ADMINISTRACIÓN DE SISTEMAS

PRÁCTICA FINAL

VNiVERSiDAD D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

Juan José López Gómez - 70926305D
Sergio Sánchez García - 70961594Q

ÍNDICE

INTRODUCCIÓN

A lo largo de esta práctica explicaremos el proceso para crear un servidor LINUX que permite realizar las funciones básicas y principales que todo administrador de sistemas tiene que llevar a cabo con el objetivo de tener un sistema fiable, eficiente y seguro.

El servidor ofrece la funcionalidad necesaria para administrar una plataforma de estudiantes y alumnos, permitiendo todos los requisitos especificados en el enunciado de esta práctica.

INSTALACIÓN DE IMAGEN DEL SERVIDOR

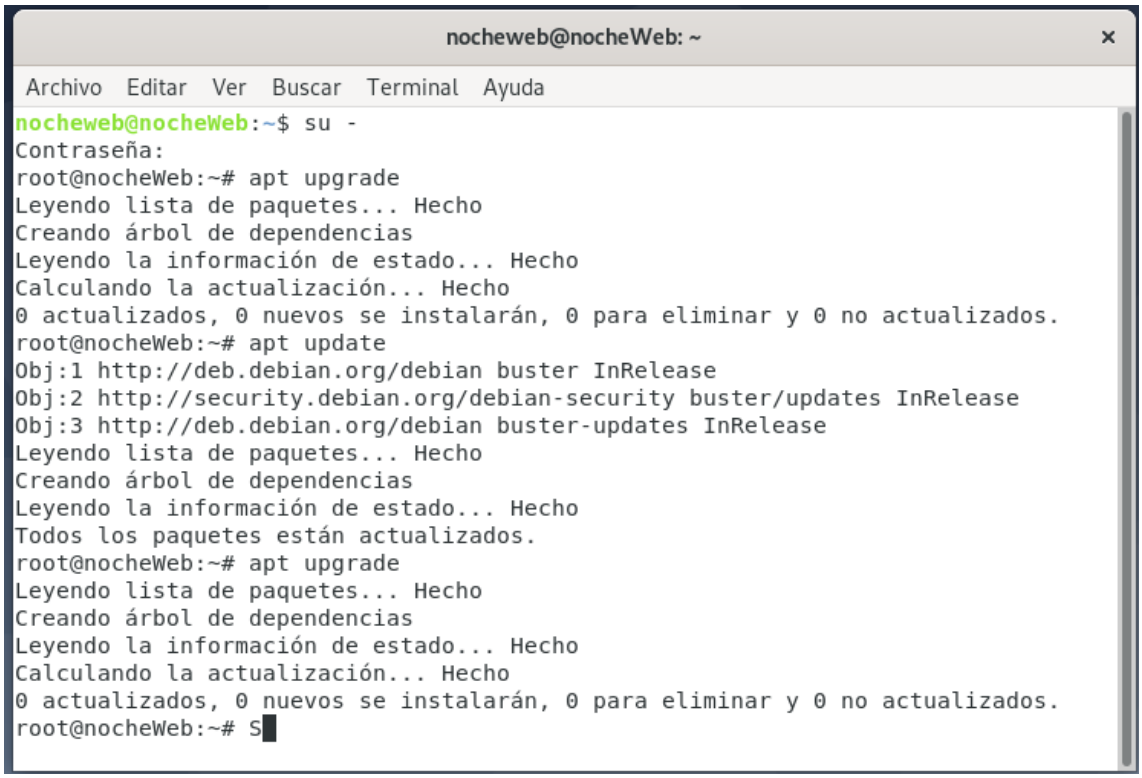
Lo primero que hicimos fue montar una imagen ISO de Debian 10 sobre una máquina virtual.

Es importante mencionar que al realizar esta instalación decidimos hacer una partición de home haciendo que el sistema consta de dos particiones, una que contiene toda la funcionalidad del propio servidor y la segunda la información de los usuarios. Esto se ha realizado con el objetivo de desacoplar la gestión de los usuarios del resto del sistema para garantizar una mayor seguridad y mejorar el rendimiento del servidor, esto se puede apreciar por ejemplo en el sistema de cuotas que sólo se aplicará a la partición de los usuarios en lugar de a todo el sistema.

CONFIGURACIÓN INICIAL

Tras finalizar la instalación, abrimos una terminal y mediante el comando **su -** nos convertimos en súper usuario para poder realizar el resto de los pasos sin ningún problema y facilitar la creación del sistema.

Después ejecutamos **apt update** y **apt upgrade** para actualizar el sistema y la lista de paquetes disponibles, esta es una buena práctica que nos permite mantener el sistema actualizado.



