

ADMINISTRACIÓN DE SISTEMAS

PRÁCTICA FINAL 2024

RENOVACIÓN DIAWEB



**VNiVERSiDAD
DE SALAMANCA**

Álvaro García Sánchez - 70924450V

agarsan@usal.es

ÍNDICE

1. INTRODUCCIÓN.....	3
2. INSTALACIÓN DEL ENTORNO.....	3
3. CONFIGURACIÓN INICIAL.....	4
4. SERVIDOR SSH.....	6
5. SERVIDOR WEB.....	7
6. GESTIÓN DE USUARIOS.....	13
7. DIRECTORIO SKEL.....	14
8. BASE DE DATOS MARIADB.....	15
9. CONFIGURACIÓN DE QUOTAS.....	17
10. ELIMINAR USUARIOS SIN CONFIRMAR.....	19
11. BACKUPS.....	19
12. AVISO ROOT.....	21
13. SERVICIO CORREO ELECTRÓNICO.....	22
14. SERVICIO SFTP.....	23
15. MONITORIZACIÓN DEL SISTEMA.....	23
16. SCRIPTS.....	24
16.1. HTML.....	25
17. BLOG PERSONAL.....	29

1. INTRODUCCIÓN

Durante este informe, explicaré en detalle el proceso completo para crear un servidor funcional, en mi caso soportado sobre un sistema operativo Debian 12 (LINUX), que permitirá realizar las funciones de gestión del sistema por parte del administrador, así como prestar los servicios propuestos en el enunciado de la práctica a los usuarios con el objetivo de crear un entorno fiable, eficiente y seguro.

El servidor permite administrar una plataforma similar a la existente diaweb en la que usuarios alumno y usuarios profesor, tienen acceso a una serie de servicios o documentos especificados en el enunciado.

La empresa ficticia que he creado para realizar la práctica se llama Avanzall y por tanto, el servidor, la máquina y el usuario administrador tendrán el mismo nombre o similar.

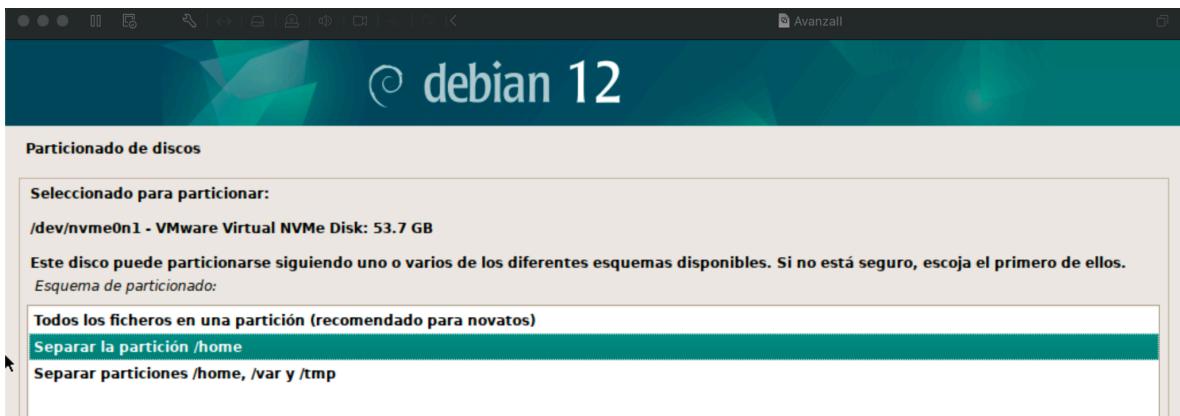
2. INSTALACIÓN DEL ENTORNO

Primero que nada, debemos instalar el sistema operativo sobre el que vamos a trabajar, como comenté en el punto anterior, en mi caso usaré la última versión de Debian (Debian 12), descargado desde el sitio oficial completamente gratis.

El siguiente paso es montar esta imagen de Debian sobre un servicio de virtualización o de creación de máquinas virtuales, en mi caso estaré usando VMware.

Para la instalación o creación de esta máquina virtual, elegiremos la versión de instalación gráfica y seleccionaremos y rellenaremos las opciones que se nos presentan por pantalla (idioma, teclado, huso horario, usuarios, contraseñas, nombre de la máquina, etc).

Un punto interesante a destacar a la hora de realizar la instalación del sistema operativo, es que decidí separar la partición **/home** del resto de directorios ya que, de este modo, permitirá al administrador realizar una gestión más eficiente al estar desacoplados los datos de los usuarios de los del servidor. Este método no sólo dota al servidor de una mayor seguridad si no que también lo vuelve más eficiente ya que, en el momento de crear nuevos usuarios y asignarles cuotas, estas únicamente se aplicarán a los directorios **/home** de cada usuario.



3. CONFIGURACIÓN INICIAL

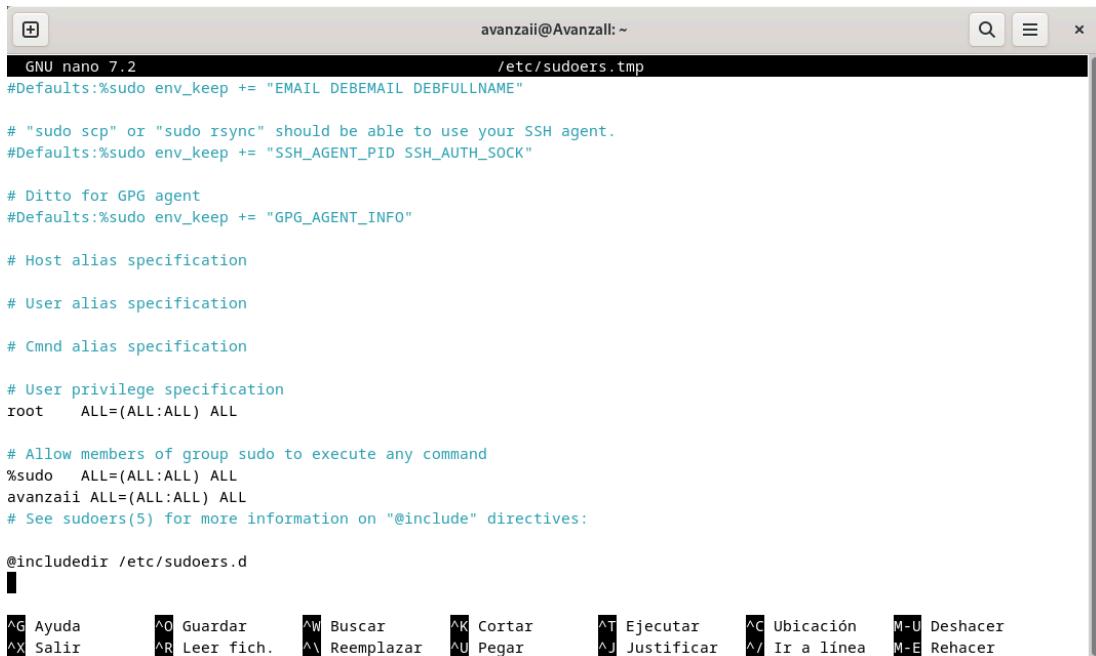
Una vez completada la instalación del sistema, abrimos una terminal y ejecutamos los siguientes comandos para hacer una actualización integral del sistema:

- Accedemos al superusuario con: **su -**
- Ejecutamos: **apt update** y **apt upgrade**

```
avanzaii@AvanzaII:~$ su -
Contraseña:
root@AvanzaII:~# apt update
Obj:1 http://deb.debian.org/debian bookworm InRelease
Obj:2 http://deb.debian.org/debian bookworm-updates InRelease
Obj:3 http://security.debian.org/debian-security bookworm-security InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 23 paquetes. Ejecute «apt list --upgradable» para verlos.
root@AvanzaII:~# apt upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Utilice «apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes NUEVOS:
  linux-image-6.1.0-21-arm64
Se actualizarán los siguientes paquetes:
  gir1.2-javascriptcoregtk-4.0 gir1.2-javascriptcoregtk-4.1 gir1.2-webkit2-4.0 gir1.2-webkit2-4.1
  gnome-shell gnome-shell-common gnome-shell-extension-prefs less libc-bin libc-110n libc6 libdav1d6
  libglib2.0-0 libglib2.0-bin libglib2.0-data libjavascriptcoregtk-4.0-18 libjavascriptcoregtk-4.1-0
  libjavascriptcoregtk-6.0-1 libwebkit2gtk-4.0-37 libwebkit2gtk-4.1-0 libwebkitgtk-6.0-4 linux-image-arm64
  locales
23 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 0 B/151 MB de archivos.
```

Para llevar a cabo tareas de administración dentro del sistema, es preferible crear un usuario con permisos distintos del propio root, para ello, utilizaremos el usuario principal que hemos creado durante la instalación (avanzaii). El proceso para otorgarle permisos a dicho usuario es el siguiente:

- Accedemos a la terminal en modo root: **su -**
- Añadimos al usuario al grupo sudo: **usermod -aG sudo avanzaii**
- Abrimos el fichero sudoers: **visudo**
- Añadimos la siguiente línea: **avanzaii ALL=(ALL:ALL) ALL**



```

GNU nano 7.2 avanzaii@Avanzall: ~
/etc/sudoers.tmp

#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
avanzaii ALL=(ALL:ALL) ALL
# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d

```

^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar ^C Ubicación M-U Deshacer
 ^X Salir ^R Leer fich. ^V Reemplazar ^U Pegar ^J Justificar ^/ Ir a linea M-E Rehacer

Siguiendo con la configuración del sistema, instalamos el paquete propio de Debian “build essential”, que instalará una serie de paquetes con aplicaciones básicas que necesitaremos, como por ejemplo, gcc, g++, make... Entre otras muchas. De esta forma hacemos una instalación más rápida:

apt-get install build-essential

Instalamos también el paquete de herramientas net-tools: **apt install net-tools**

Finalmente, podemos personalizar un poco el fichero de inicio de sesión, message of the day (motd) para personalizarlo un poco. Para ello utilizamos la herramienta toilet y la instalamos con **apt install toilet**.

→ **toilet –metal bienvenidos a avanzaii > /etc/motd**

4. SERVIDOR SSH

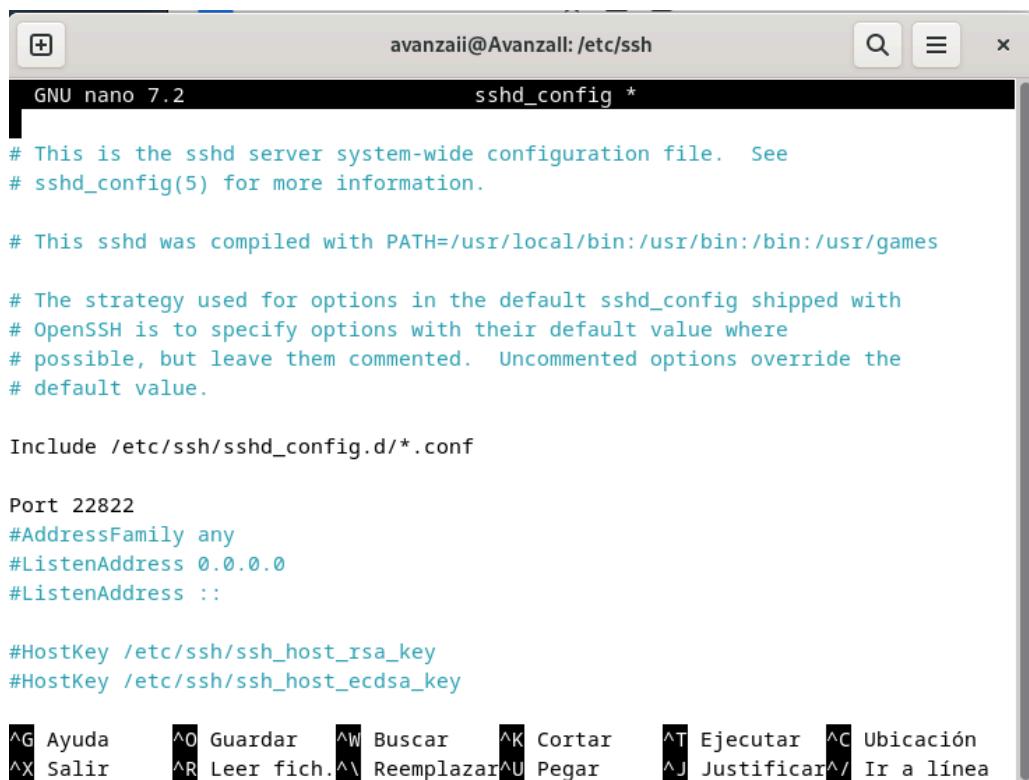
He decidido usar el protocolo de administración remota SSH ya que permite autenticar un usuario de manera remota y permite realizar un cifrado completo de los datos intercambiados entre ambos extremos cliente-servidor.

