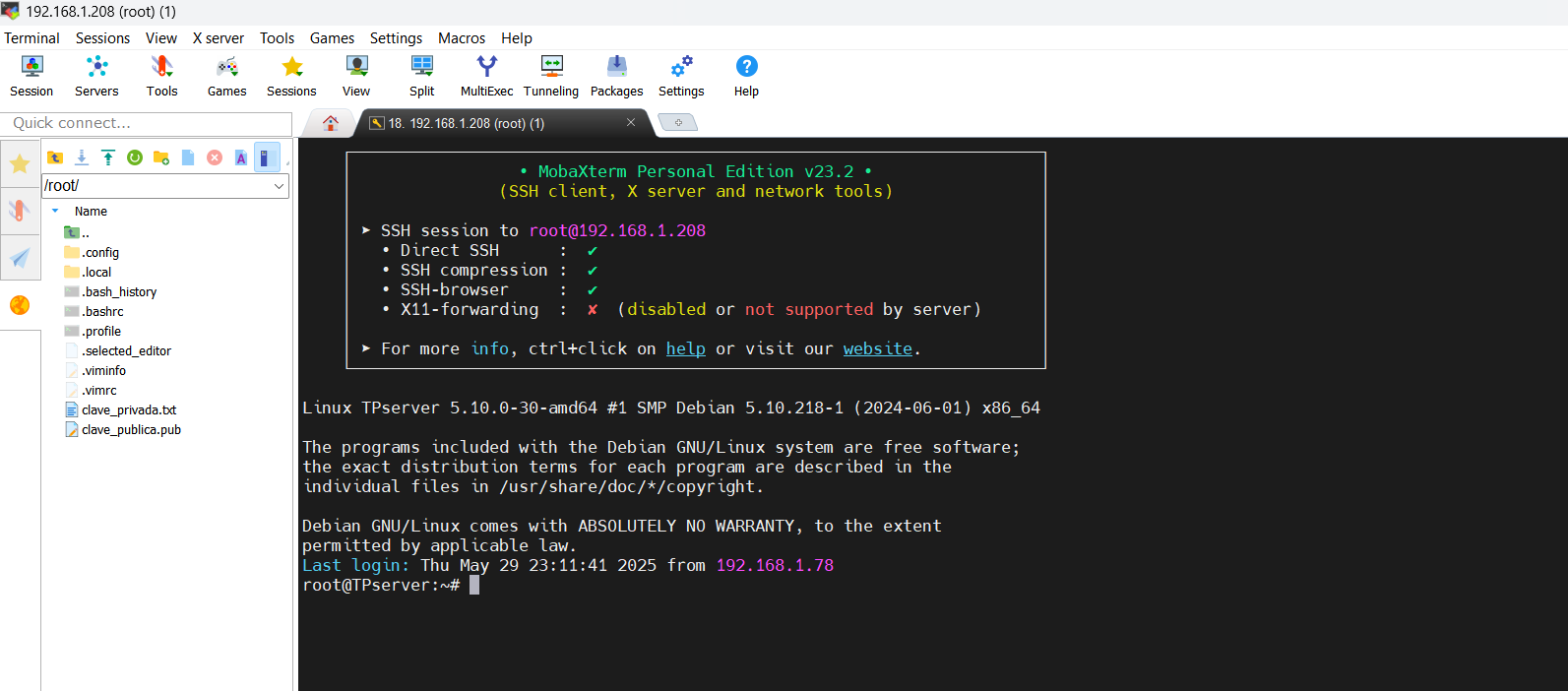


**Configuramos una conexión usando la clave publica**





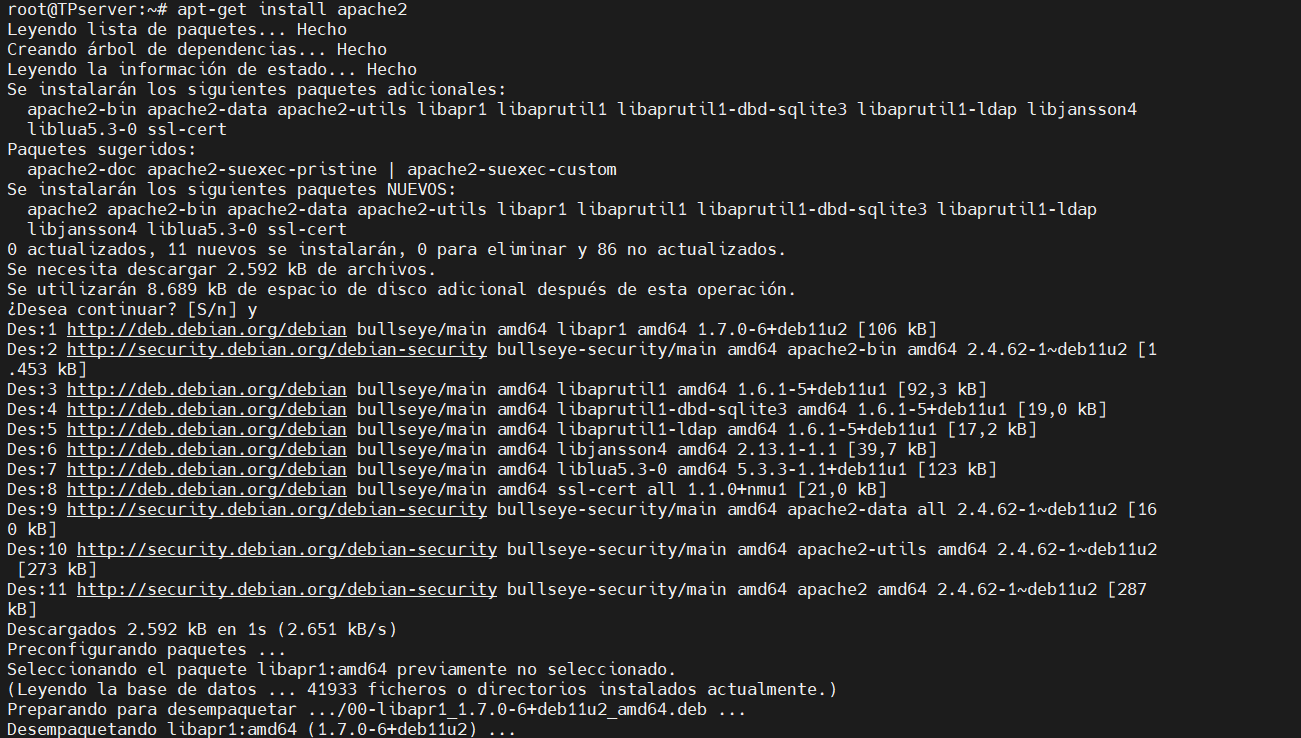
**Una vez configurada la conexión con la clave pública no nos tiene que pedir contraseña al conectar**



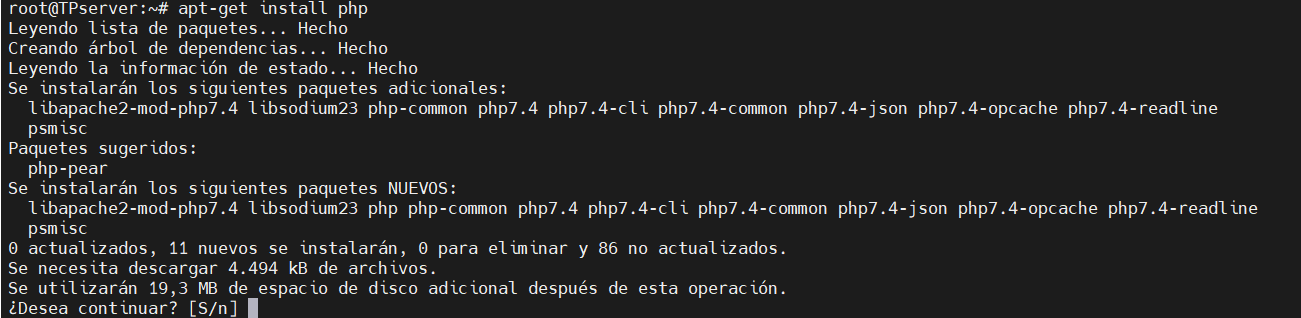
2) **WEB:** instalar y configurar el servidor **Apache** con soporte para PHP (versión 7.3 o superior). Configurar el servidor para servir el archivo “index.php” y “logo.png”, disponible junto con la máquina virtual en **Blackboard**.