Después, mediante **apt-get install build-essential**, que es un paquete propio de Debian que contiene una serie de paquetes como gcc, g++ y make entre otros. De esta forma, instalamos todos estos con un único comando.

Ejecutamos el comando **apt install net-tools**

Ahora, modificamos el fichero de inicio de sesión (message of the day) y el nombre del servidor para personalizarlo un poco, instalando primeramente con **apt install toilet**.

- **toilet --metal nocheweb > /etc/motd**
- **echo “nocheweb” > /etc/hostname**

Un tema de gran importancia para todo administrador son los permisos que tiene cada usuario ya que no se deben conceder permisos a la ligera porque esto genera una gran brecha de seguridad en el sistema. Comenzaremos concediendo todos los permisos al grupo root, ya que en este grupo solo se deben encontrar aquellos usuarios que requieren de una gran cantidad de permisos y libertad para llevar a cabo sus tareas, uno de ellos es el propio usuario root. Esto se realiza modificando el archivo *‘/etc/sudoers*’ y añadiendo la siguiente línea: **root ALL=(ALL:ALL) ALL**

CONFIGURAR SERVIDOR DE SSH

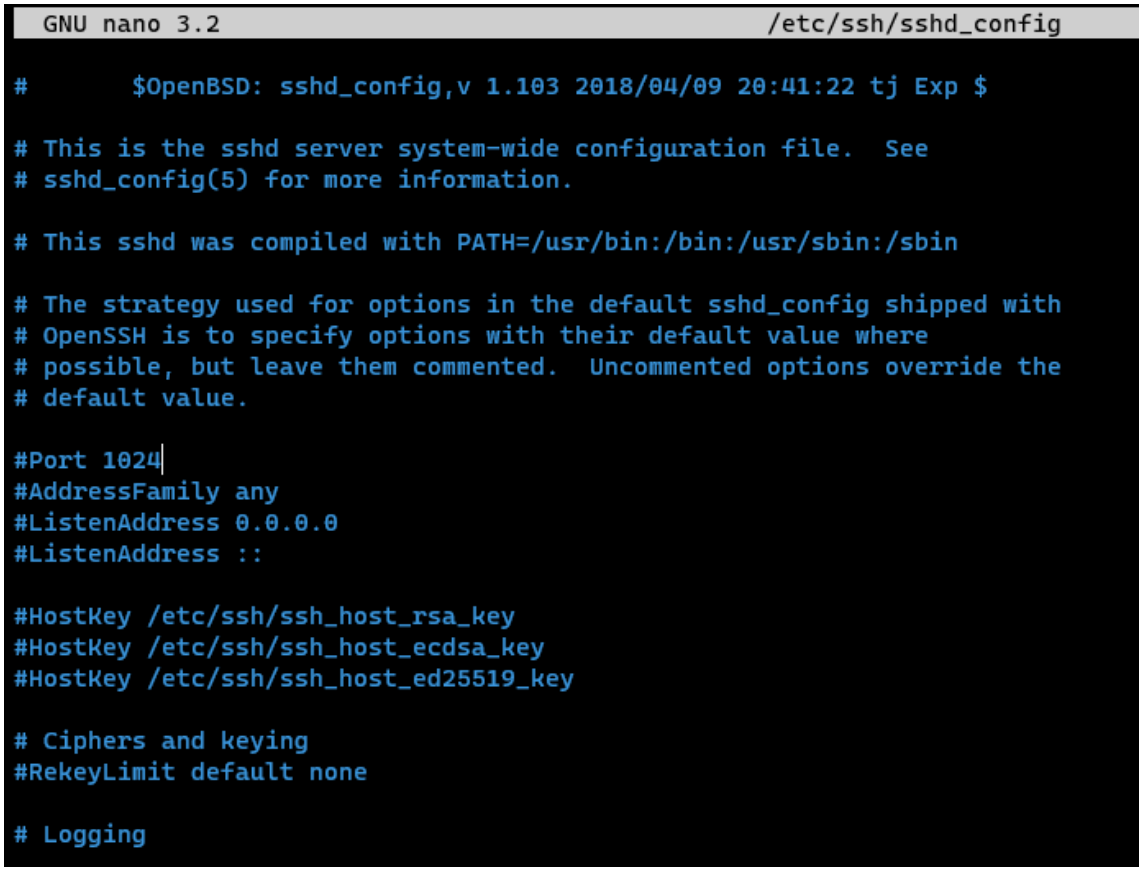
Hemos decidido usar el protocolo de administración remota SSH ya que permite autenticar un usuario remoto y garantiza un cifrado completo de los datos intercambiados.