Lo instalamos con: **apt install openssh-server**

Una vez instalado el servidor, realizaremos una serie de configuraciones para incrementar el nivel de seguridad. Todas estas modificaciones se realizarán en el fichero de configuración del servidor **/etc/ssh/sshd_config**

- Cambiamos el puerto del servicio por defecto (22), al **22822**, por ejemplo.
Podemos usar cualquier puerto que no esté asignado por defecto siempre y cuando sea un número superior a 1000 (65355 o FFFF en hexadecimal es el número máximo de puertos en un sistema linux)
- Modificamos el tiempo de gracia para que el usuario tenga un tiempo máximo de 1 min para que el usuario introduzca sus credenciales de acceso al sistema.
- Cambiamos el número máximo de intentos que puede utilizar un usuario para autenticarse en el sistema, pasamos de 6 por defecto, a 3.

Descomentamos todas las líneas modificadas, guardamos el fichero y reiniciamos el servicio con la orden **systemctl restart ssh.service**



```
avanzaii@Avanzall: /etc/ssh
GNU nano 7.2          sshd_config *

# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

Port 22822
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

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

^G Ayuda      ^O Guardar    ^W Buscar    ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich.  ^\ Reemplazar ^U Pegar      ^J Justificar ^/ Ir a línea
```

```

GNU nano 7.2                               sshd_config *

#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 1m
#PermitRootLogin prohibit-password
#StrictModes yes
MaxAuthTries 3

^G Ayuda      ^O Guardar     ^W Buscar      ^K Cortar      ^T Ejecutar    ^C Ubicación
^X Salir      ^R Leer fich.  ^\ Reemplazar  ^U Pegar       ^J Justificar ^/ Ir a línea

```

5. SERVIDOR WEB

Para mostrar el contenido del servidor y ofrecer los distintos servicios a los usuarios, he decidido utilizar apache2 como servidor web de fácil uso y código abierto, de este modo, podré ofrecer el contenido web a través de internet.

- Instalamos con: **apt-get install apache2**
- Para mayor seguridad, utilizaremos suExec de apache: **apt install apache2-suexec-custom**

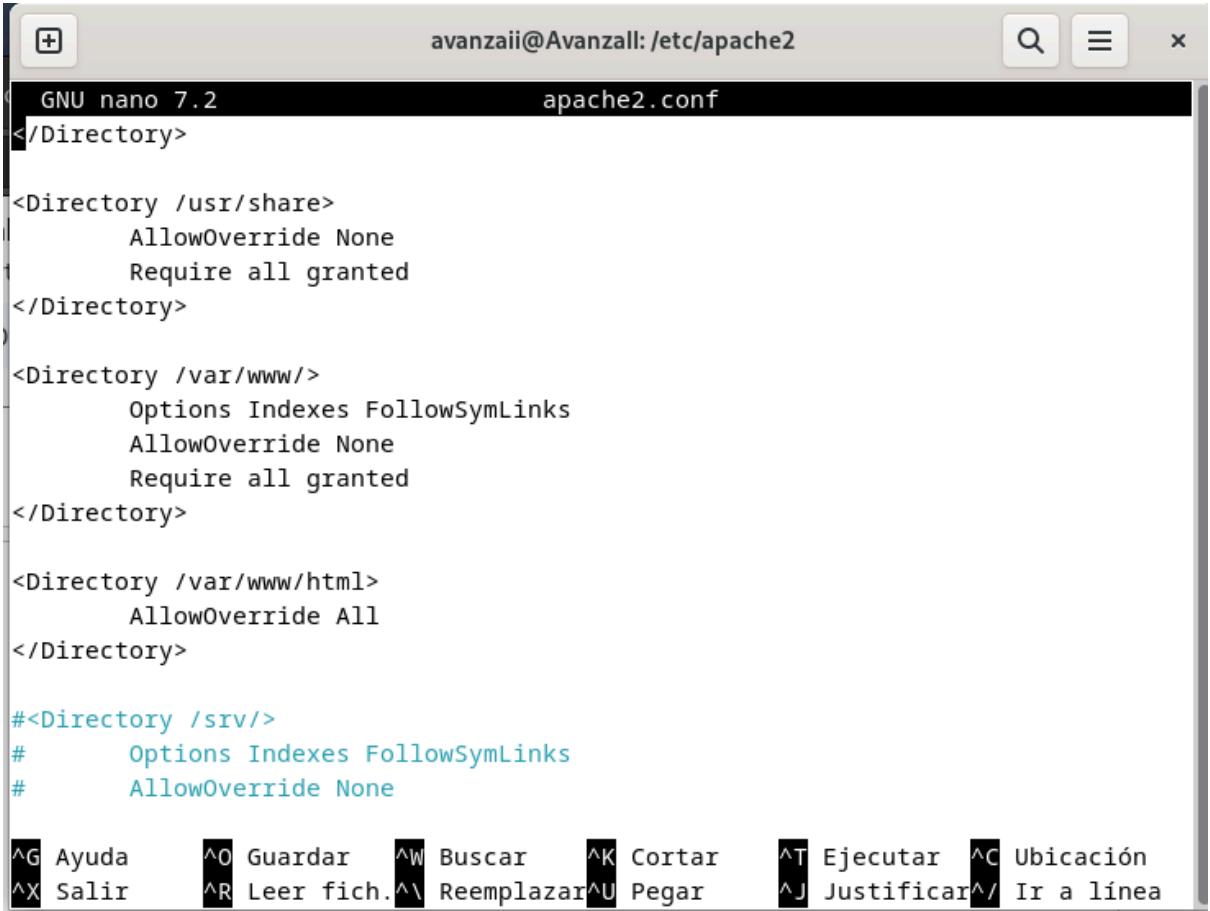
Suexec se utiliza por el servidor HTTP Apache para cambiar a otro usuario antes de ejecutar programas CGI. Para lograr esto, debe ejecutarse como root. Dado que el demonio HTTP normalmente no se ejecuta como root, el ejecutable suexec necesita el bit setuid establecido y debe ser propiedad de root.

Después de la instalación, añadiremos un certificado ssl para encriptar los datos SSL a través del protocolo seguro de transferencia de hipertexto (HTTPS). Dado que será un certificado autofirmado, el navegador no lo tendrá en cuenta para comunicar una conexión segura, pero el procedimiento para realizarlo es el siguiente:

- Instalamos openSSL que nos dará el certificado SSL de manera gratuita: **apt install openssl**
- Nos aseguramos de que el soporte para SSL/TLS está activo: **a2enmod ssl**

- Permitimos realizar un DNS con la configuración del SSL: **a2enmod rewrite**
- Reiniciamos el servicio para que se guarden todos los cambios: **systemctl restart apache2**

Vamos ahora a modificar el fichero de configuración de apache para añadir unas líneas que nos permitirán modificar el directorio libremente. Este fichero de configuración se encuentra en **/etc/apache2/apache2.conf**



```

GNU nano 7.2                               apache2.conf
</Directory>

<Directory /usr/share>
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/html>
    AllowOverride All
</Directory>

#<Directory /srv/>
#    Options Indexes FollowSymLinks
#    AllowOverride None

^G Ayuda      ^O Guardar     ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich.  ^\ Reemplazar ^U Pegar      ^J Justificar ^/ Ir a línea

```

Creamos ahora un directorio en la misma carpeta de apache2 en la que vamos a alojar el certificado SSL, para ello:

- **mkdir /etc/apache2/cert**
- **cd /etc/apache2/cert**

Finalmente, deberemos generar una clave “key” para que el certificado sea válido, para ello, hacemos uso de la propia herramienta openssl:

- **openssl req -new -newkey rsa:4096 -x509 -sha256 -days 365 -nodes -out apache-certificate.crt -keyout apache.key**

Una vez creado tanto certificado como clave, deberemos modificar el fichero de configuración `/etc/apache2/sites-enabled/000-default.conf` para que el servidor web funcione con el certificado SSL que hemos generado activando y linkeando las opciones con los datos de nuestro dominio:

```
GNU nano 7.2                               000-default.conf
<VirtualHost *:80>
    RewriteEngine On
    RewriteCond %{HTTPS} !=on
    RewriteRule ^/?(.*) https:// %{SERVER_NAME}/%1 [R=301,L]
</VirtualHost>

<VirtualHost *:443>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    SSLEngine on
    SSLCertificateFile /etc/apache2/cert/apache-certificate.crt
    SSLCertificateKeyFile /etc/apache2/cert/apache.key
</VirtualHost>

[ El fichero «000-default.conf» no es de escritura ]
```

Reiniciamos apache2 y accedemos al navegador o bien con la ip, o bien con el nombre establecido en el SSL (<https://avanzaii>).