**Buscamos e Instalamos el apache**

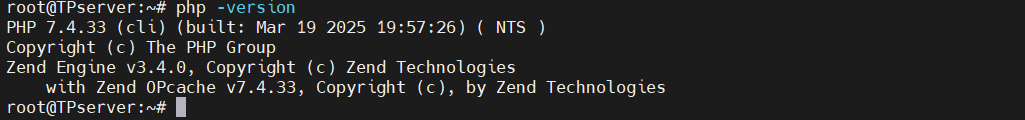




**Instalamos php**



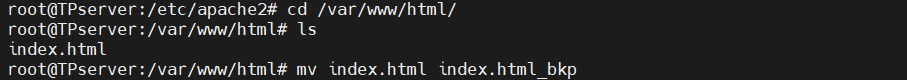
**Verificamos la version**



**Probamos ingresar a través del browser al servidor y verificamos que apache esta andando**



**Nos hacemos un backup del index.html default y copiamos el index.php y logo.png proporcionado por la catedra**





Probamos abrir la página nuevamente y vemos que ya no muestra la página default, esto indica que el servidor apache estaría andando con la nueva página web



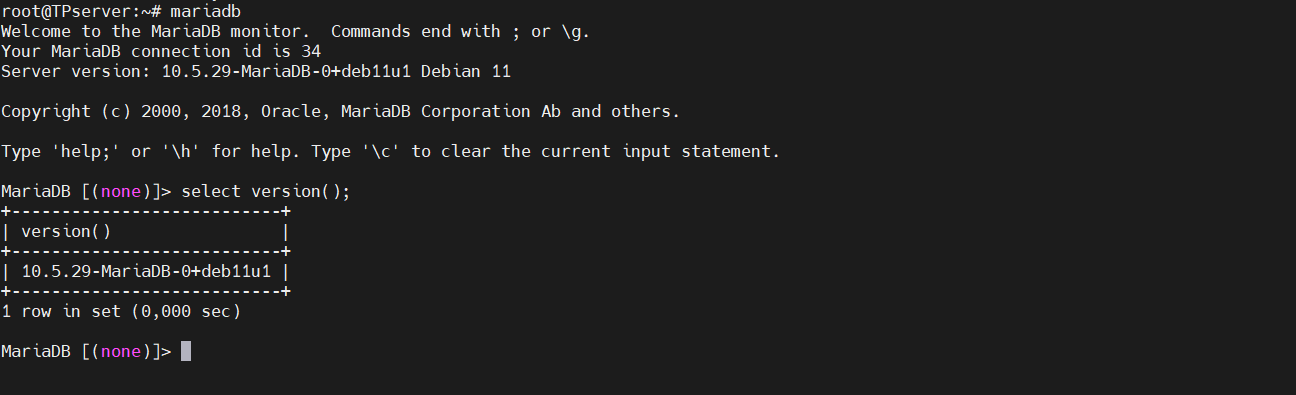
3) **Base de datos:** instalar y configurar **MariaDB**. Cargar, en el motor de base de datos, el script SQL, denominado “**db.sql**”, disponible junto con la máquina virtual en **Blackboard**.

**Nota**: las pruebas de conectividad y acceso al sitio web deben realizarse desde una máquina física u otra máquina en la misma red.