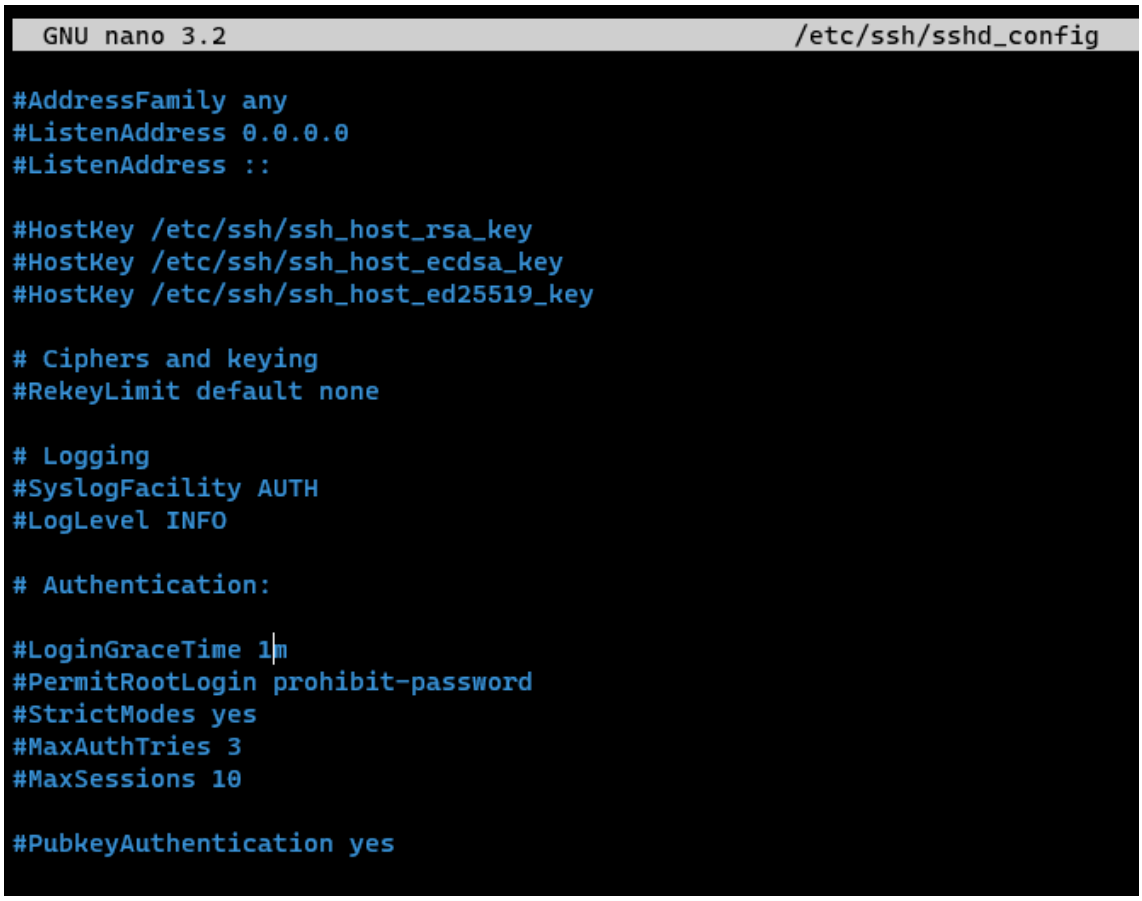
Lo primero es instalarlo: **apt install openssh-server**

Ahora realizaremos una serie de configuraciones para aumentar la seguridad de este protocolo, esto se realizará modificando el fichero *'/etc/sshd_config*'.

- Cambiar el puerto al cual se va a hacer referencia el protocolo ssh, por defecto es el 22, nosotros lo asignaremos al 1024.
- Después modificamos el tiempo de gracia para que solo haya un minuto para introducir la contraseña.
- Por último, cambiaremos el número de intentos a tres, cuando por defecto estaba en seis.

Descomentar las opciones, guardar el fichero y después reiniciar el servicio con **systemctl restart ssh.service**. Imágenes de los cambios realizados:





CONFIGURAR SERVIDOR WEB

Para ofrecer el contenido de nuestro sistema a los usuarios hemos decidido hacerlo mediante Apache 2 que es un servidor web de código abierto que permite ofrecer contenido web a través de Internet.