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|--- apache2.conf
|   `--- ports.conf
```

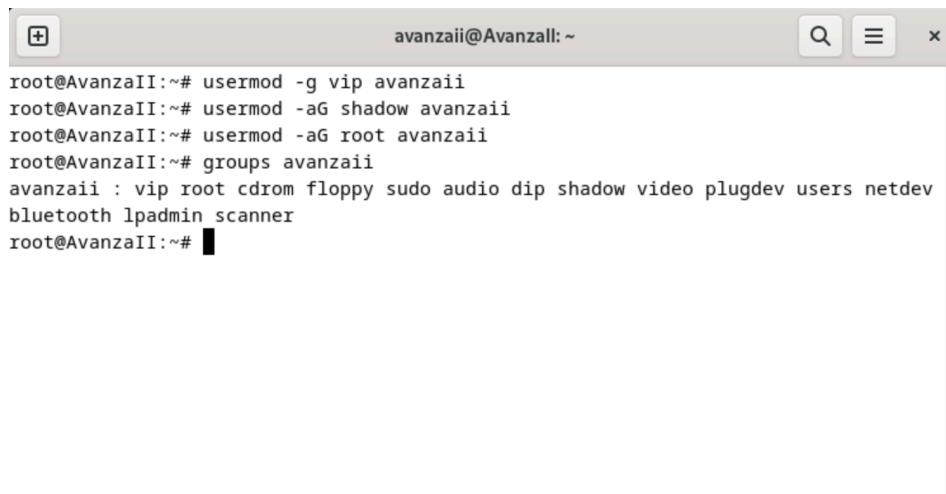
Una vez comprobamos que todo funciona correctamente, activamos los siguientes módulos:

- Habilitamos la ejecución de scripts en el servidor: **a2enmod cgi**
- Habilitamos las páginas personales en el directorio /home de cada usuario: **a2enmod userdir**
- Habilitamos el plugin suexec para dar más seguridad al servidor, esto permite ejecutar scripts a un grupo reducido de usuarios: **a2enmod suexec**

Las peticiones que se realizan en el servidor, son todas mediante el usuario www-data por lo que para blindar un poco más el sistema, los permisos de ejecución de los scripts CGI, los tendrá el usuario que creamos al principio con el objetivo de delegarle tareas de los usuarios principales que se ocupan de este tipo de funciones dentro del servidor, por lo tanto vamos a asignarle al usuario “avanzaii” los grupos y permisos necesarios:

- Creamos el grupo vip: **groupadd vip**
- Le asignamos el grupo vip como grupo principal a nuestro usuario: **usermod -g vip avanzaii**

- Le añadimos al grupo shadow para que pueda gestionar correctamente los ficheros shadow y las altas y bajas de los usuarios del sistema: **usermod -aG shadow avanzaii**
- Le añadimos también al grupo root para que tenga permisos de ejecución: **usermod -aG root avanzaii**



A screenshot of a terminal window titled "avanzaii@AvanzaII: ~". The window contains the following text:

```

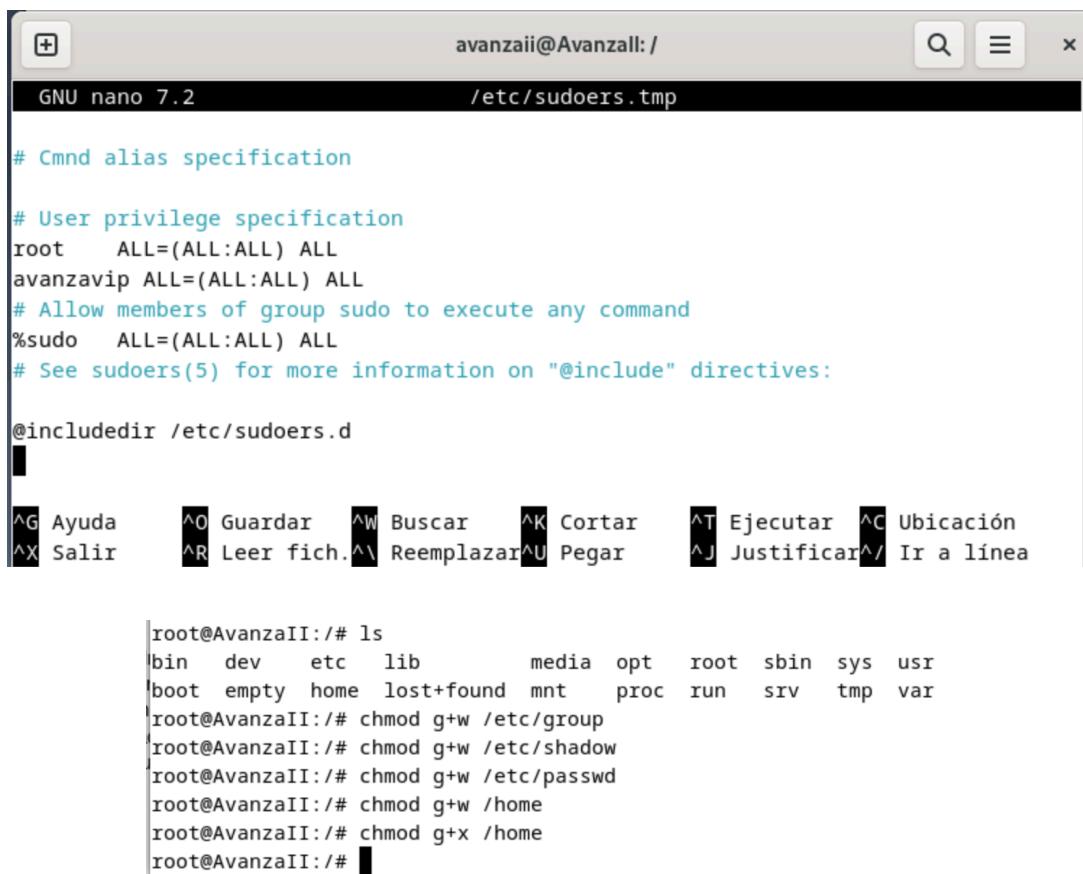
root@AvanzaII:~# usermod -g vip avanzaii
root@AvanzaII:~# usermod -aG shadow avanzaii
root@AvanzaII:~# usermod -aG root avanzaii
root@AvanzaII:~# groups avanzaii
avanzaii : vip root cdrom floppy sudo audio dip shadow video plugdev users netdev
bluetooth lpadmin scanner
root@AvanzaII:~#

```

Como podemos apreciar en la siguiente captura, el usuario avanzaii es un usuario completamente normal del sistema que pertenece a los grupos que ofrecen distintos servicios y adicionalmente, también a grupos privilegiados. Esto puede producir alguna que otra brecha de seguridad por lo que vamos a repetir el proceso anterior pero con un nuevo usuario que no tendrá ni opción de loguearse en el sistema, ni acceso a un directorio personal, para ello, utilizamos el siguiente comando: **adduser -system -home /empty avanzavip -shell=/bin/false**

Y por supuesto, le retiramos los grupos al anterior usuario así como el acceso al grupo sudo que se estableció en el punto 3 de configuración inicial del sistema.

Finalmente añadimos este nuevo usuario al fichero sudoers y modificamos los ficheros necesarios para la baja y alta de usuarios en el sistema, en concreto, permisos de lectura y escritura al grupo creador de los ficheros passwd, shadow, gshadow, groups y home, en este último, también le daremos permisos de ejecución:



```

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
avanzavip ALL=(ALL:ALL) ALL
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
# See sudoers(5) for more information on "@include" directives:

@include /etc/sudoers.d

[8 líneas leídas]

^G Ayuda      ^O Guardar     ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich.  ^\ Reemplazar ^U Pegar      ^J Justificar ^/ Ir a línea

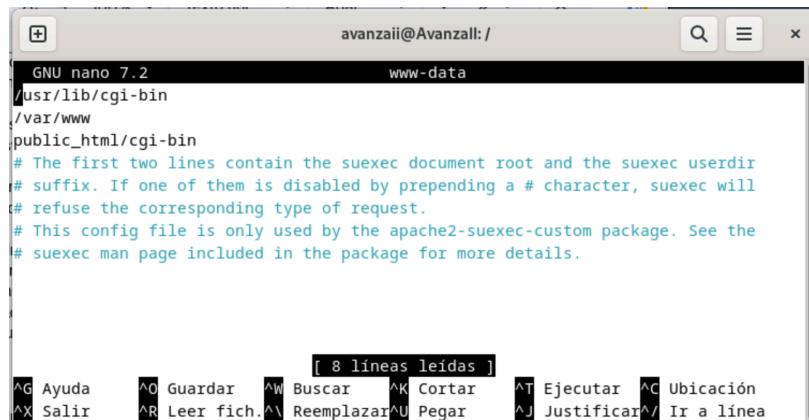
```

```

root@AvanzaII:/# ls
/bin  dev   etc   lib   media   opt   root   sbin   sys   usr
/boot empty home lost+found mnt   proc   run   srv   tmp   var
root@AvanzaII:/# chmod g+w /etc/group
root@AvanzaII:/# chmod g+w /etc/shadow
root@AvanzaII:/# chmod g+w /etc/passwd
root@AvanzaII:/# chmod g+w /home
root@AvanzaII:/# chmod g+x /home
root@AvanzaII:/#

```

Finalmente, modificamos los siguientes ficheros para que el usuario www-data pueda comunicarse sin problema con nuestro usuario avanzavip y sepa encontrar los scripts.



```

www-data

/usr/lib/cgi-bin
/var/www
/public_html/cgi-bin
# The first two lines contain the suexec document root and the suexec userdir
# suffix. If one of them is disabled by prepending a # character, suexec will
# refuse the corresponding type of request.
# This config file is only used by the apache2-suexec-custom package. See the
# suexec man page included in the package for more details.

[8 líneas leídas]

^G Ayuda      ^O Guardar     ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich.  ^\ Reemplazar ^U Pegar      ^J Justificar ^/ Ir a línea

```

```
GNU nano 7.2 avanzaii@Avanzall: / 000-default.conf

<VirtualHost *:443>
    ServerName www.avanzaii.es
    ServerAdmin webmaster@localhost

    SuExecUserGroup avanzavip vip
    <Directory "/usr/lib/cgi-bin">
        Options +ExecCGI
        AddHandler cgi-script .cgi .pl
        AddHandler default-handler .css .png .jpeg .jpg
    </Directory>

[ 24 líneas leidas ]
^G Ayuda      ^O Guardar     ^W Buscar     ^K Cortar     ^T Ejecutar ^C Ubicación
^X Salir      ^R Leer fich. ^\ Reemplazar ^U Pegar      ^J Justificar ^/ Ir a línea
```

Como se puede ver en la imagen, hemos añadido la directiva `SuExecUserGroup` para indicar el usuario que va a ejecutar los scripts y después el directorio donde van a ir dichos scripts, y adicionalmente, un par de controladores que permitirán detectar correctamente el formato de los distintos ficheros que voy a estar utilizando.

6. GESTIÓN DE USUARIOS

Para comenzar con este apartado, vamos a crear los grupos de usuarios que nos permitirán hacer distinciones entre profesores y alumnos, en concreto:

- `groupadd -g 230 profesores`
- `groupadd -g 231 alumnos`

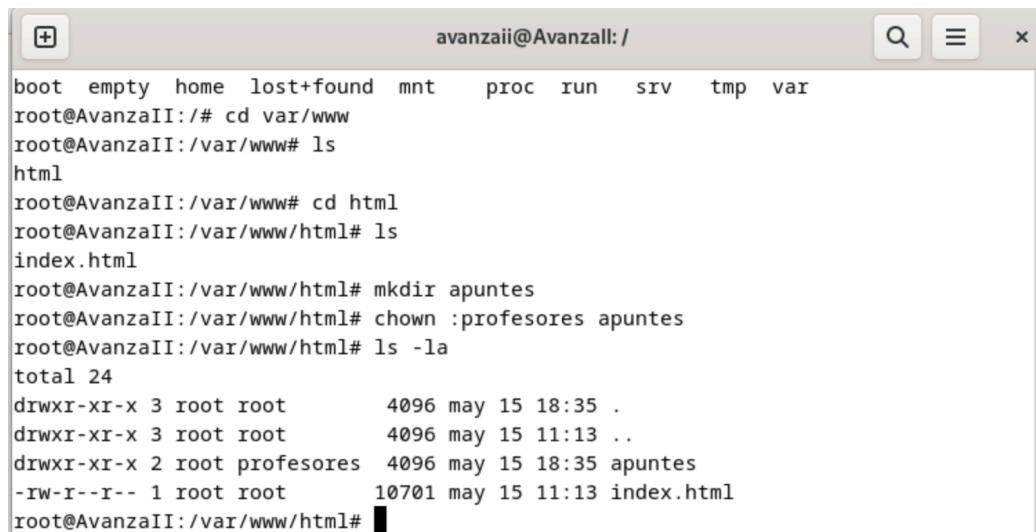
```
profesores:x:230:
alumnos:x:231:
root@AvanzaII:/etc#
```

Luego, creamos el directorio `/apuntes` en `"/var/www/html"` para que pueda ser accesible desde el servidor web.

- `mkdir /var/www/html/apuntes`

Y le damos la propiedad del directorio al grupo de los profesores para que únicamente ellos puedan realizar modificaciones: `chown :profesores /var/www/html/apuntes` y luego `chmod 755 /var/www/html/apuntes` para dar los permisos necesarios al resto de ficheros que se creen dentro de este directorio.

Asignamos el bit setid para el grupo con el objetivo de que los usuarios pertenecientes al mismo grupo puedan interactuar sobre los mismos archivos sin problemas: **chmod g+s apuntes/**