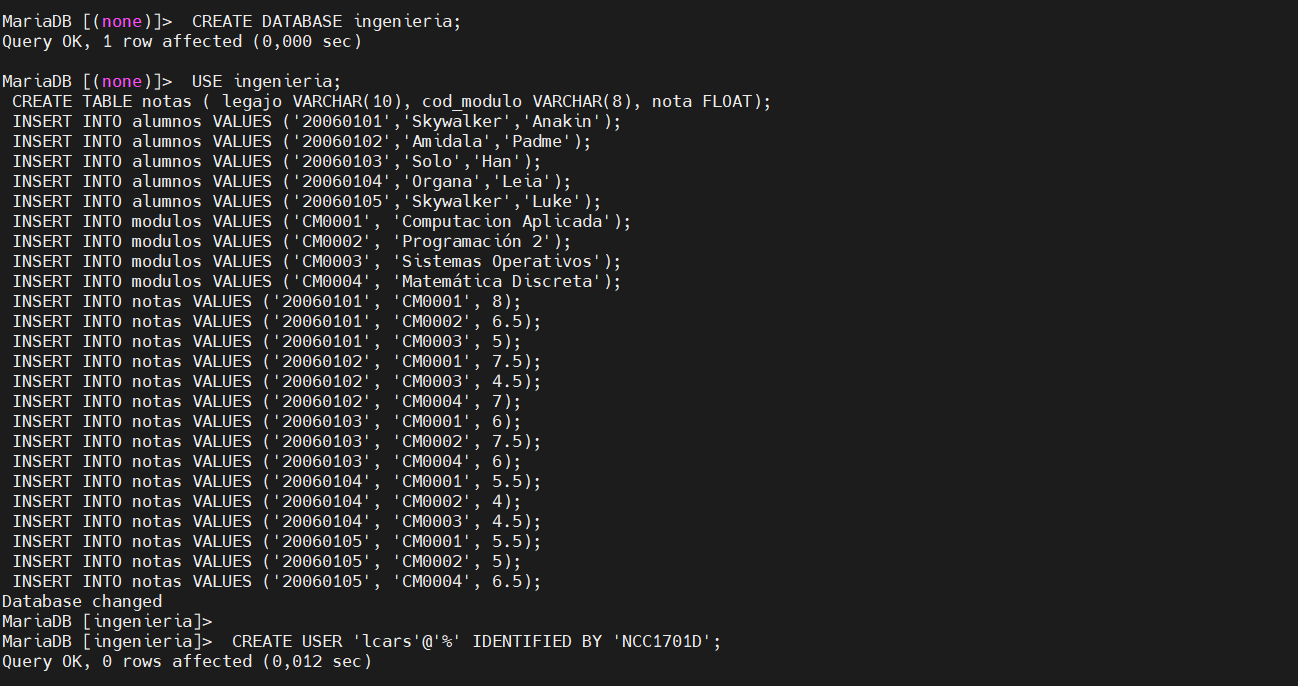
**Instalamos el servidor mariadb**

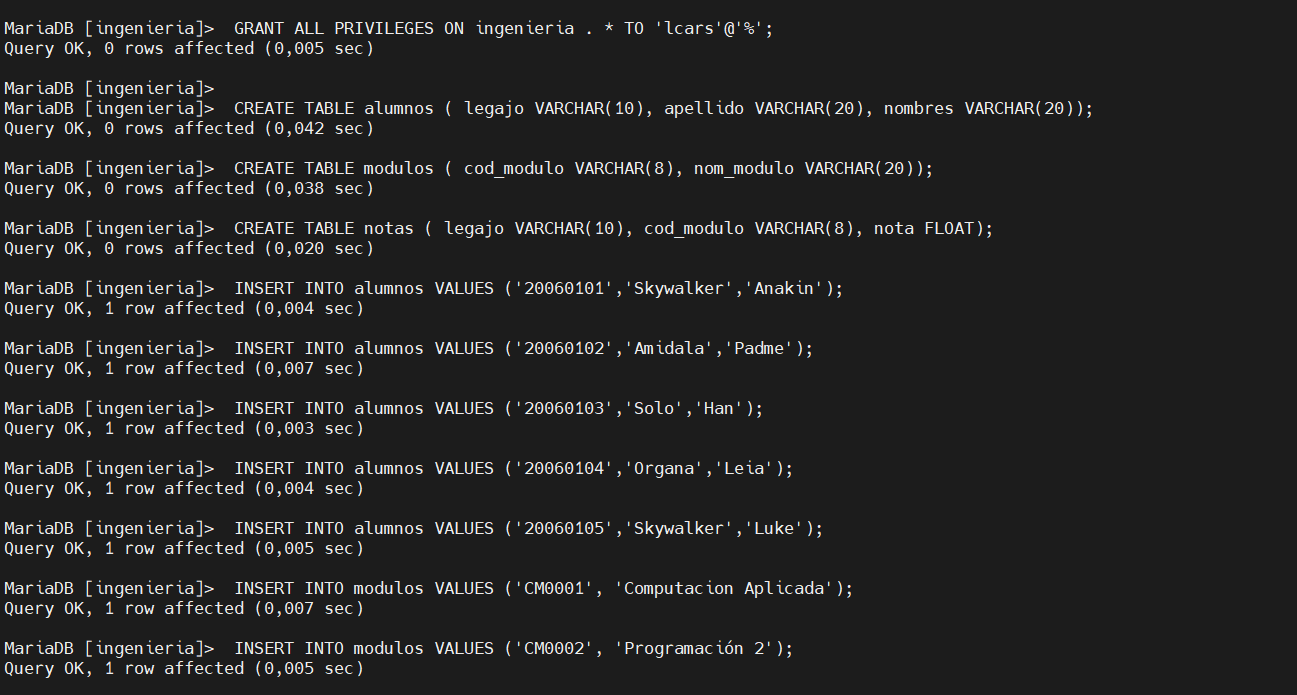


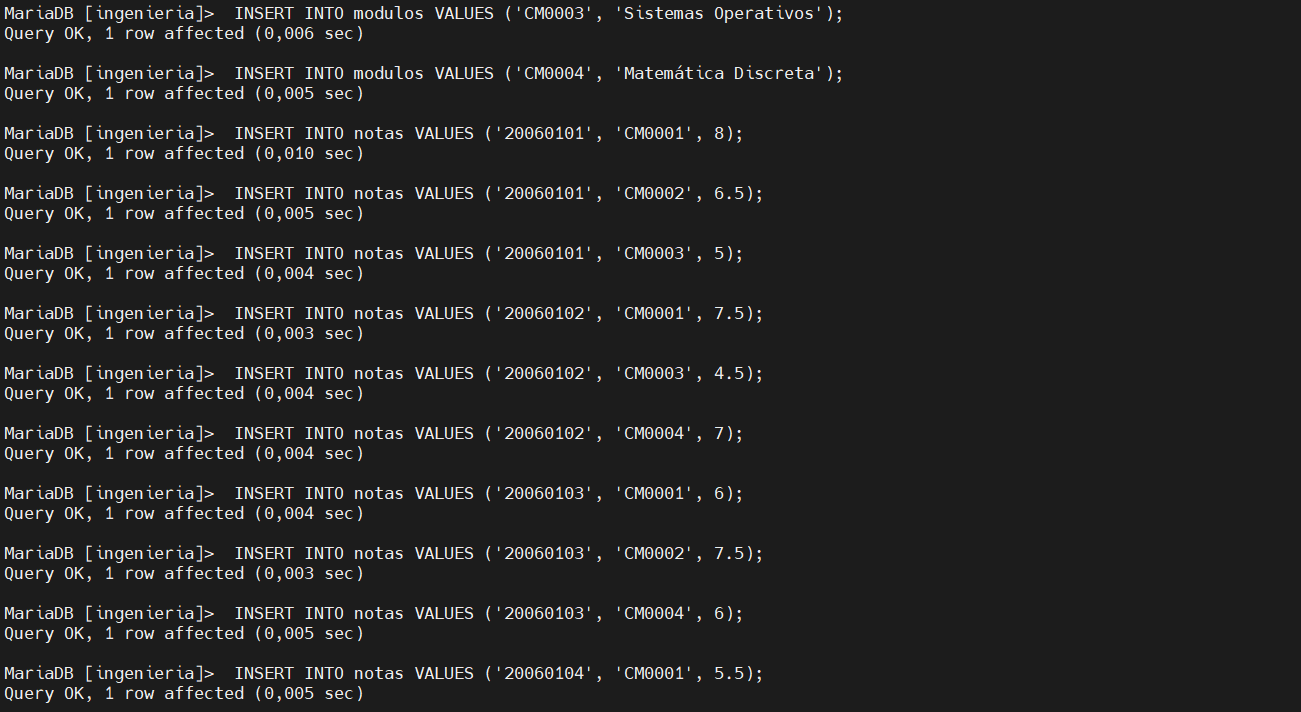
**Probamos que funcione, por ejemplo conectando y consultando la version**

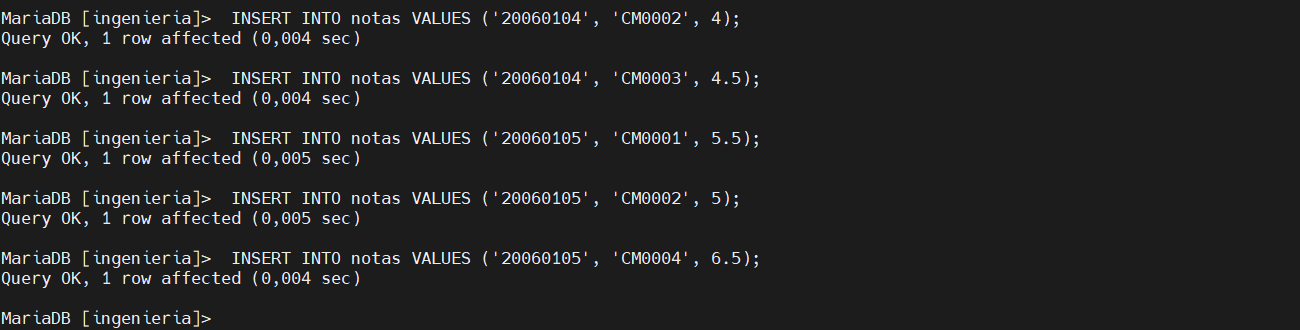


**Corremos los scripts proporcionados por la catedra**

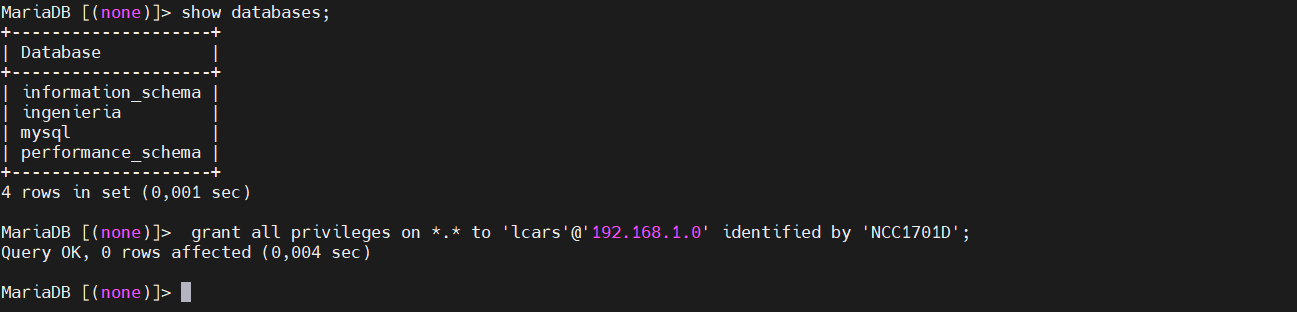




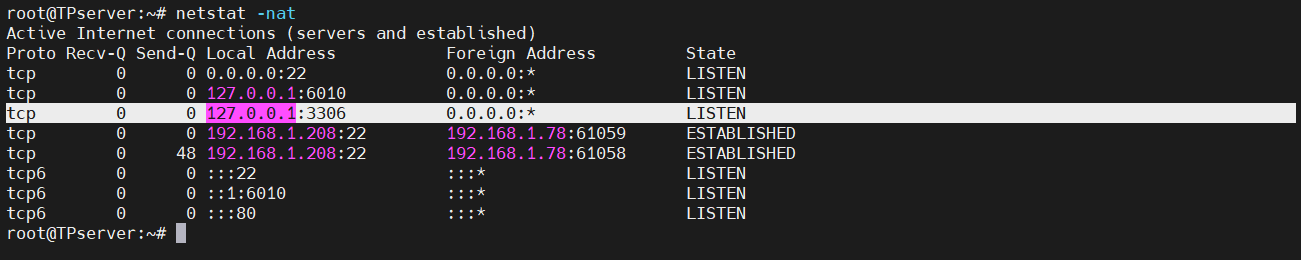




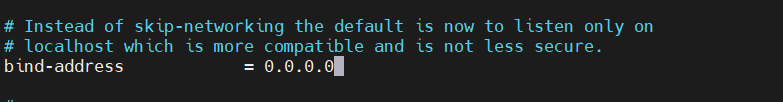
**Verificamos la base creada**



**Vemos que mariadb solo está escuchando por conexiones locales**



**Entonces editamos el archivo de configuración para que escuche para todas las direcciones**



**Reiniciamos el servicio para que tome los cambios**



**Comprobamos nuevamente y vemos que ya esta escuchando en todas las direcciones**



**Intentamos abrir la página y no nos muestra nada**



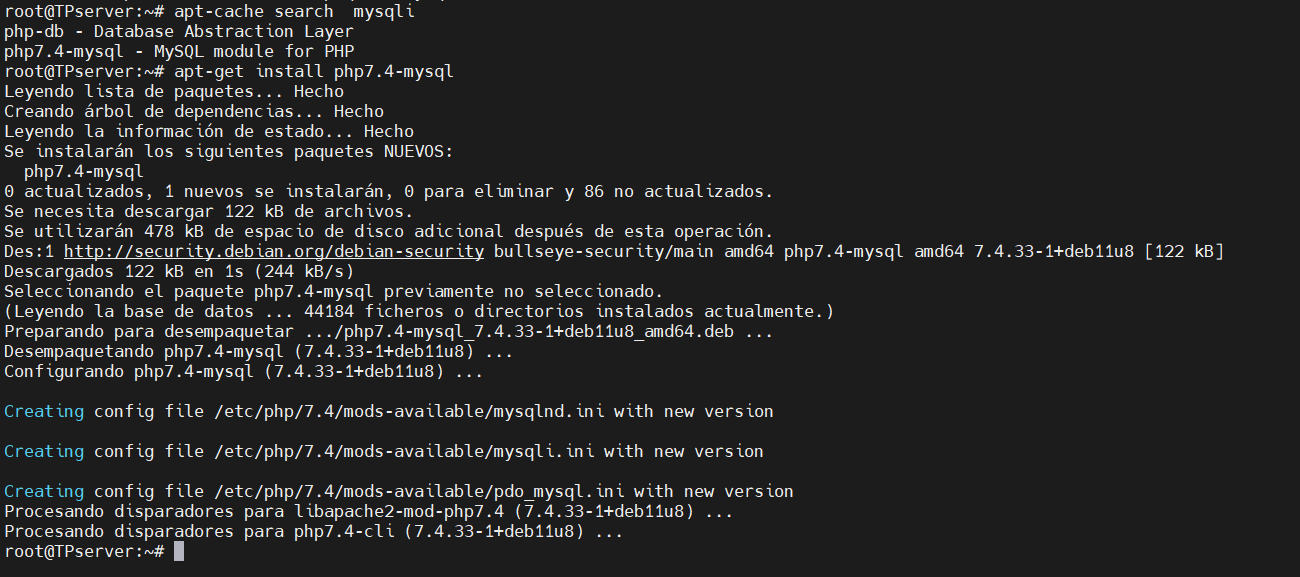
**asique habilitamos el log de errores en el código de php**