Lo primero es instalar el paquete: **apt-get install apache2**

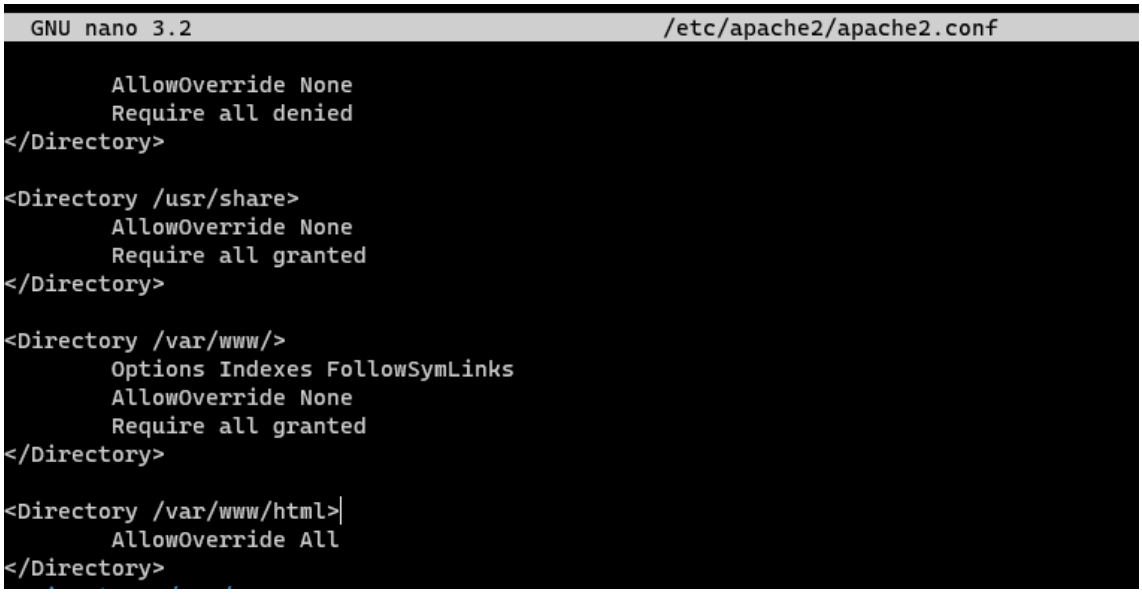
Para una mayor seguridad hemos usado el plugin suExec custom de apache: **apt install apache2-suexec-custom**

Después, añadiremos una capa extra de seguridad a la encriptación de los datos SSL mediante el protocolo HTTPS que es una evolución de HTTP. esto se consigue realizando los siguientes pasos:

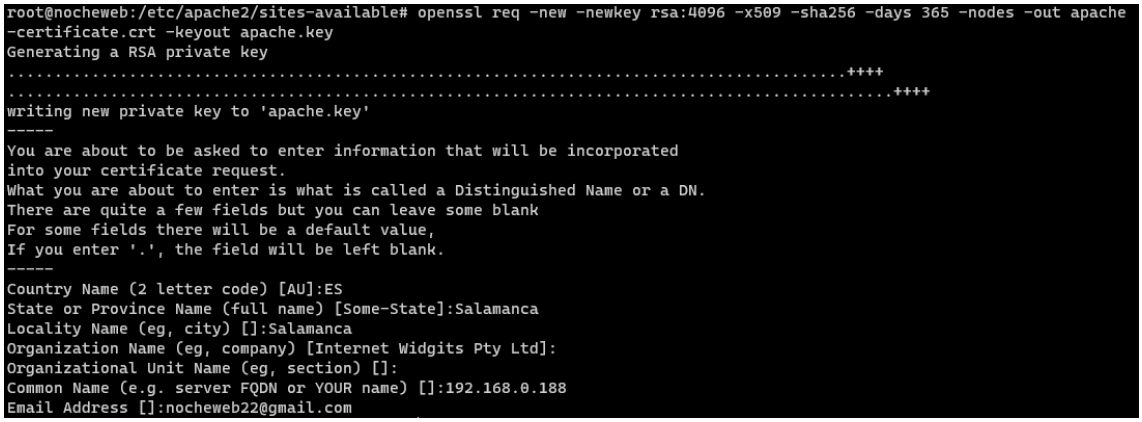
- **apt install openssl**
- **a2enmod ssl** → para asegurarnos de que el soporte para SSL/TLS está activado.
- **a2enmod rewrite** → para permitir hacer un “DNS” con la configuración del SSL.

- **systemctl restart apache2** → para que estos cambios se registren correctamente.

Posteriormente, modificamos el archivo de configuración de apache *'/etc/apache2/apache2.conf'* añadiendo las tres líneas indicadas en la imagen, esto permite que podamos modificar el directorio libremente.