```
avanzaii@AvanzaII: /boot empty home lost+found mnt proc run srv tmp var
root@AvanzaII:# cd var/www
root@AvanzaII:/var/www# ls
html
root@AvanzaII:/var/www# cd html
root@AvanzaII:/var/www/html# ls
index.html
root@AvanzaII:/var/www/html# mkdir apuntes
root@AvanzaII:/var/www/html# chown :profesores apuntes
root@AvanzaII:/var/www/html# ls -la
total 24
drwxr-xr-x 3 root root      4096 may 15 18:35 .
drwxr-xr-x 3 root root      4096 may 15 11:13 ..
drwxr-xr-x 2 root profesores 4096 may 15 18:35 apuntes
-rw-r--r-- 1 root root     10701 may 15 11:13 index.html
root@AvanzaII:/var/www/html#
```

Por otro lado, crearemos también un enlace simbólico al fichero /var/log/apache2/access.log para poder visualizar el log de los accesos de los usuarios, este enlace solo será utilizado por el root.

In -s /var/log/apache2/access.log /accesos.log

7. DIRECTORIO SKEL

Este será el directorio que se replicará al crear cada usuario y al cual estos tendrán acceso a los archivos que haya dentro, es un directorio genérico, por lo cual, vamos a hacer una serie de ajustes.

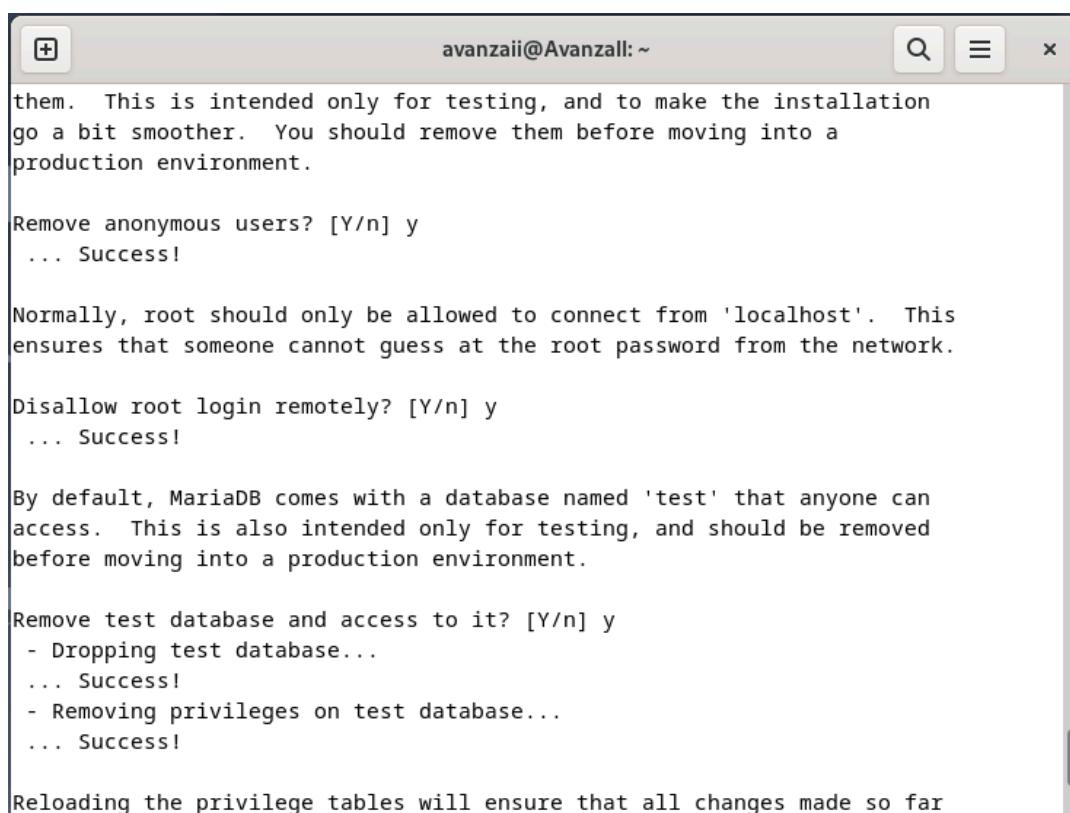
- Enlazamos de manera simbólica este directorio con el directorio /apuntes creado anteriormente, así los nuevos usuarios tendrán acceso a los apuntes desde su directorio personal: **In -s /var/www/html/apuntes /etc/skel**
- Creamos el fichero “condiciones.txt” como se especifica en el enunciado de la práctica: **nano condiciones.txt**
- Directorio en el que se va a albergar la página personal, inicialmente se encuentra desactivada por lo que tiene un nombre distinto a public_html

8. BASE DE DATOS MARIADB

Como sugiere el enunciado de la práctica, estaré usando MariaDB como base de datos permanente ya que posee una gran potencia y es software libre, además por supuesto de trabajar en sintaxis SQL.

Comenzamos instalando el paquete que contiene el servidor de la base de datos de MariaDB: **apt install mariadb-server mariadb-client**

Después, ejecutamos el siguiente script para realizar una instalación segura de la base de datos, esto nos permitirá establecer unas credenciales de acceso para el usuario root de la base de datos y además, también nos permitirá eliminar cuentas root que sean accesibles desde fuera del entorno local, así como eliminar cuentas anónimas. **mysql_secure_installation**



```
avanzaii@Avanzall: ~
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

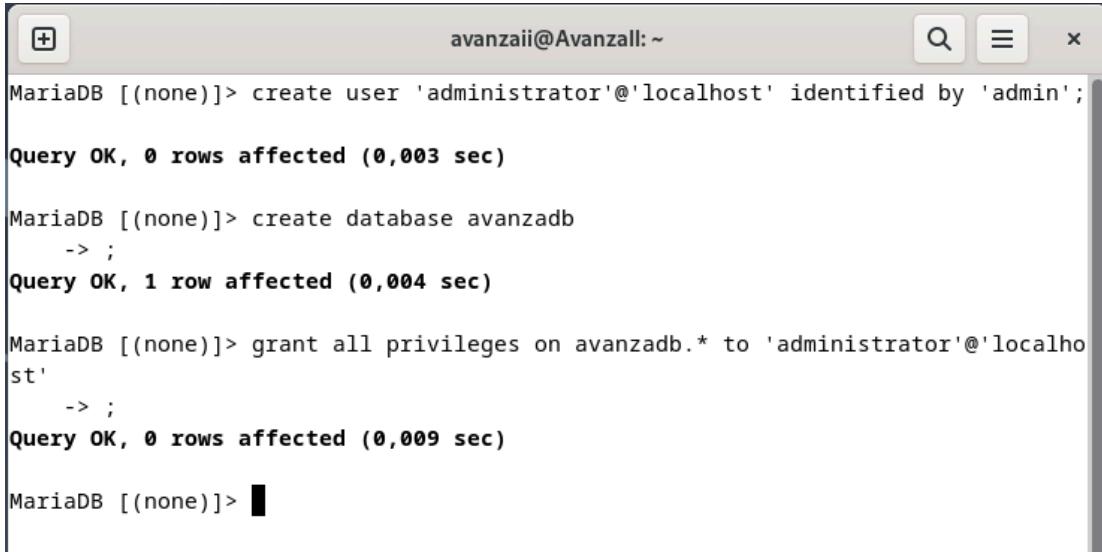
Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
```

Ahora, deberemos crear un nuevo usuario gestor de la base de datos ya que estos no tienen nada que ver con los usuarios del sistema linux, para ello, ejecutamos los siguientes comandos:

- Entramos en la consola de MariaDB: **mysql**
- Creamos un nuevo usuario con: **create user 'administrator'@'localhost' identified by 'admin';**
- Creamos la base de datos que vamos a utilizar: **create database avanzadb;**

- Le asignamos permisos al usuario administrator sobre todas las tablas de avanzadb: **grant all privileges on avanzadb.* to 'administrator'@'localhost';**



```

avanzaii@Avanzall: ~
MariaDB [(none)]> create user 'administrator'@'localhost' identified by 'admin';
Query OK, 0 rows affected (0,003 sec)

MariaDB [(none)]> create database avanzadb
    -> ;
Query OK, 1 row affected (0,004 sec)

MariaDB [(none)]> grant all privileges on avanzadb.* to 'administrator'@'localhost';
    -> ;
Query OK, 0 rows affected (0,009 sec)

MariaDB [(none)]>

```

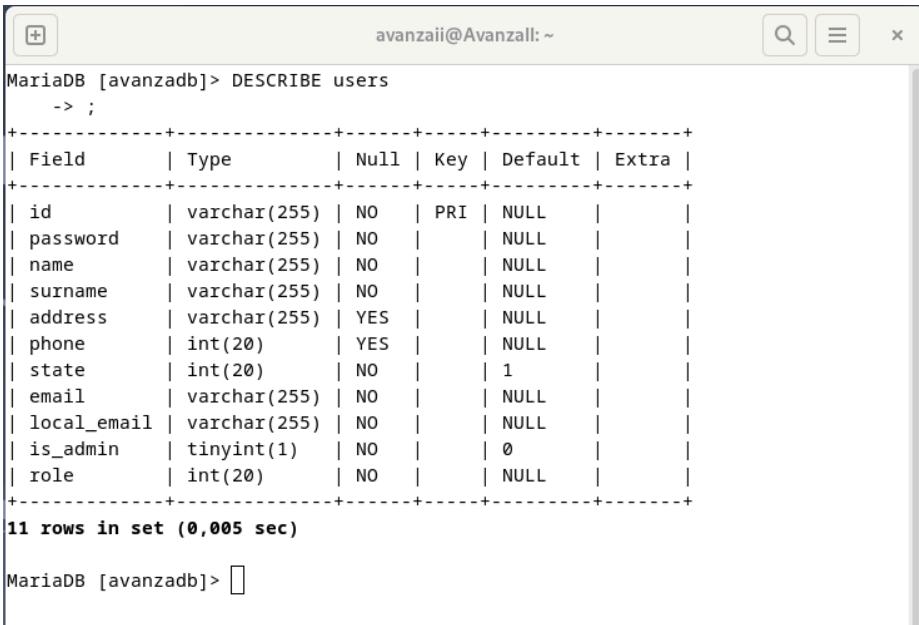
- Accedemos a la base de datos avanzadb: **USE avanzadb;**
 → Finalmente, creamos una tabla en avanzadb llamada users, con todos los campos que necesitaremos para registrar a un usuario:

```

create table users (id varchar(255) not null primary key, password
varchar(255) not null, name varchar(255) not null, surname varchar(255)
not null, address varchar(255), phone int(20), state int(20) not null
default 1, email varchar(255) not null, local_email varchar(255) not null,
is_admin BOOLEAN not null default false, role int(20) not null);

```

- Verificamos que todo se ha realizado correctamente con: **DESCRIBE users;**