**Y al intentar abrir la página de nuevo nos da el siguiente error**



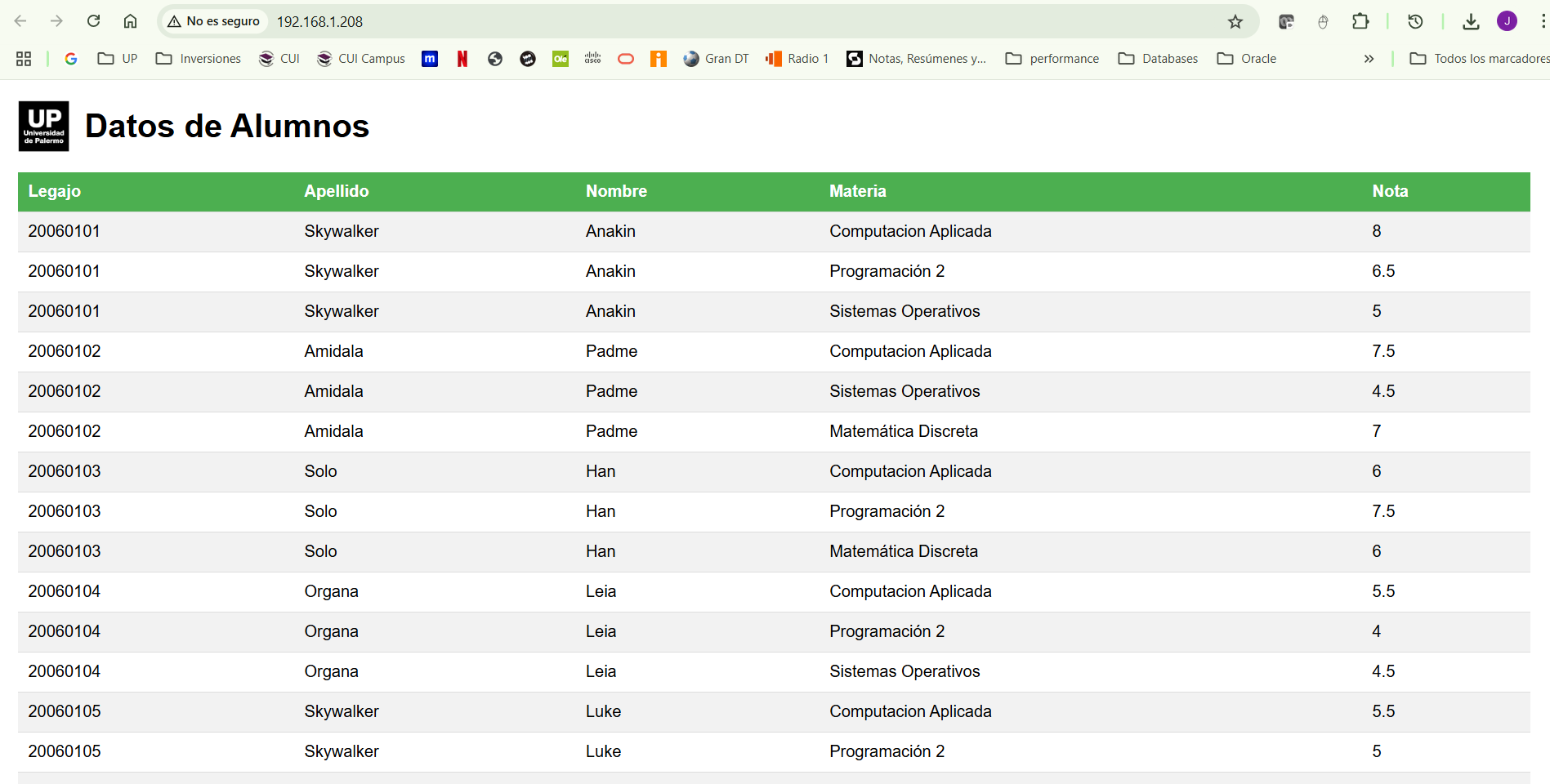
**Instalamos el soporte para mysqli**



**Reiniciamos apache**



**Y volvemos a probar ingresar al sitio y vemos que ya nos lista todos los datos**

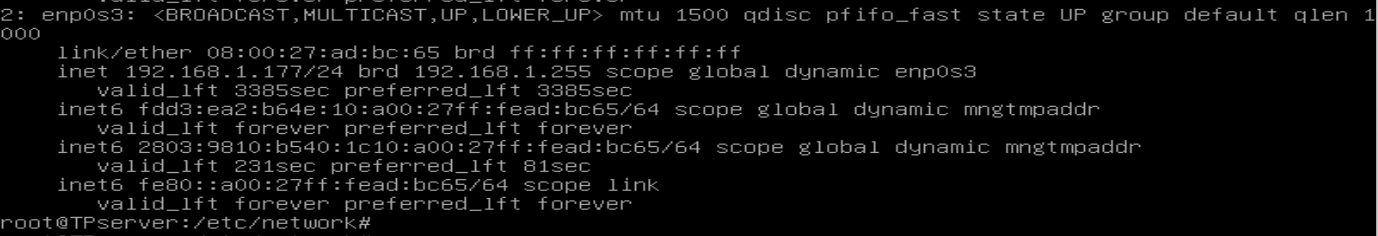


*3) Configuración de Red*

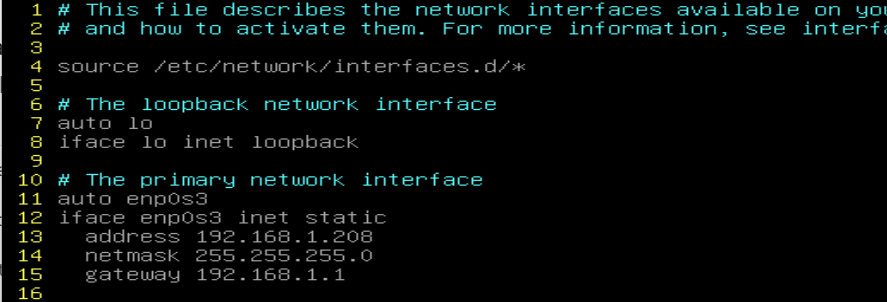
1) Configurar la interfaz de red con una IP estática en el archivo de configuración. La IP debe pertenecer al mismo rango red de la máquina física.

2) El archivo de configuración debe incluir los campos ***ADDRESS***, ***NETMASK*** y ***GATEWAY*.**