Creamos un directorio en la carpeta propia de apache2 con **mkdir /etc/apache2/certificate** y accedemos a el **cd /etc/apache2/certificate**. Ahora tenemos que generar una key para que el certificado sea válido: **openssl req -new -newkey rsa:4096 -x509 -sha256 -days 365 -nodes -out apache-certificate.crt -keyout apache.key**



Modificamos el fichero *'/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. Debe quedar de la siguiente manera:

```
 GNU nano 3.2                                    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/certificate/apache-certificate.crt
        SSLCertificateKeyFile /etc/apache2/certificate/apache.key
</VirtualHost>
```

Después reiniciamos Apache2, abrimos el navegador y ya sea con la IP o con el nombre establecido en el SSL (https://nocheweb) podemos acceder a la página por defecto que ofrece Apache.

A continuación, activamos los siguientes módulos:

a2enmod cgi → habilita la ejecución de scripts en el servidor.

a2enmod userdir → habilita las páginas personales en el directorio home de cada usuario.

a2enmod suexec → para añadir mayor seguridad a la ejecución de estos scripts y que sólo puedan ser ejecutados por ciertos usuarios.

Apache administra las peticiones que se le realizan mediante el usuario www-data, para añadir mayor seguridad los permisos de ejecución de los CGI no los tendrá www-data sino un nuevo usuario, que será el que disponga de los permisos para llevar a cabo esta tarea, caparemos a este usuario del sistema para que no tenga directorio propio ni opción de login.

adduser --system --home /empty gyermo --shell=/bin/false → creamos nuestro querido usuario ‘gyermo’.

passwd gyermo compu → establecemos una contraseña para él.

Ahora le asignamos un grupo principal y le añadimos a shadow y root para que disponga de los permisos necesarios:

groupadd gyermo 232 → creamos el grupo.

usermod -g gyermo gyermo → le asignamos este grupo al usuario como grupo principal.

usermod -a -G shadow gyermo → le añadimos al grupo shadow para que pueda modificar los ficheros shadow y gestionar correctamente las altas y bajas de los usuarios.

usermod -a -G root gyermo → le añadimos al grupo root para que disponga de permisos de ejecución.

```
root@nocheweb:/# usermod -a -G shadow gyermo
root@nocheweb:/# usermod -a -G root gyermo
root@nocheweb:/# groups gyermo
gyermo : nogroup root shadow
root@nocheweb:/#
```

```
root@nocheweb:/var/www/html# groups gyermo
gyermo : gyermo root shadow
root@nocheweb:/var/www/html#
```

Después, lo añadimos al archivo *'/etc/sudoers'* como habíamos realizado previamente con root y modificamos los permisos a los ficheros necesarios para la creación de usuarios y sus directorios propios, damos el permiso de lectura y escritura al grupo del creador de los ficheros passwd, shadow, gshadow, groups y home, en este último, también el de ejecución.

```
 GNU nano 3.2                                  /etc/sudoers

#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
gyermo  ALL=(ALL:ALL) ALL
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
```

```
chmod g+w /etc/group
chmod g+w /etc/shadow
chmod g+w /etc/gshadow
chmod g+w /etc/passwd
chmod g+w /home/
chmod g+x /home/
```

Finalmente, modificamos de nuevo los siguientes archivos para que nuestro usuario se comunique de forma correcta con www-data y sepa dónde encontrar los scripts.

```
 GNU nano 3.2                                                    /etc/apache2/suexec/www-data

/usr/lib/cgi-bin
/var/www
public_html/cgi-bin
```

```
 GNU nano 3.2                                                  /etc/apache2/sites-available/000-default.conf

<VirtualHost *:80>
        RewriteEngine On
        RewriteCond %{HTTPS} !=on
        RewriteRule ^/?(.*) https://%{SERVER_NAME}/$1 [R=301,L]
</virtualhost>
<VirtualHost *:443>

        ServerName www.nocheweb.es
        ServerAdmin nocheweb@nocheweb

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

        DocumentRoot /var/www/html
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
        SSLEngine on
        SSLCertificateFile /etc/apache2/certificate/apache-certificate.crt
        SSLCertificateKeyFile /etc/apache2/certificate/apache.key
</VirtualHost>
```

En el fichero *'/000-default.conf'* hemos añadido la directiva SuExecUserGroup para indicar el usuario que va a ejecutar los scripts y a continuación el directorio en el que se van a almacenar los ficheros cgi, además de algunas órdenes para que reconozcan de forma exitosa los tipos de archivos y no se produzca error alguno.