```

avanzaii@Avanzall: ~
MariaDB [avanzadb]> DESCRIBE users
    -> ;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id    | varchar(255) | NO  | PRI | NULL   |       |
| password | varchar(255) | NO  |     | NULL   |       |
| name  | varchar(255) | NO  |     | NULL   |       |
| surname | varchar(255) | NO  |     | NULL   |       |
| address | varchar(255) | YES |     | NULL   |       |
| phone  | int(20)    | YES |     | NULL   |       |
| state  | int(20)    | NO  |     | 1      |       |
| email  | varchar(255)| NO  |     | NULL   |       |
| local_email | varchar(255)| NO  |     | NULL   |       |
| is_admin | tinyint(1) | NO  |     | 0      |       |
| role   | int(20)    | NO  |     | NULL   |       |
+-----+-----+-----+-----+-----+
11 rows in set (0,005 sec)

MariaDB [avanzadb]>

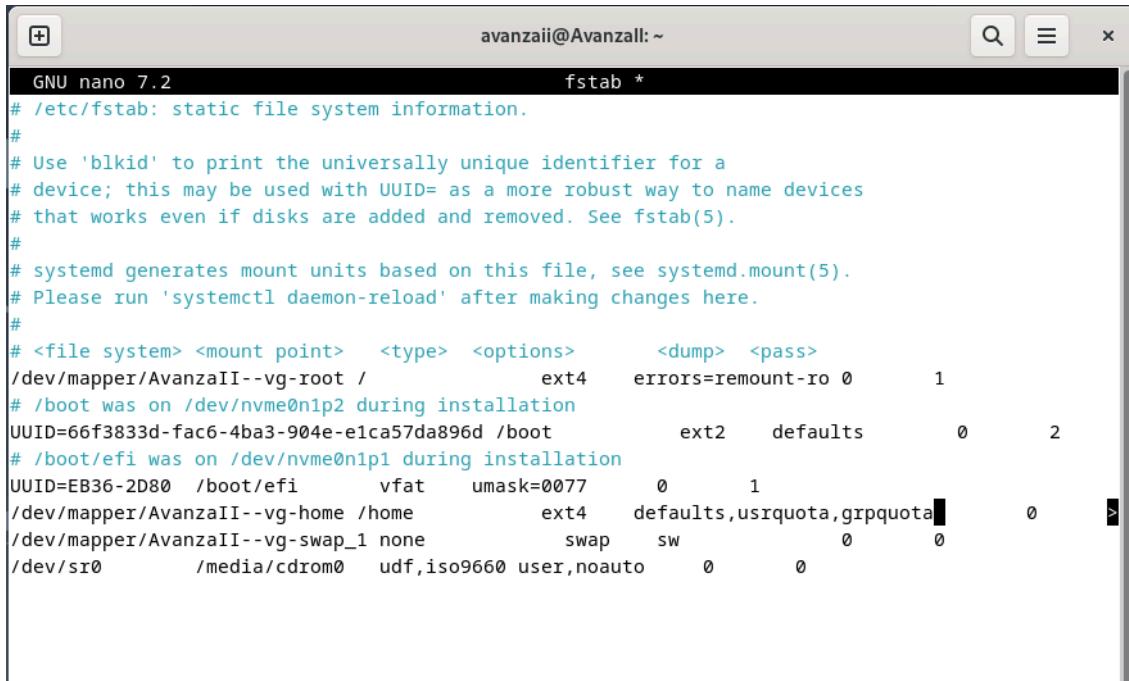
```

Para terminar, registramos al usuario avanzaii: **INSERT INTO users (id, password, name, surname, state, email, local_email, is_admin, role) VALUES ('avanzaii', 'SoyV1P?', 'administrador', 'avanzaii', 2, 'alvarogarciatic@gmail.com', 'avanzaii@avanzaii', 1, 2);**

9. CONFIGURACIÓN DE QUOTAS

Ahora, vamos a designar una serie de cuotas para que los usuarios no colapsen el sistema con archivos innecesarios. Como en el enunciado no se especifica ningún límite de espacio, vamos a establecer que cada usuario disponga de 500 Mb en disco, tanto los alumnos como los profesores. Antes de comenzar, debemos instalar y activar las cuotas, para ello:

- Instalamos el paquete quota: **apt-get install quota**
- Editamos el fichero /etc/fstab para añadir usrquota y grpquota, en concreto, lo tendremos que hacer en el sistema de ficheros /home ya que se hizo esta partición al instalar el sistema operativo, por los motivos que se comentaron anteriormente: **nano /etc/fstab**

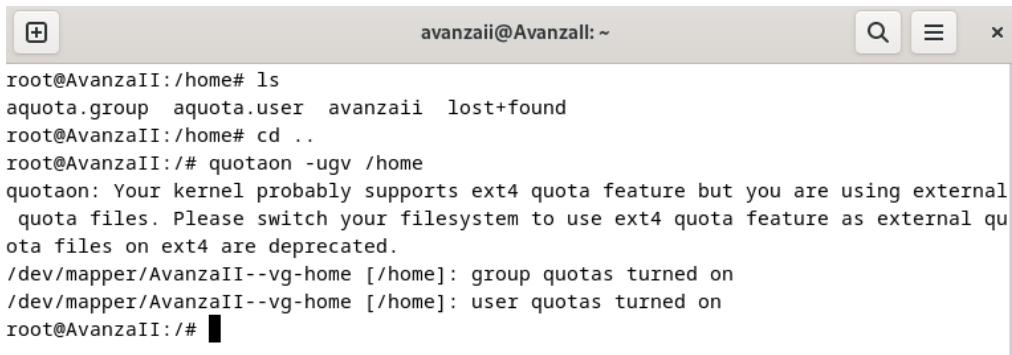


```

GNU nano 7.2                               avanzaii@Avanzall: ~
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).
#
# systemd generates mount units based on this file, see systemd.mount(5).
# Please run 'systemctl daemon-reload' after making changes here.
#
# <file system> <mount point>   <type>   <options>      <dump>   <pass>
/dev/mapper/AvanzaII--vg-root /           ext4     errors=remount-ro 0      1
# /boot was on /dev/nvme0n1p2 during installation
UUID=66f3833d-fac6-4ba3-904e-e1ca57da896d /boot       ext2     defaults        0      2
# /boot/efi was on /dev/nvme0n1p1 during installation
UUID=EB36-2D80  /boot/efi      vfat    umask=0077        0      1
/dev/mapper/AvanzaII--vg-home /home       ext4     defaults,usrquota,grpquota 0      0
/dev/mapper/AvanzaII--vg-swap_1 none       swap    sw            0      0
/dev/sr0      /media/cdrom0 udf,iso9660 user,noauto 0      0

```

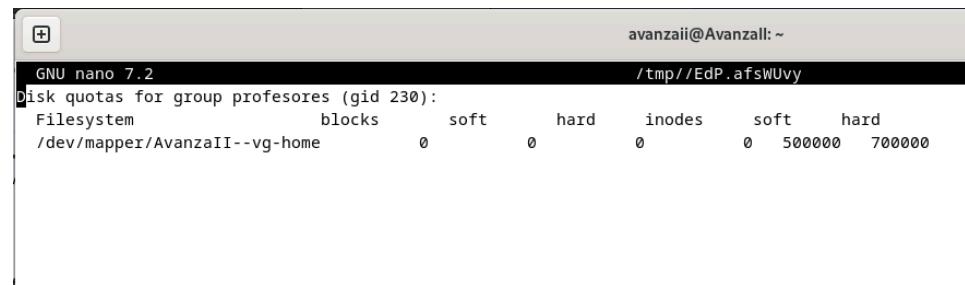
- Reiniciamos con: **reboot**
- Verificamos que se hayan creado los ficheros **aquota.user** y **aquota.group**, si no, ejecutamos: **quotacheck -mfcug /home**



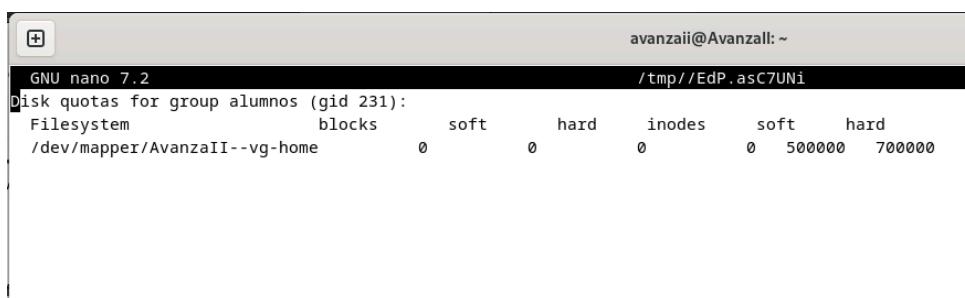
```
avanzaii@AvanzaII: ~
root@AvanzaII:/home# ls
aquota.group aquota.user avanzaii lost+found
root@AvanzaII:/home# cd ..
root@AvanzaII:# quotaon -ugv /home
quotaon: Your kernel probably supports ext4 quota feature but you are using external
quota files. Please switch your filesystem to use ext4 quota feature as external qu
ota files on ext4 are deprecated.
/dev/mapper/AvanzaII--vg-home [/home]: group quotas turned on
/dev/mapper/AvanzaII--vg-home [/home]: user quotas turned on
root@AvanzaII:/#
```

Una vez configuradas y activas las cuotas en los directorios /home de los usuarios, utilizamos el comando **edquota** para asignarles límites de espacio, tanto blandos, como duros. Más concretamente, se lo asignaremos a los distintos grupos de usuarios (profesores y alumnos) para que todos sus miembros tengan los mismos recursos asignados, para ello:

- Establecemos un límite blando de 500 Mb y un límite duro de 700 Mb para cada grupo: **edquota -g profesores** y **edquota -g alumnos**



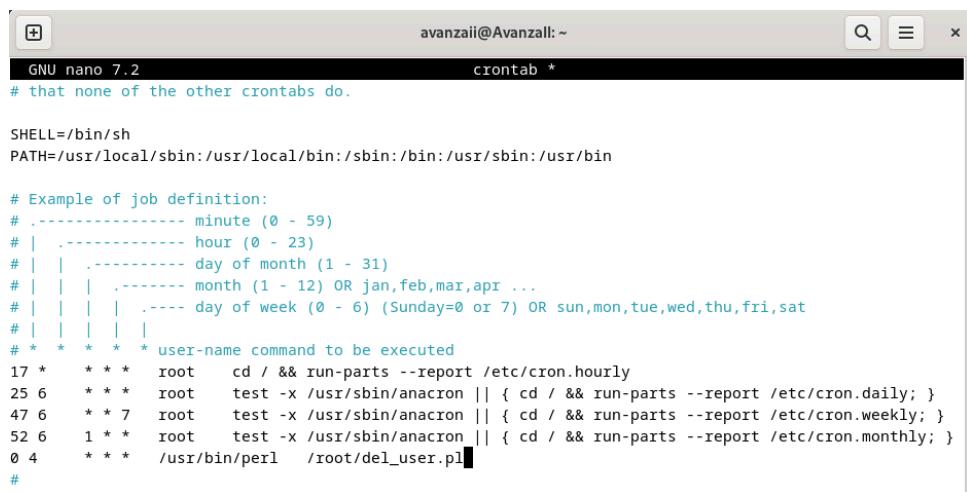
```
avanzaii@AvanzaII: ~
GNU nano 7.2
/tmp//EdP.afsWUvy
Disk quotas for group profesores (gid 230):
Filesystem      blocks      soft      hard      inodes      soft      hard
/dev/mapper/AvanzaII--vg-home      0        0        0        0    500000    700000
```



```
avanzaii@AvanzaII: ~
GNU nano 7.2
/tmp//EdP.asC7UNi
Disk quotas for group alumnos (gid 231):
Filesystem      blocks      soft      hard      inodes      soft      hard
/dev/mapper/AvanzaII--vg-home      0        0        0        0    500000    700000
```

10. ELIMINAR USUARIOS SIN CONFIRMAR

Con el objetivo de mantener el sistema actualizado y optimizado, debemos eliminar los datos que no son necesarios y están ocupando un espacio limitado, por ello, debemos eliminar los usuarios que no hayan confirmado su cuenta en un periodo delimitado a través del email que se le enviará en el momento de darle de alta. Para llevar a cabo esta tarea, creamos un script en perl que realice esta función (`del_user.pl`) y crearemos también el fichero crontab que nos permitirá ejecutar órdenes de manera automática estableciendo unos tiempos concretos.