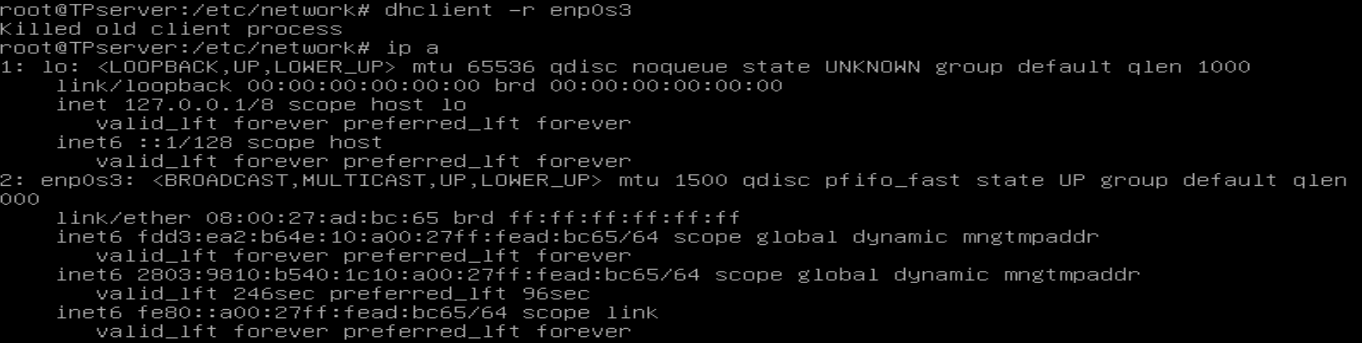
**Verificamos la ip actual**



**Editamos el archivo de configuracion**



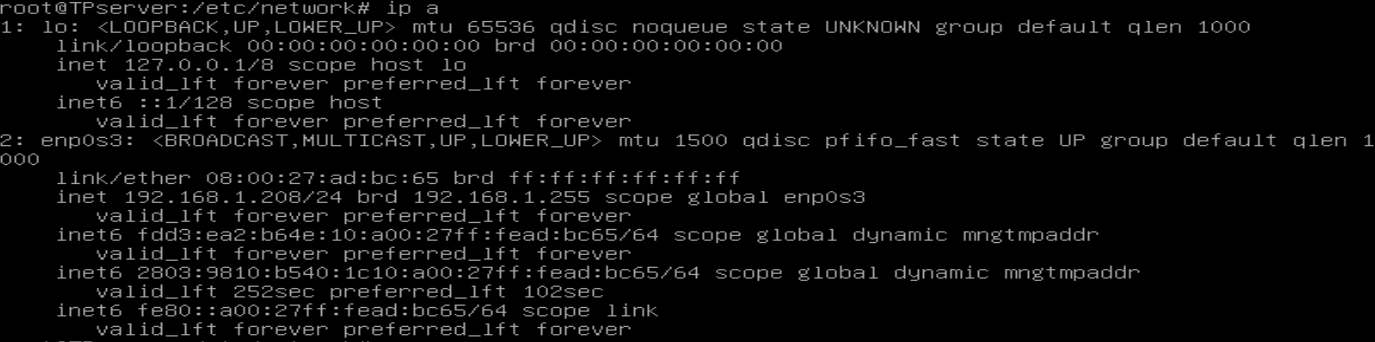
**Liberamos la ip que tenía asignada**



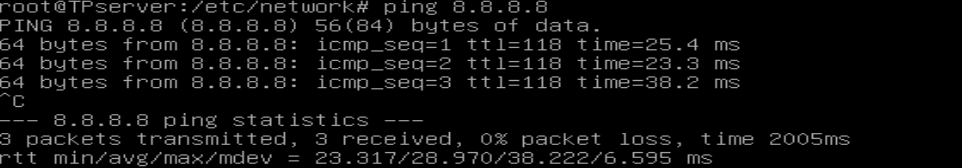
**Reiniciamos el servicio de red**



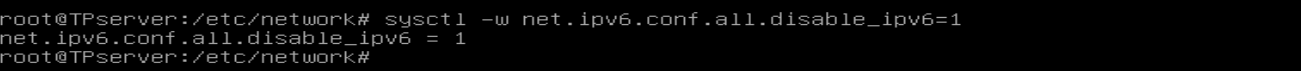
**Verificamos que ahora tiene la ip .208**



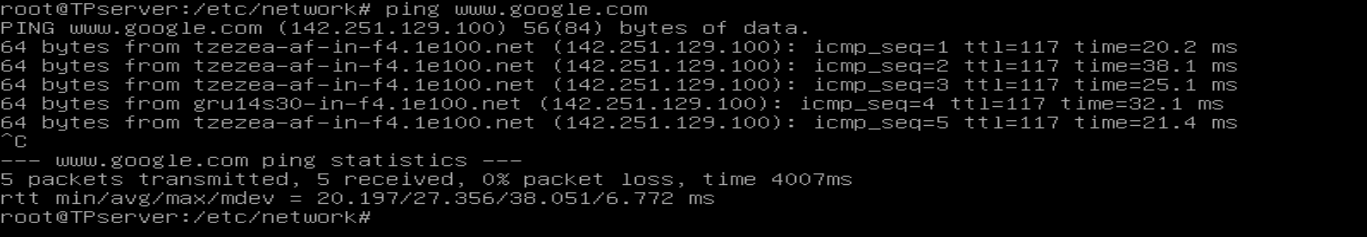
**Probamos la conexión**



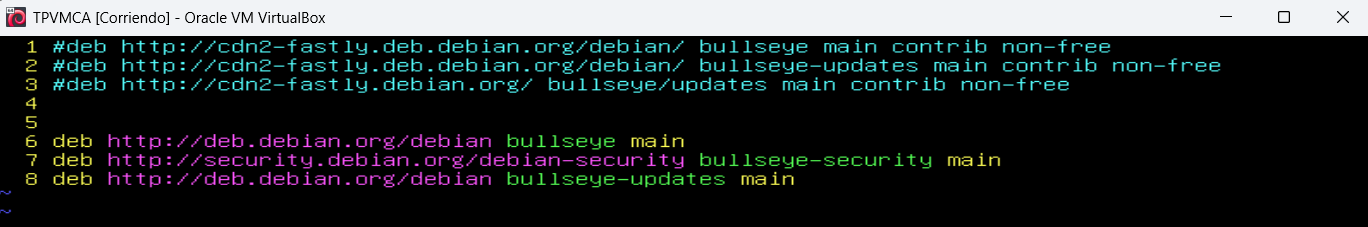
**Vemos que quiere resolver por ipv6 asique lo deshabilitamos**