GESTIÓN DE USUARIOS

Creamos dos grupos, uno para cada tipo de cuenta de usuario que debemos administrar. Les asignamos un GID específico para facilitar el registro de los usuarios.

groupadd -g 230 profesores

groupadd -g 231 estudiantes

```
profesores:x:230:
estudiantes:x:231:
```

Luego creamos el directorio apuntes en *'/var/www/html '* para que se pueda acceder a este desde la página web del servidor.

mkdir /var/www/html/apuntes

chown :profesores /var/www/html/apuntes → para que el grupo profesores sea el propietario.

chmod 755 /var/www/html/apuntes → para dar los permisos necesarios al resto de ficheros que se vayan creando en dicho directorio.

```
root@nocheweb:~# mkdir /var/www/html/apuntes
root@nocheweb:~# chown :profesores /var/www/html/apuntes/
```

```
root@nocheweb:~# chmod 775 /var/www/html/apuntes
```

chmod g+s apuntes/ → 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.

```
root@nocheweb:/var/www/html# chmod g+s apuntes/
root@nocheweb:/var/www/html# ls -l
total 16
drwxrwSr-x 2 root profesores  4096 May 25 15:45 apuntes
-rw-r--r-- 1 root root       10701 May 25 06:13 index.html
```

Adicionalmente creamos un enlace simbólico al fichero *'/var/log/apache/access.log'* para poder visualizar el log de los accesos de los usuarios, este enlace sólo podrá ser utilizado por el root.

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

DIRECTORIO /ETC/SKEL

En este directorio meteremos los archivos mencionados a continuación para que todos los usuarios al registrarse dispongan de ellos en su home.

- enlace simbólico a la carpeta de apuntes, se realiza con el siguiente comando: **ln -s /var/www/html/apuntes /etc/skel**

- documento de texto con las condiciones del servidor
- 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.

BASE DE DATOS (MARIADB)

Como base de datos permanente hemos decidido usar MariaDB debido a que es de software libre, a su potencia y a su lenguaje de sintaxis SQL.

Primero instalamos el siguiente paquete: **apt install mariadb-server mariadb-client**

Después ejecutamos el script **mysql_secure_installation** que mejora la seguridad en la instalación mediante el establecimiento de una contraseña (root) para las cuentas root, también permite eliminar las cuentas root que son accesibles desde fuera del local host y eliminar cuentas anónimas.

Tenemos que crear un nuevo usuario en la base de datos, ya que estos son independientes de los usuarios del propio sistema.

```
MariaDB [(none)]> create user 'administrador'@'localhost' identified by 'admin';
```

A continuación, creamos la base de datos, a la que llamaremos ‘nocheweb’.

```
MariaDB [(none)]> create database nocheweb;
```

Le asignamos permisos al usuario ‘administrador’ a todas las tablas que contenga la base de datos ‘nocheweb’.

```
MariaDB [(none)]> grant all privileges on nocheweb.* to 'administrador'@'localhost';
```

Creamos una tabla en ‘nocheweb’ llamada ‘users’ con los campos que se indican a continuación.

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);

```
MariaDB [nocheweb]> 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 (0.000 sec)
```

Y por último, registramos al usuario ‘nocheweb’ en esta:

insert into users(id,password ,name, surname, state, email, local_email, is_admin, role) values (“nocheweb”, ”YWRtaW4=”, ”administrador”, ”nocheweb”, 2, ”nocheweb2022@gmail.com”, “nocheweb@nocheweb”, 1, 2);

CONFIGURAR QUOTAS

Para ellos seguiremos los siguientes pasos explicados a lo largo de la asignatura:

- Instalamos el paquete cuota con **apt-get install quota**
- Abrimos el archivo ‘*/etc/fstab’* con un editor de texto
- Añadimos defaults, usrquota, grpquota. En nuestro sistema lo tenemos que hacer en el sistema de ficheros /home ya que al instalar la máquina decidimos hacer esta partición por los motivos mencionados anteriormente.

```
 GNU nano 3.2                                    /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=54c348c6-73ab-4c2f-a99c-ec848246cb6b /               ext4    errors=remount-ro 0       1
# /home was on /dev/sda6 during installation
UUID=33103d4d-3e50-4ac5-9e9f-1a9f92f5efc7 /home           ext4    defaults,usrquota,grpquota        0       2
# swap was on /dev/sda5 during installation
UUID=93cd7df4-d79f-425a-a260-b7b2d0f461cd none            swap    sw              0       0
/dev/sr0        /media/cdrom0   udf,iso9660 user,noauto     0       0
```

- Montamos el sistema de fichero otra vez mediante **mount -o remount /home**

Para comprobar que están correctamente instaladas podemos ejecutar uno de los muchos comandos para revisar las cuotas, en nuestro caso escogimos **quotacheck -ugmv /home**

```
root@nocheweb:~# quotacheck -ugmv /home
quotacheck: Your kernel probably supports journaled quota but you are not using it. Consider switching to journaled quot
a to avoid running quotacheck after an unclean shutdown.
quotacheck: Scanning /dev/sda6 [/home] done
quotacheck: Cannot stat old user quota file /home/aquota.user: No such file or directory. Usage will not be subtracted.
quotacheck: Cannot stat old group quota file /home/aquota.group: No such file or directory. Usage will not be subtracted
.
quotacheck: Cannot stat old user quota file /home/aquota.user: No such file or directory. Usage will not be subtracted.
quotacheck: Cannot stat old group quota file /home/aquota.group: No such file or directory. Usage will not be subtracted
.
quotacheck: Checked 82 directories and 51 files
quotacheck: Old file not found.
quotacheck: Old file not found.
```

Como podemos observar en la imágen encontramos varios errores porque no se han creado los archivos necesarios de cuotas ya que aún no están activas.