```
GNU nano 7.2                               crontab *
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# ----- minute (0 - 59)
# | ----- hour (0 - 23)
# | | ----- day of month (1 - 31)
# | | | ----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | ----- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.daily; }
47 6    * * 7   root    test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.weekly; }
52 6    1 * *   root    test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.monthly; }
0 4    * * *   /usr/bin/perl  /root/del_user.pl
#
```

Como podemos ver, programamos la tarea para que se ejecute cada 4 min.

11. BACKUPS

Este es un punto de vital importancia en cualquier sistema, por lo que vamos a establecer una política de copias de seguridad segura, eficiente y automática, para ello, haremos uso de la librería `File::Rotate::Backup` de CPAN, que nos permitirá realizar una estrategia de copia incremental en la que se añade a la copia, únicamente los archivos que han sido modificados a lo largo del día. Además, utilizaremos la herramienta `rclone` para realizar un respaldo adicional en la nube. Seguimos los siguientes pasos:

- Instalamos rclone: `apt install rclone`
- Creamos dos directorios en la raíz, uno donde se almacenarán las copias de seguridad y otro de respaldo del primero: `mkdir /back` y `mkdir /back2`
- Añadimos a /back los scripts necesarios para realizar las copias de seguridad, `back.sh` y `back.pl`

```
avanzaii@Avanzall: ~
GNU nano 7.2          back.sh
#!/bin/bash

/bin/perl /back/back.pl

/bin/ls | /bin/grep -P "^[^.]*$" | /bin/xargs -d"\n" rm -r

/bin/rclone -v sync /back/ avz_bck:back
```

```
avanzaii@Avanzall: ~
GNU nano 7.2          back.pl
#!/usr/bin/perl

use File::Rotate::Backup;

my $params = { archive_copies => 3,
               dir_copies => 1,
               backup_dir => '/back',
               file_prefix => 'av_back_',
               secondary_backup_dir => '/back2',
               secondary_archive_copies => 3,
               verbose => 0,
               use_flock => 1,
             };

my $backup = File::Rotate::Backup->new($params);

$backup->backup([ ['/home/' => 'home'],
                  ['/sbin/' => 'sbin'],
                  ['/etc/skel' => 'etc_skel'],
                  ['/var/www/' => 'var_www'],
                  ['/usr/lib/cgi-bin/' => 'usr_lib_cgibin'],
                  ['/var/lib/mysql/' => 'var_lib_mysql'],
                ]);
$backup->rotate;
```

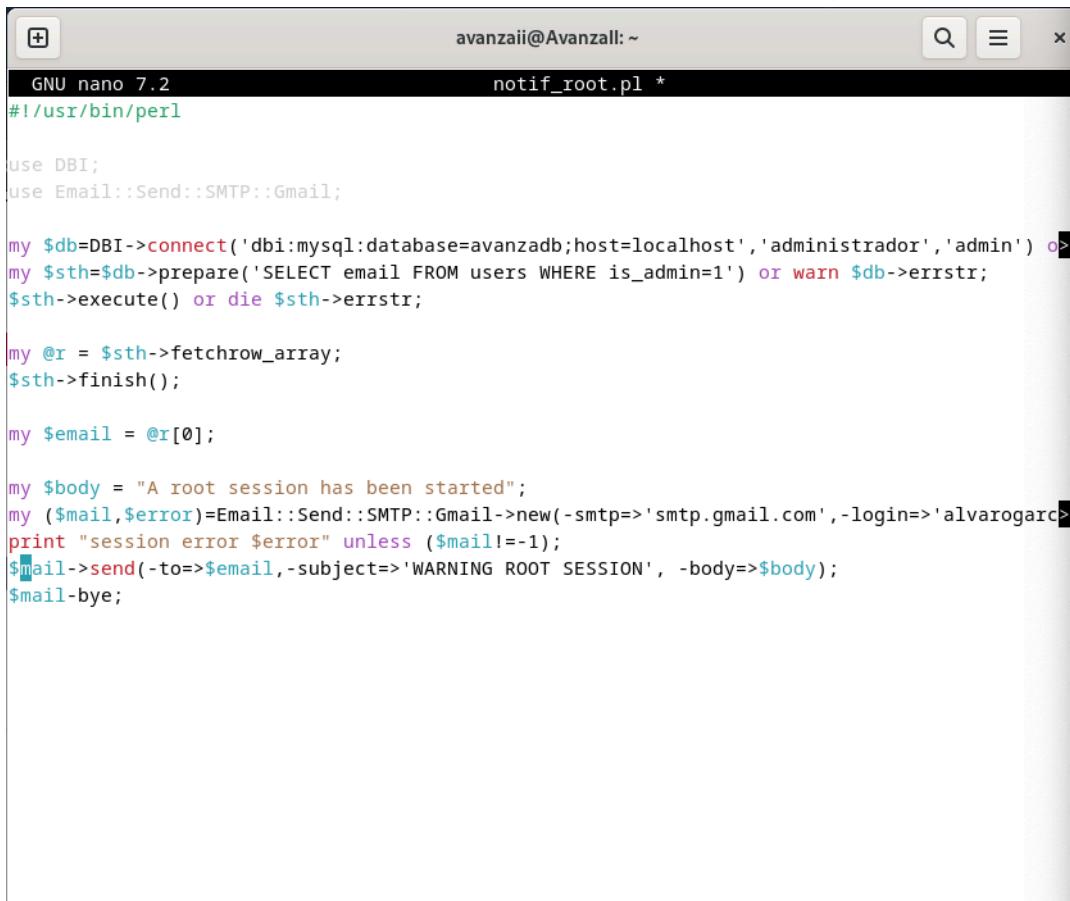
Finalmente, añadimos el fichero back.sh al crontab para que la tarea se ejecute todos los días, en concreto, a las 10pm:

0 0	22 * *	/bin/bash	/back/back.sh
-----	--------	-----------	---------------

12. AVISO ROOT

Como se sugiere en el enunciado de la práctica, una de las mejoras en cuanto a seguridad que se propone integrar, es el aviso y notificación por correo electrónico al usuario root cada vez que este se loguee en el sistema. Para ello, modificaremos el fichero del /root .bashr que se ejecuta cada vez que se va a iniciar la sesión y le añadimos la siguiente línea: perl /root/notif_root.pl

Creamos el script en perl en el directorio root y haremos uso de la librería Email::Send::SMTP::Gmail para enviar un correo electrónico al root:



The screenshot shows a terminal window titled "GNU nano 7.2" with the file name "notif_root.pl" at the top. The script content is as follows:

```
#!/usr/bin/perl

use DBI;
use Email::Send::SMTP::Gmail;

my $db=DBI->connect('dbi:mysql:database=avanzadb;host=localhost','administrador','admin') or die "Database connection failed";
my $sth=$db->prepare('SELECT email FROM users WHERE is_admin=1') or warn $db->errstr;
$sth->execute() or die $sth->errstr;

my @r = $sth->fetchrow_array;
$sth->finish();

my $email = @r[0];

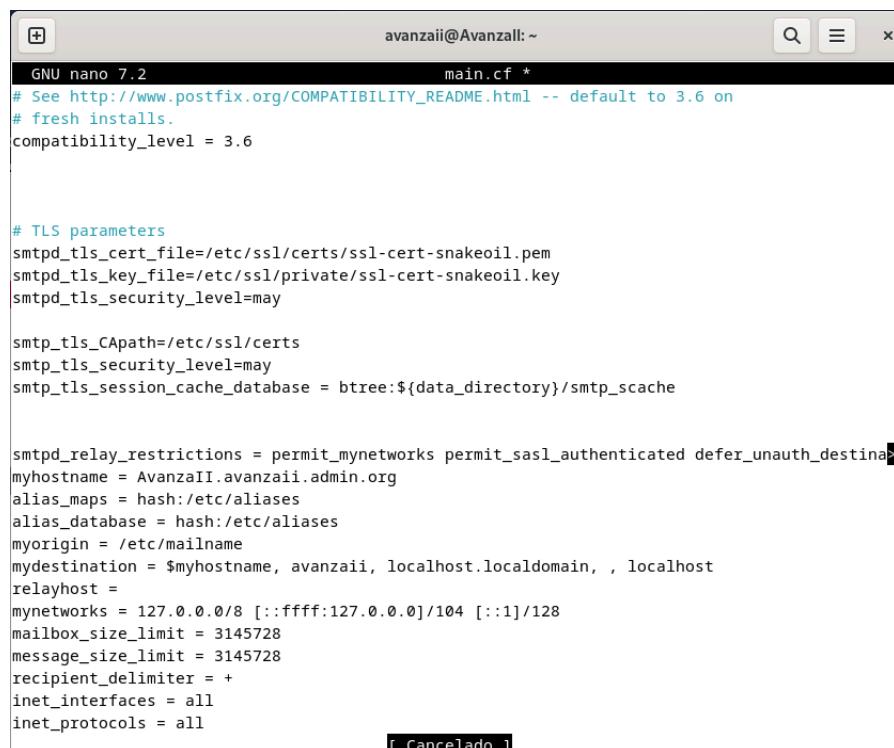
my $body = "A root session has been started";
my ($mail,$error)=Email::Send::SMTP::Gmail->new(-smtp=>'smtp.gmail.com',-login=>'alvarogarcia1993@gmail.com',-password=>'contraseña');
print "session error $error" unless ($mail!=-1);
$mail->send(-to=>$email,-subject=>'WARNING ROOT SESSION', -body=>$body);
$mail->bye;
```

13. SERVICIO CORREO ELECTRÓNICO

Para ofrecer a los usuarios un servicio de correo electrónico, he decidido emplear la herramienta Postfix para que se encargue del envío de los correos, Dovecot como servidor y RoundCube como webmail para el servidor web. Vamos a ver paso a paso el entorno y configuración de todas estas herramientas:

- Instalamos postfix: **apt install postfix** (opción servicio internet)
- Instalamos DoveCot: **apt install dovecot-imapd**
- Instalamos roundcube: **apt install roundcube**

Después, pasamos a configurarlos mediante el fichero /etc/postfix/main.cf, en el que podemos ajustar el tamaño de los mensajes y del buzón, en este caso, 3 Mb.