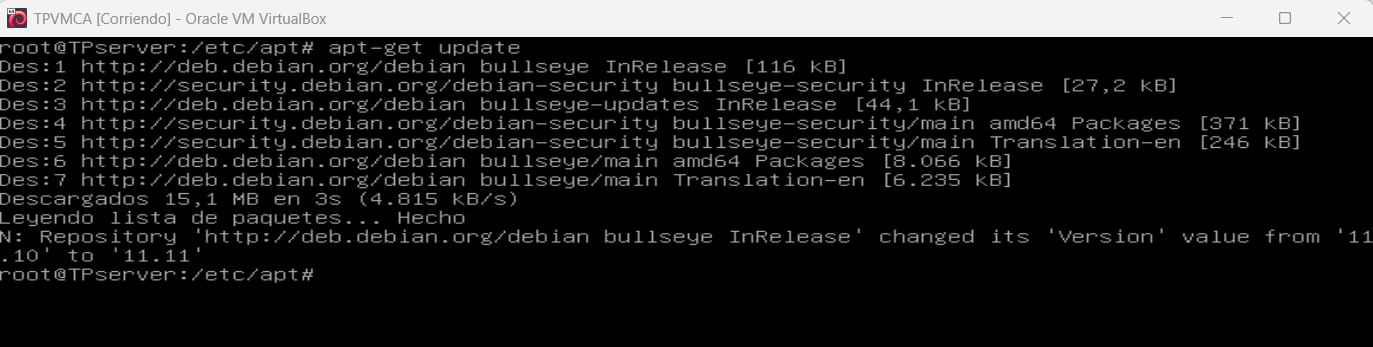
**Ahora vemos que responde por ipv4**



**Agregamos los repositorios**



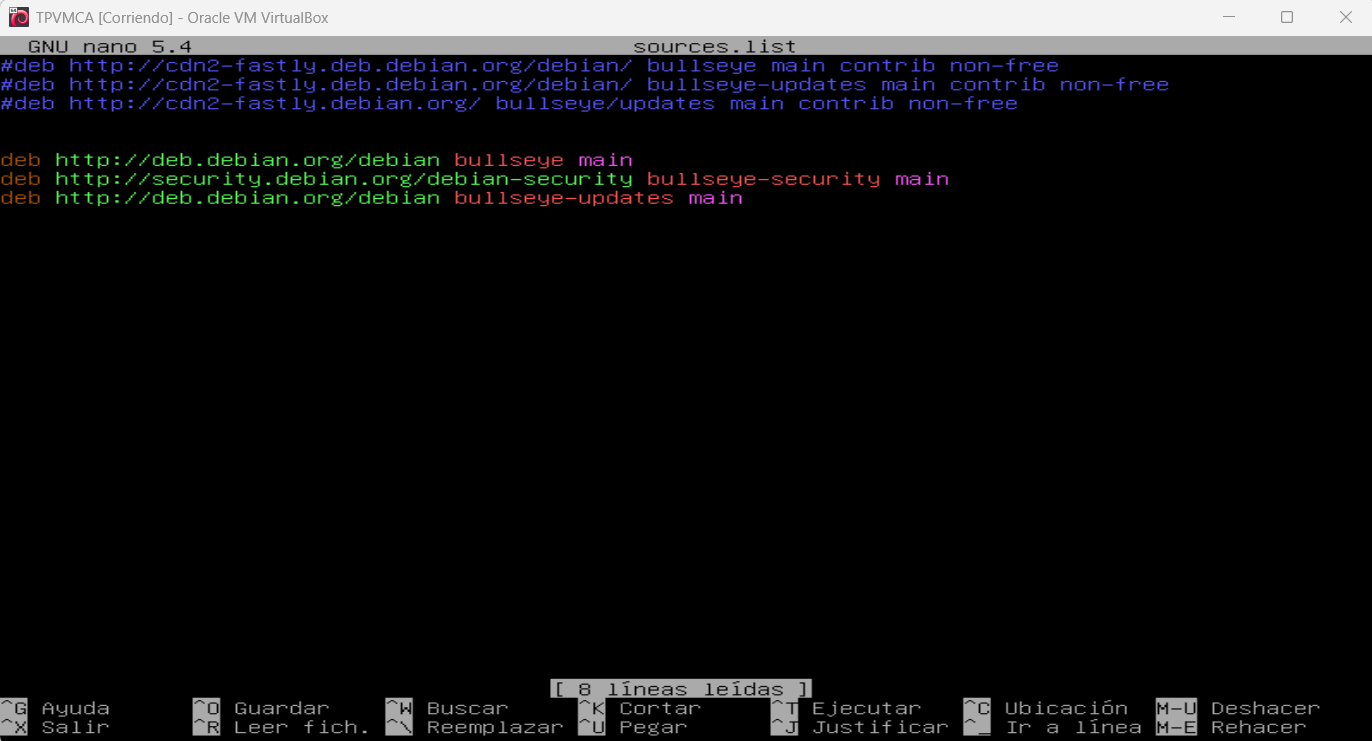
**Y probamos que funcione**





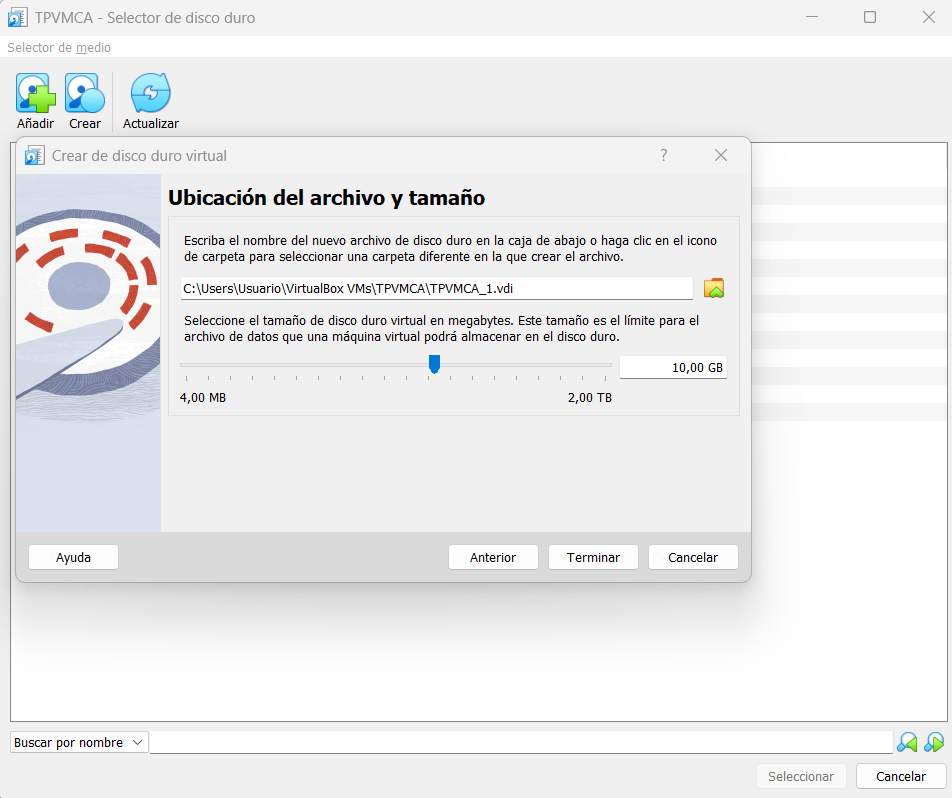
**Abrimos el sourcelist con el nano recién instalado y comprobamos que funciono correctamente**

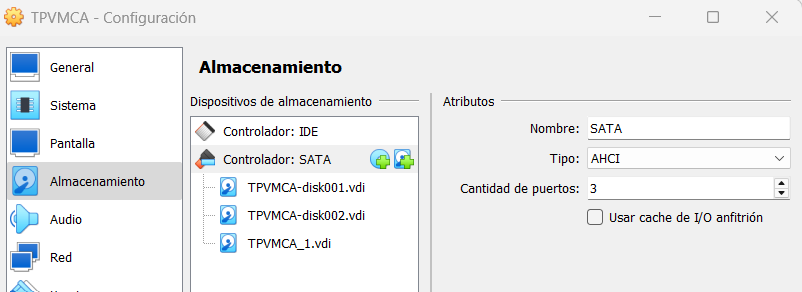




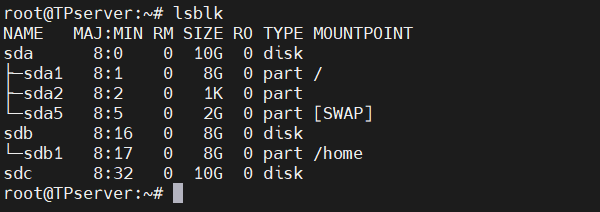
*4) Almacenamiento*

1) Agregar un nuevo disco de **10 GB** adicional a la máquina virtual.





Comprobamos que ya vemos el nuevo disco de 10G

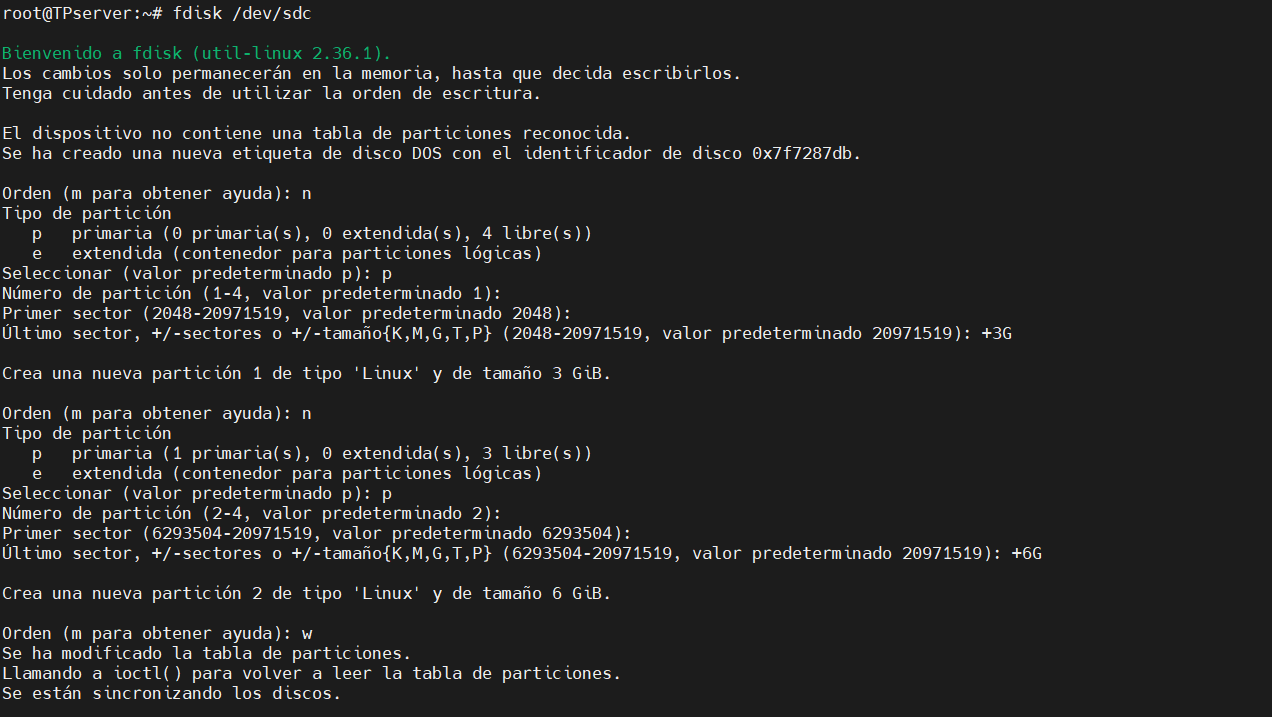


2) Crear dos particiones estándar (tipo 83), con las siguientes capacidades:

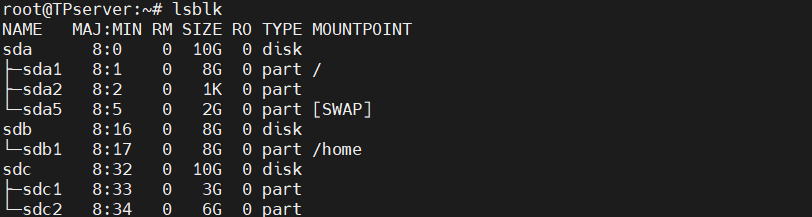
 **/www\_dir: 3 GB**

 **/backup\_dir: 6 GB**

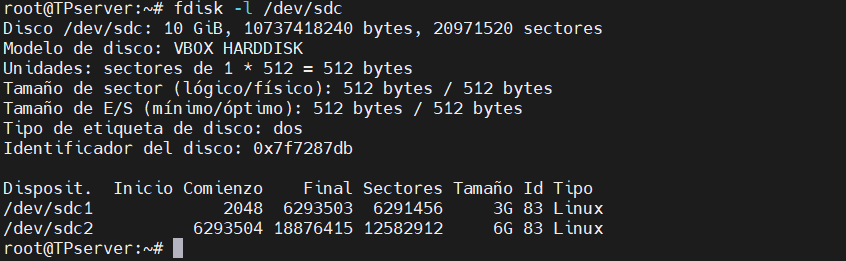
**Creamos las dos particiones con los tamaños indicados en el disco sdc**