Procedemos a activarlas en la partición /home con **quotaon -ugv /home**

```
root@nocheweb:~# quotaon -ugv /home
/dev/sda6 [/home]: group quotas turned on
/dev/sda6 [/home]: user quotas turned on
root@nocheweb:~#
```

ELIMINAR USUARIOS QUE NO HAYAN CONFIRMADO

Para mantener nuestro sistema actualizado y optimizado debemos eliminar los datos que no son necesarios, uno de estos datos es eliminar la información de aquellos usuarios que no han confirmado su cuenta.

Primero creamos el archivo crontab para con **crontab -u root -e** y después añadir la siguiente línea:

```
# m h  dom mon dow   command
0 4 * * * /usr/bin/perl /root/del_user.pl
```

Finalmente, poner el archivo ‘del_user.pl’ en la ruta indicada.

COPIAS DE SEGURIDAD

Para aumentar la seguridad y fiabilidad de nuestro sistema debemos llevar a cabo copias de seguridad de forma periódica, estas se guardarán en el sistema y en una nube, para ello vamos a hacer uso de la herramienta rclone.

La instalamos y completamos su configuración: **apt install rclone**

Posteriormente creamos dos directorios, en uno se van a situar las copias de seguridad, mientras que el otro va a ser de respaldo:

mkdir /nw_back → fichero que almacena las copias de seguridad.

mkdir /nw_back_back → fichero backup del anterior.

En ‘*/nw_back*’ añadimos los scripts necesarios para la realización de las copias, estos son ‘nw_back.sh’ y ‘nw_back.pl’.

Y acto seguido usando **crontab -u root -e** añadimos al fichero de tareas periódicas la ejecución del script ‘nw_back.sh’ de la siguiente forma:

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

MONITORIZACIÓN

Esta tarea la realizaremos mediante la herramienta acct que permite generar informes sobre los tiempos de conexión de los usuarios y ejecución de los procesos.

Lo instalamos: **apt install acct**

Se activa con el siguiente comando: **accton on**

Creamos el directorio ‘acct’ en el que se van a guardar todos los archivos relacionados con el monitoreo, este se encuentra en el directorio del usuario root **mkdir /root/acct** e incluimos los ficheros ‘acct.sh’ y ‘acct.pl’.

Añadir al fichero de tareas periódicas la entrada para la ejecución de ‘acct.sh’:

```
0 4 * * * /usr/bin/perl /root/del_user.pl
0 4 * * * /bin/bash /nw_back/nw_back.sh
0 4 * * * /bin/bash /root/acct/acct.sh
```

Además, vamos a instalar un software de monitoreo opensource que nos facilite la interpretación de los datos. Ejecutamos **apt install monitorix** y accedemos a ‘ip:8080/monitorix’ para comprobar que ha sido instalado correctamente.

AVISO ROOT

Relacionado con la monitorización del sistema debemos avisar al administrador del inicio de los usuarios, para ello, modificamos el fichero de configuración que está en el directorio home del root ‘.bashrc’. Este fichero se ejecuta cada vez que se va a iniciar una sesión. Añadimos la siguiente línea **perl /root/aviso.pl**

SFTP

Para realizar el enjaulado de los usuarios en su directorio /home cuando acceden a su directorio personal de forma remota usaremos el servidor Vsftpd. Lo instalamos con **apt install vsftpd**

Modificamos el archivo de configuración que se encuentra en *‘/etc/vsftpd.conf’* y finalmente reiniciamos el servicio **systemctl restart vsftpd.service**

También tenemos que crear un enlace simbólico hacia vsfpd.log que genera el propio software → ln -s /var/log/vsftpd.log /vsftp.log

CORREO ELECTRÓNICO

Nos decantamos por usar Postflix para que se encargue del envío de correos electrónicos, Dovecot como servidor de correos para la entrega de estos y Roundcube como webmail para visualizar los mensajes.