```
GNU nano 7.2                               main.cf *
# See http://www.postfix.org/COMPATIBILITY_README.html -- default to 3.6 on
# fresh installs.
compatibility_level = 3.6

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_tls_security_level=may

smtp_tls_CPath=/etc/ssl/certs
smtp_tls_security_level=may
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination
myhostname = AvanzaII.avanzaii.admin.org
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = $myhostname, avanzaii, localhost.localdomain, , localhost
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 3145728
message_size_limit = 3145728
recipient_delimiter =
inet_interfaces = all
inet_protocols = all
```

Deberemos modificar también el fichero /etc/roundcube/config.inc.php para poder enviar correos desde la propia plataforma:

```
// SMTP username (if required) if you use %u as the username Roundcube
// will use the current username for login
$config['smtp_user'] = '';
```

Finalmente, creamos un enlace simbólico para que se pueda acceder desde la web:
In -s /usr/share/roundcube /var/www/html/webMail

14. SERVICIO SFTP

Para encapsular a los usuarios en su directorio /home cuando acceden de manera remota, usaremos el servidor Vsftpd. **apt install vsftpd**

Modificamos el fichero de configuración /etc/vsftpd.conf y reiniciamos el servicio con **systemctl restart vsftpd.service**

Finalmente, creamos un enlace simbólico hacia vsftpd.log que genera el propio software: **In -s /var/log/vsftpd.log /vsftp.log**

15. MONITORIZACIÓN DEL SISTEMA

Llevaremos a cabo la tarea de realizar la monitorización del sistema con la herramienta acct que permite generar informes sobre los tiempos de conexión de los usuarios y ejecución de procesos.

- Lo instalamos: **apt install acct**
- Lo activamos: **accton on**

Creamos el directorio /acct en el que se van a guardar todos los archivos generados. Este directorio deberá crearse en el directorio principal del root /root/acct y adicionalmente, incluimos los scripts acct.sh y acct.pl.

Añadimos la ejecución del script acct.sh al crontab para que se ejecute manera periódica