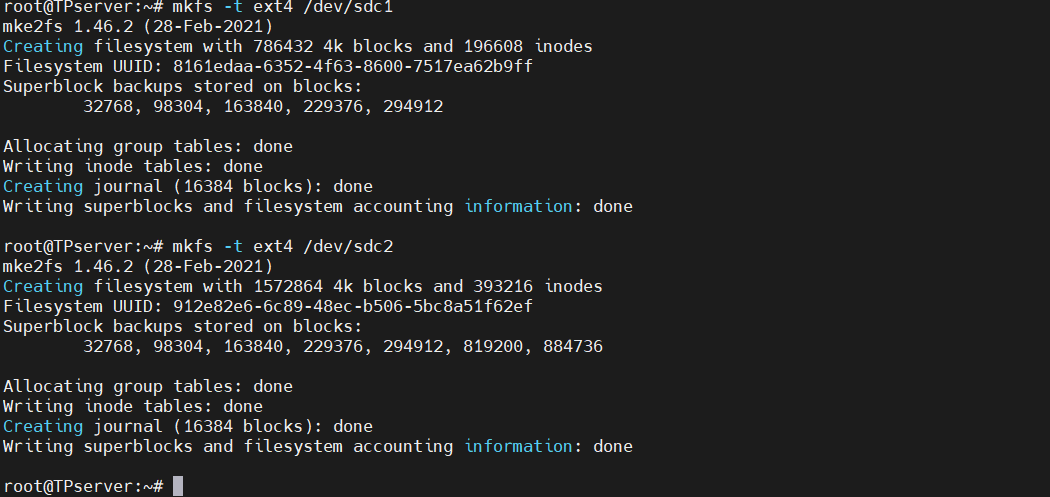
**Verificamos las particiones creadas**



**Con fdisk verificamos que sean de tipo 83**



**Formateamos las particiones creadas**



**Creamos lo puntos de montaje y montamos las particiones**



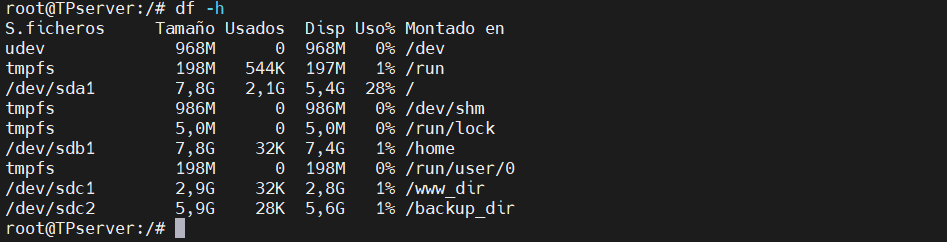


**Asignamos permisos en los directorios(permisos provisorios, después se pueden cambiar dependiendo de las necesidades)**

****

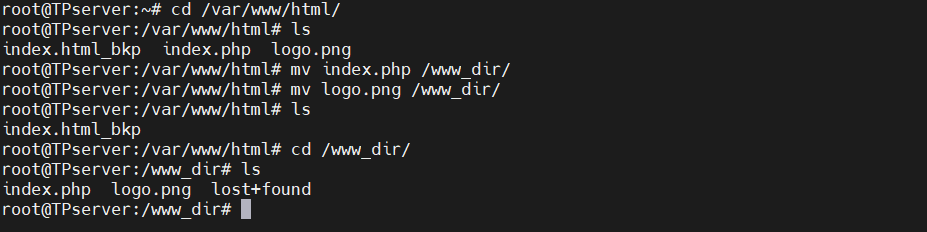


**Verificamos que estén montados**

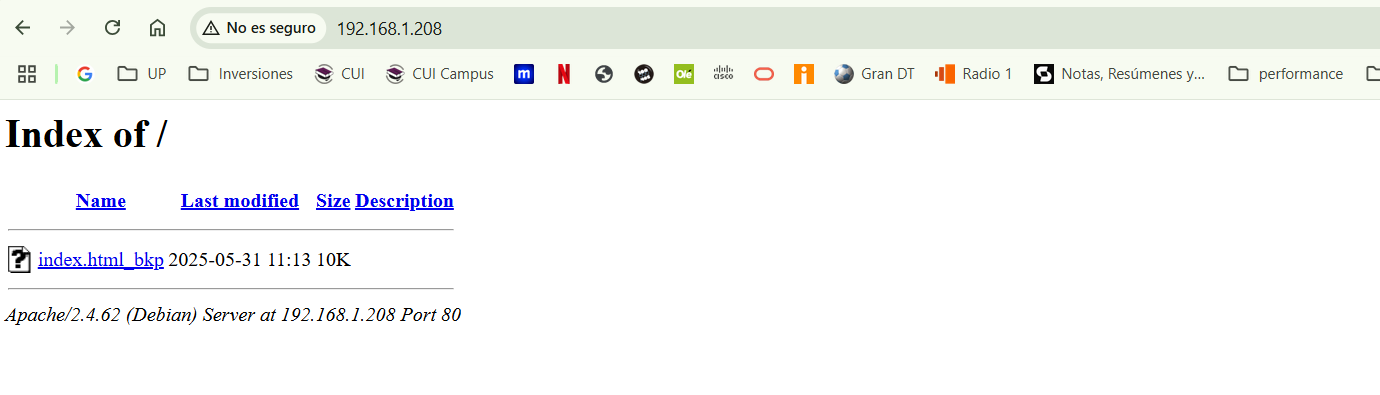


3) Configurar el directorio **/www\_dir** para alojar el archivo **index.php** y **logo.png**. Actualizar el archivo de configuración de Apache para que éste apunte a la nueva ubicación (ver archivos 000-default.conf y apache2).

**Movemos el index.php y logo.png al directorio /www\_dir y actualizamos la configuración de apache**

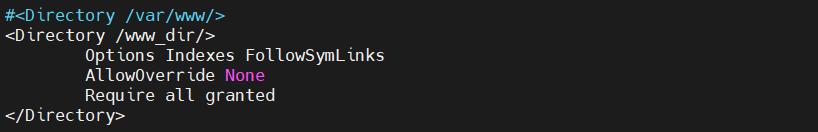


**Probamos ingresar por el browser y no debería mostrar la página porque ya no está el index.php y el .html lo habíamos renombrado con \_bkp**



**Editamos la configuración de apache para que apunte al nuevo directorio**

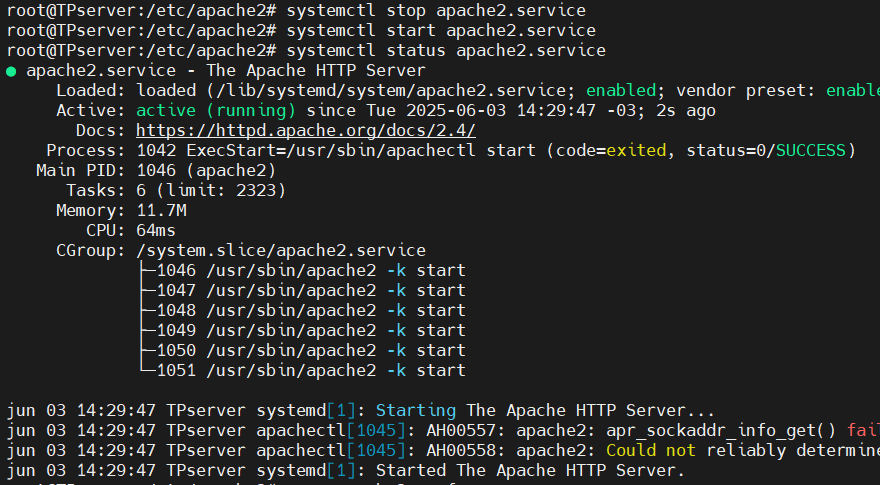
**En apache2.conf reemplazamos la siguiente línea**



En /000-default.conf reemplazamos la siguiente línea



**Reiniciamos el servidor apache**



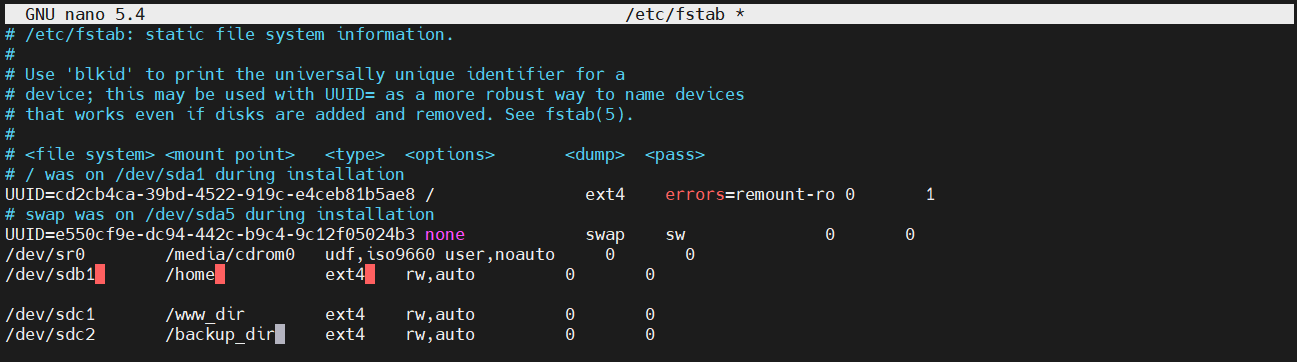
**Probamos acceder a la página nuevamente y funciona correctamente**



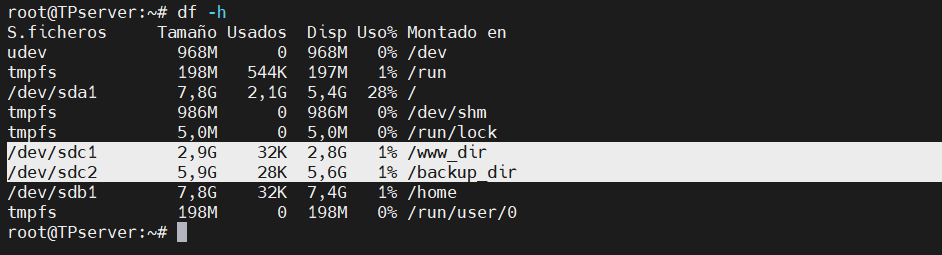
4) Configurar el directorio **/www\_dir** para que se monte automáticamente al iniciar el sistema operativo.

5) Configurar el directorio **/backup\_dir** para que se monte automáticamente al iniciar el sistema operativo.

**Agregamos al fstab las dos particiones (luego reiniciar y verificar que estén montadas)**



**Despues del reinicio vemos que están montadas automáticamente**



**Nota**: se debe crear un archivo en /proc llamado “particion”, y redirigir el contenido del archivo “partitions” ubicado en /proc (el archivo original es efímero y se pierde al apagar la máquina).

*5) Backup*

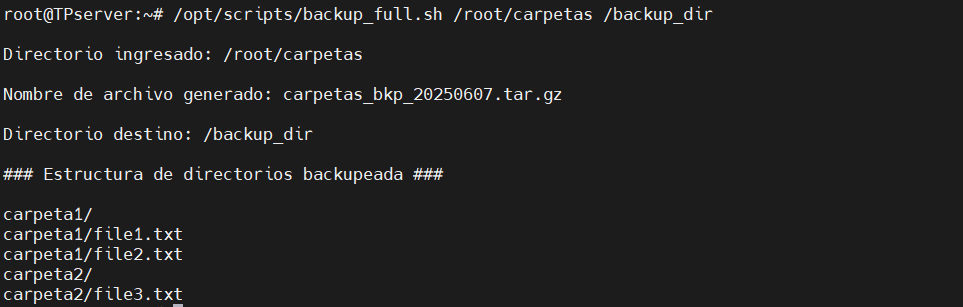
1) Desarrollar un script de *backup* denominado “**backup\_full.sh**”, y guardarlo en **/opt/scripts.**







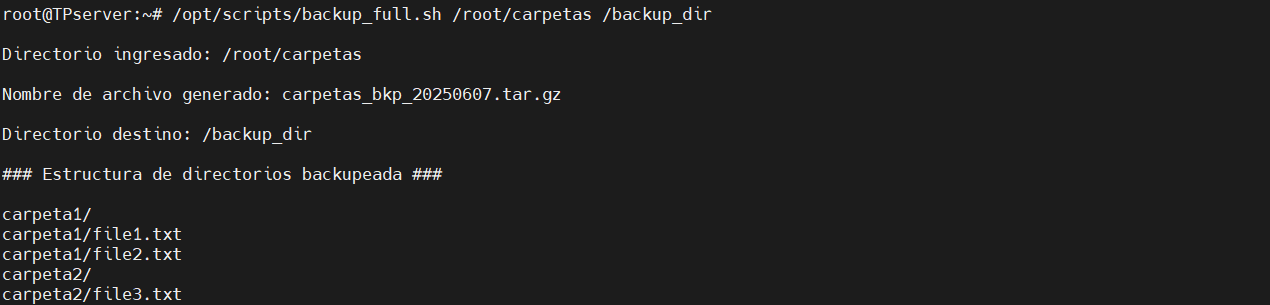
2) El script debe backupear los directorios indicados con nombres que incluyan la fecha en formato **ANSI (YYYMMDD)**. Por ejemplo, para /**var/log**, el archivo generado debería llamarse “**log\_bkp\_20240302.tar.gz**”.



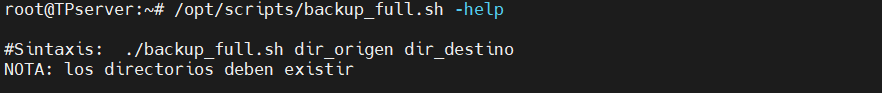
3) Los backups generados deben almacenarse en la partición que tiene montado el directorio /**backup\_dir.**



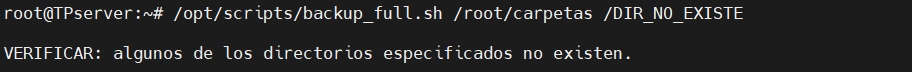
4) El script debe aceptar argumentos como **origen** (lo que se va a backupear) y **destino** (dónde se va a backupear).



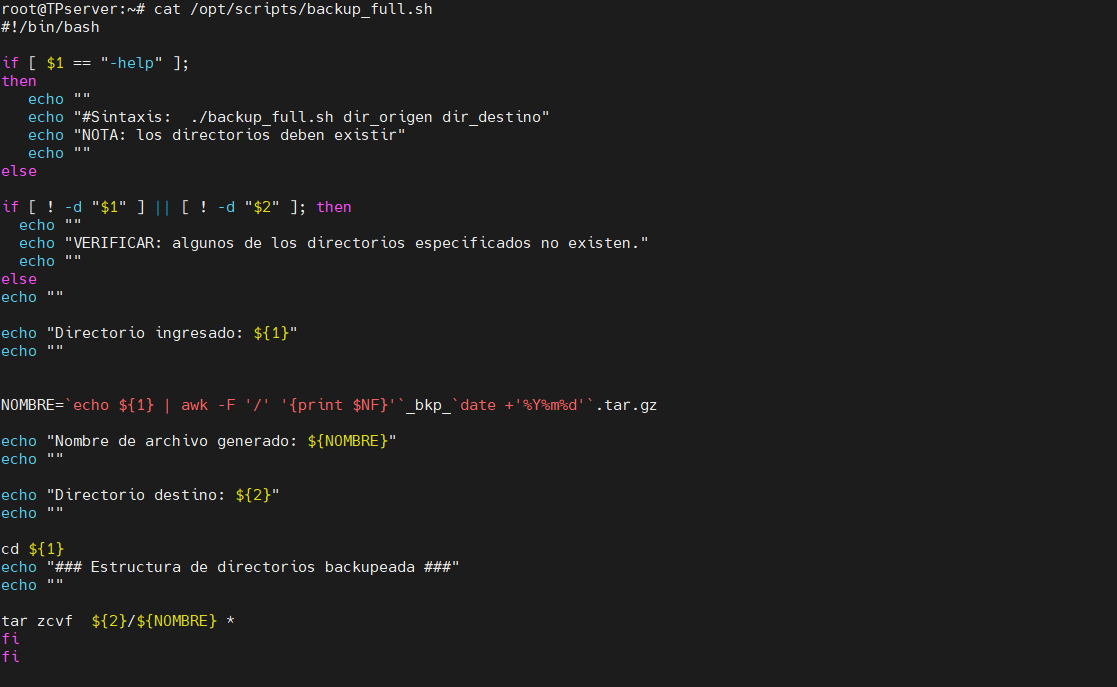
5) El script debe incluir una opción de ayuda (**-help**), para guiar al usuario en el uso del script.



6) El script **debe validar** que los sistemas de archivos de **origen** y **destino** estén disponibles antes de ejecutar el backup.



SCRIPT



root@TPserver:~# cat /opt/scripts/backup\_full.sh

#!/bin/bash

if [ $1 == "-help" ];

then

echo ""

echo "#Sintaxis: ./backup\_full.sh dir\_origen dir\_destino"

echo "NOTA: los directorios deben existir"

echo ""

else

if [ ! -d "$1" ] || [ ! -d "$2" ]; then

echo ""

echo "VERIFICAR: algunos de los directorios especificados no existen."

echo ""

else

echo ""

echo "Directorio ingresado: ${1}"

echo ""

NOMBRE=`echo ${1} | awk -F '/' '{print $NF}'`\_bkp\_`date +'%Y%m%d'`.tar.gz

echo "Nombre de archivo generado: ${NOMBRE}"

echo ""

echo "Directorio destino: ${2}"

echo ""

cd ${1}

echo "### Estructura de directorios backupeada ###"

echo ""

tar zcvf ${2}/${NOMBRE} \*

fi

fi

7) El script debe ser incluido en un calendario de tareas para correr automáticamente:

 **TODOS LOS DÍAS a las 00:00 hs: Backupear “/var/logs”**

 **LUNES, MIÉRCOLES, VIERNES a las 23:00 hs: Backupear “/www\_dir”**

**Agregamos al crontab la planificación indicada**