```
# m h  dom mon dow    command
0 4 * * * /usr/bin/perl /root/del_user.pl
0 0 22 * * /bin/bash /back/back.sh
0 0 21 * * /bin/bash /root/acct/acct.sh
```

16. SCRIPTS

Para realizar ciertas tareas que se piden en el enunciado de la práctica, estaré utilizando una serie de librerías de perl y una serie de scripts que paso a comentar brevemente (algunos han sido comentados en puntos anteriores de este documento):

Para hacer uso de las librerías CPAN de perl, ejecutamos los siguientes comandos:

- Instalamos el paquete de CPAN: **apt install cpanminus**
- Instalamos las librerías necesarias: **cpanminus install nombreLibreria**

Haré uso de los siguientes módulos o librerías de CPAN:

- CGI
- CGI::Session
- DBI
- Sudo
- Linux::usermod
- Email::Send::SMTP::Gmail
- File::Copy::Recursive
- Mime::Base64
- Authen::Simple::PAM
- File::Rotate::Backup
- Sys::Hostname
- Socket

Adicionalmente a estos módulos, estaré también haciendo uso de la librería de PAM que me permitirá comprobar el login de los usuarios en el sistema: **apt-get install libauthen-simple-pam-perl**

Los scripts que estaré utilizando son los siguientes:

- acct.pl: Crea el archivo con las estadísticas del sistema y lo envía por correo al administrador.
- act_pag.cgi: permite activar la página personal.
- activacion.cgi: Finaliza el alta de los usuarios en el sistema
- notif_root.pl: Notifica al administrador de los inicios de sesión
- chg_pass.cgi: Modifica la contraseña de un usuario.
- cerrar_sesion.cgi: Finaliza la sesión de un usuario de manera segura
- del_user.cgi: Elimina los usuarios sin confirmar su cuenta.
- eliminar.cgi: Elimina de forma permanente a un usuario del sistema.
- login.cgi: Permite al usuario iniciar sesión en el sistema con un username y una contraseña.
- back.pl: realizar las copias de seguridad
- redirect.cgi y redirect_pag.cgi: Permite controlar la navegación entre las distintas páginas.

- registro.cgi: Da de alta un nuevo usuario en la base de datos
- rec_pass.cgi: Permite a un usuario recuperar su contraseña a través de un email.
- services.cgi: Indica a los usuarios el estado de los servicios que ofrece el sistema.

Finalmente, deberemos darle los permisos de ejecución y cambiar el propietario al de la carpeta del suexec:

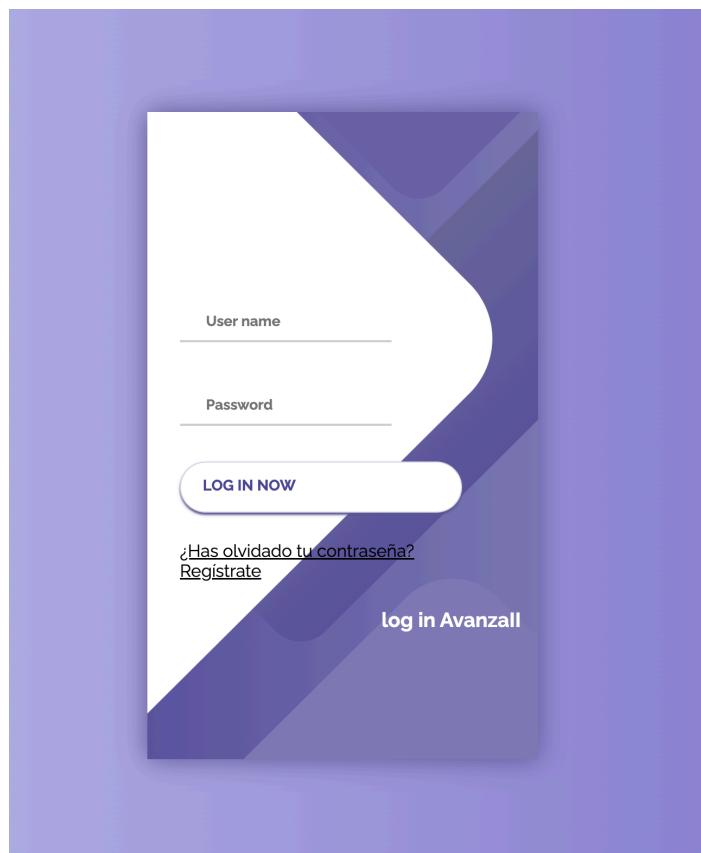
chown -R avanzavip:vip /usr/lib/cgi-bin

Con la opción -R, también afecta a la propiedad de los ficheros dentro del directorio. Finalmente, damos permisos de ejecución: **chmod a+X nombre.cgi**

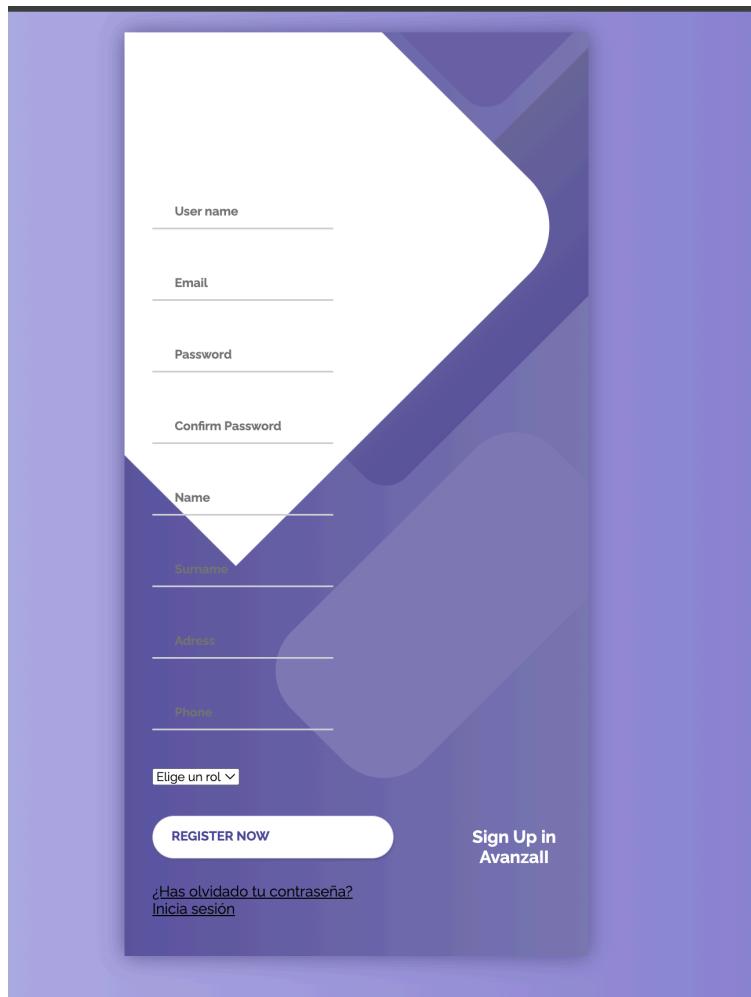
16.1. HTML

Con el objetivo de brindar una mejor experiencia dentro del propio servidor web, se han creado una serie de interfaces html para recopilar los datos de los usuarios y también, que estos puedan acceder de manera sencilla a los diferentes servicios. Las páginas creadas son las siguientes:

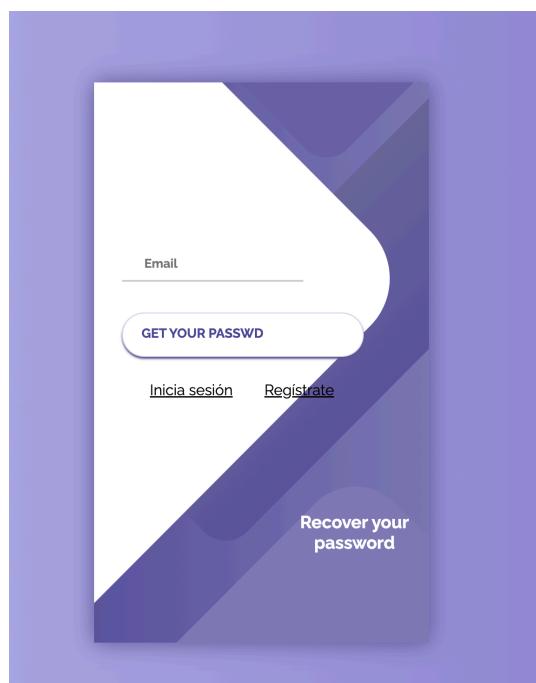
- index.html: Página de login



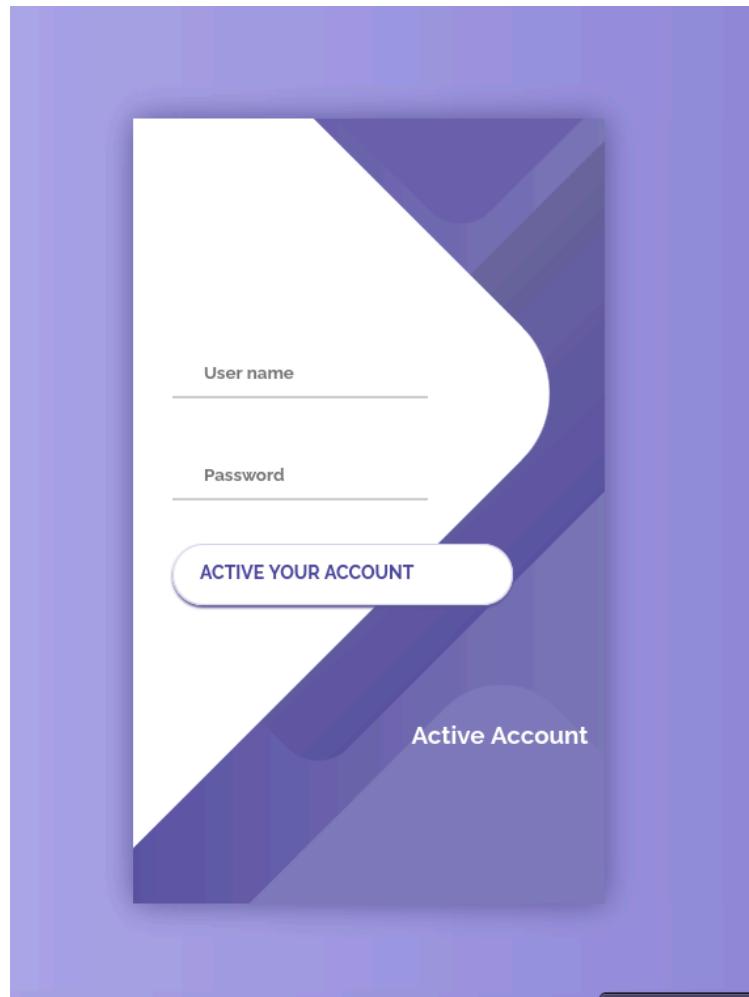
→ registro.html



→ rec_pass.html



→ activacion.html



→ admin.html



→ profesor.html



→ ajustes.html

Modificar contraseña

Old Password

New Password

Repeat New Password

CHANGE PASSWORD

Modificar datos personales

email

name

surname

address

phone

CHANGE DATA

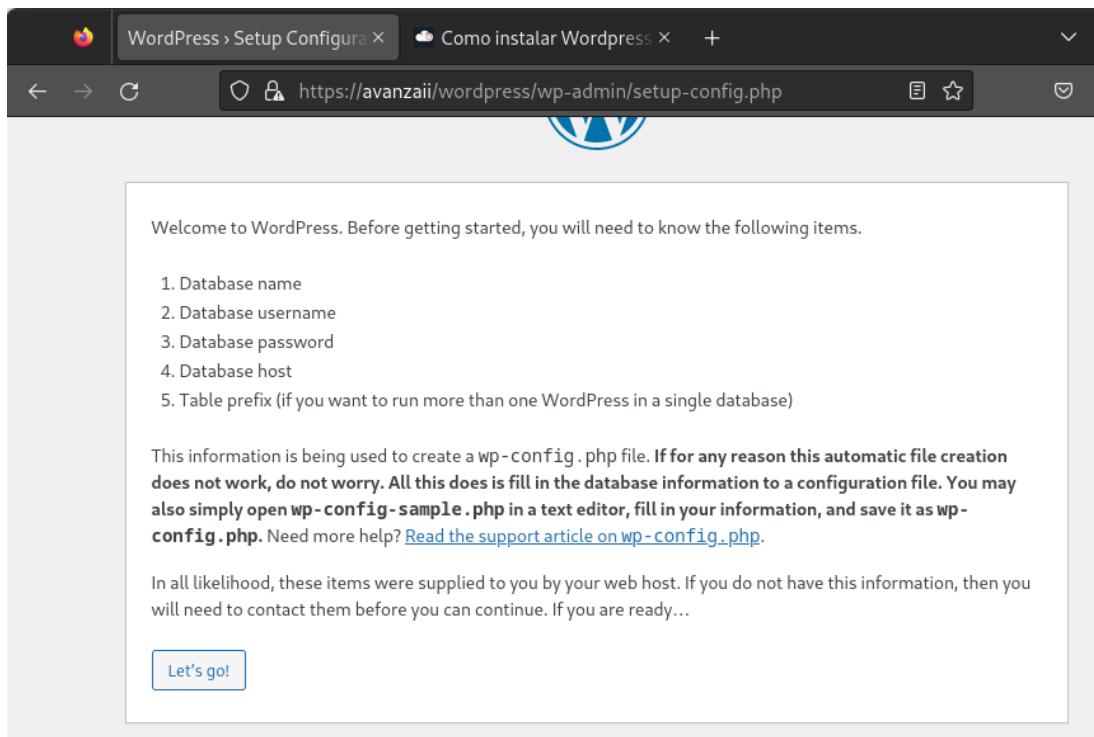
(Esta acción no se puede deshacer)

Eliminar la cuenta

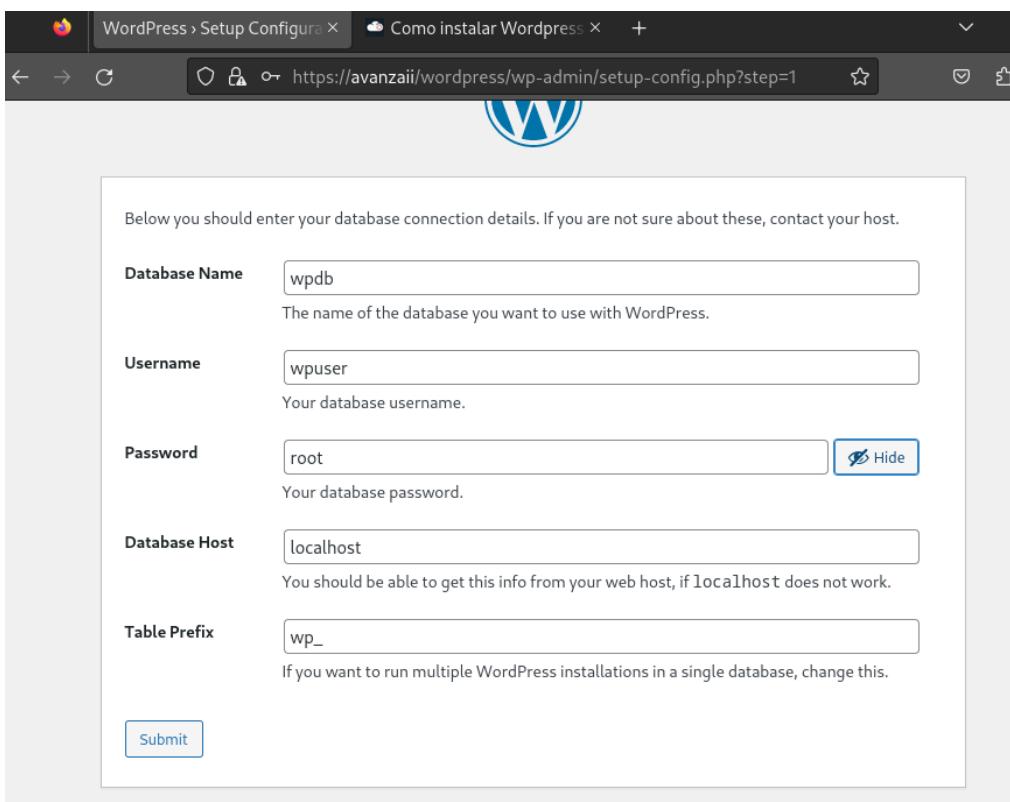
17. BLOG PERSONAL

Para este punto final, estaremos haciendo uso de la herramienta de código abierto wordpress, por lo que vamos a instalar la plataforma en nuestro servidor para poder ofrecer el servicio de blog personal, para ello:

- Descargamos el paquete: **wget <https://wordpress.org/latest.tar.gz>**
- Descomprimimos: **tar -zxvf latest.tar.gz**
- Cambiamos el propietario y los permisos para que www-data pueda acceder sin problemas: **chown www-data:www-data /var/www/html/wordpress -R**
- Configuramos la base de datos de wordpress, primero creándose en MariaDB: **CREATE DATABASE wpdb DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;**
- Creamos el usuario 'wpuser' y le damos permisos sobre la nueva base de datos: **CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'root'; GRANT ALL PRIVILEGES ON wpdb.* TO 'wpuser'@'localhost'; quit;**



Finalmente, accedemos a `/localhost/wordpress` para visualizar la interfaz de wordpress y proceder a su verificación e instalación:





Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

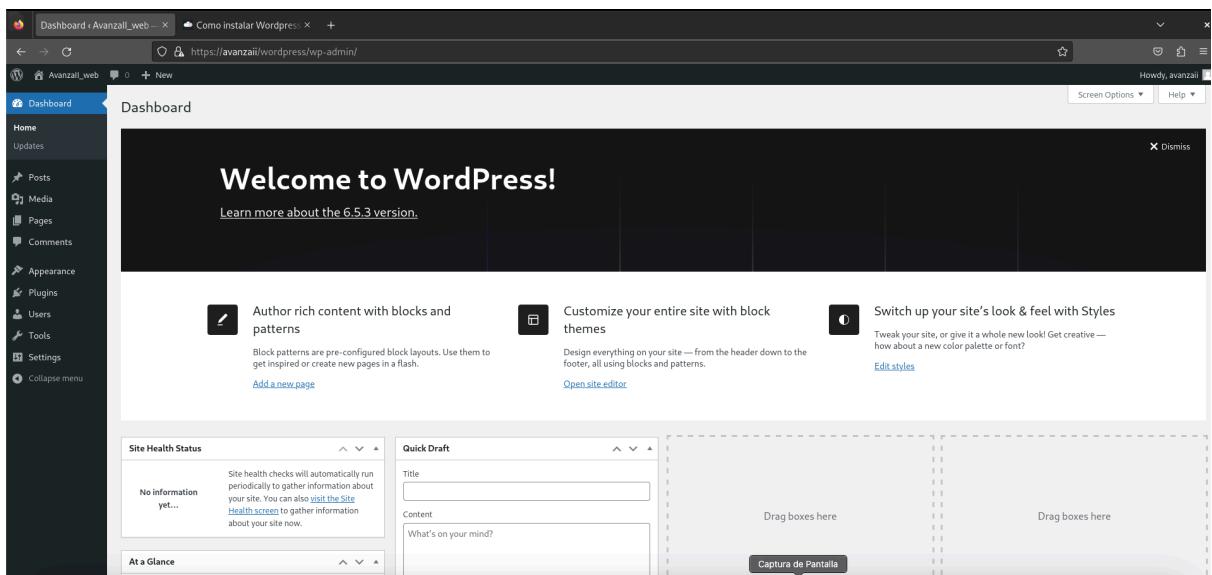
Site Title	Avanzall_web
Username	avanzaii
Password	Avanzaai_Admin_2024! Strong
Your Email	agarsan@usal.es
Search engine visibility	<input type="checkbox"/> Discourage search engines from indexing this site It is up to search engines to honor this request.

Important: You will need this password to log in. Please store it in a secure location.

Double-check your email address before continuing.

Install WordPress

Captura de Pantalla



The screenshot shows the WordPress dashboard after installation. The top navigation bar includes 'Dashboard', 'Updates', 'Posts', 'Media', 'Pages', 'Comments', 'Appearance', 'Plugins', 'Users', 'Tools', and 'Settings'. A 'Collapse menu' option is also present. The main content area features a large 'Welcome to WordPress!' heading and a link to 'Learn more about the 6.5.3 version.'. Below this, there are three cards: 'Author rich content with blocks and patterns' (with a note about block patterns), 'Customize your entire site with block themes' (with a note about design), and 'Switch up your site's look & feel with Styles' (with a note about styles). At the bottom left, there are sections for 'Site Health Status' (showing 'No information yet...') and 'Quick Draft' (with fields for 'Title' and 'Content'). On the right, there are two 'Drag boxes here' areas for the WordPress editor. A 'Captura de Pantalla' watermark is visible at the bottom center.