Instalamos ambos paquetes:

apt install postfix → Con la opción de ‘servicio de internet’

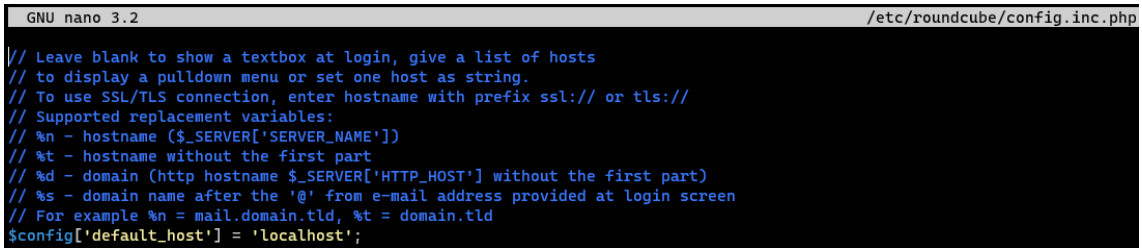
apt install dovecot-imapd

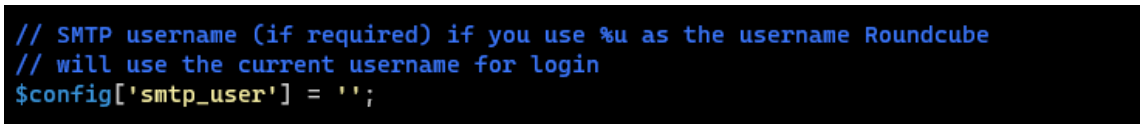
apt install roundcube

Después, pasamos a configurarlos. En el fichero ‘*/etc/postfix/main.cf’* añadimos el tamaño de los mensajes y el buzón

```
  GNU nano 3.2                                                    /etc/postfix/main.cf

# See /usr/share/postfix/main.cf.dist for a commented, more complete version


# Debian specific:  Specifying a file name will cause the first
# line of that file to be used as the name.  The Debian default
# is /etc/mailname.
#myorigin = /etc/mailname

smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

readme_directory = no

# See http://www.postfix.org/COMPATIBILITY_README.html -- default to 2 on
# fresh installs.
compatibility_level = 2



# 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_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for
# information on enabling SSL in the smtp client.

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination
myhostname = nocheweb.technicolor.net
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = $myhostname, nocheweb, 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
```

Modificamos el archivo ‘*/etc/roundcube/config.inc.php’* cambiando defult_host de “” a localhost y en el apartado de smtp user a vacío para poder enviar correos desde la propia plataforma.





Por último, creamos un enlace simbólico para que se pueda acceder desde la web:

ln -s /usr/share/roundcube /var/html/webMail

SCRIPTS

Para realizar todas las tareas que hemos descrito se han creado varios scripts de código Perl. En ellos hemos usado librerías de CPAN, por lo que previamente hemos instalado el paquete de CPAN con **apt install cpanminus**. Y después los comandos **cpanm install nombreLibreria**.

También usamos la librería de PAM, instalada con **apt-get install libauthen-simple-pam-perl**, para comprobar el login de los usuarios del sistema.

Algunos de los módulos usados son:

- 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

Funcionalidad principal de los archivos:

- acct.pl → se encarga de crear un archivo con las estadísticas del sistema y enviarlo al correo del administrador.
- act_pag.cgi → permite activar la página personal.
- activacion.cgi → para finalizar correctamente el proceso de registro de los usuarios.
- aviso.pl → avisa al administrador de los inicios de sesión.
- cambiar_datos.cgi → modificar los datos del usuario (excepto contraseña).
- cambiar_pwd.cgi → modificar la contraseña del usuario.
- cerrar_sesion.cgi → finalizar la sesión de forma prudente y segura.
- del_user.cgi → eliminar usuarios que no hayan finalizado el registro de su cuenta.
- desact_pag.cgi → permite desactivar la página personal.
- eliminar.cgi → eliminar de forma permanente a un usuario.
- login.cgi → iniciar sesión mediante username y contraseña.
- nw_back.pl → realizar copia de seguridad.
- redirect_pag.cgi y redirect.cgi → controlar la navegación entre las pantallas de los usuarios.
- registro.cgi → permite dar de alta a nuevos usuarios en el sistema.
- rem_pwd.cgi → permite la solicitud de recuerdo de contraseña.
- services.cgi → indica a los usuarios el estado de los servicios que ofrece el sistema.

Tenemos que darle los permisos de ejecución y cambiar el propietario de la carpeta al usuario del suexec:

chown -R gyermo:gyermo /usr/lib/cgi-bin/ → con la opción menos r para que se aplique a los ficheros de dentro del directorio. Una vez creados los ficheros debemos cambiar su propietario con **chmod a+X nombre.cgi**.

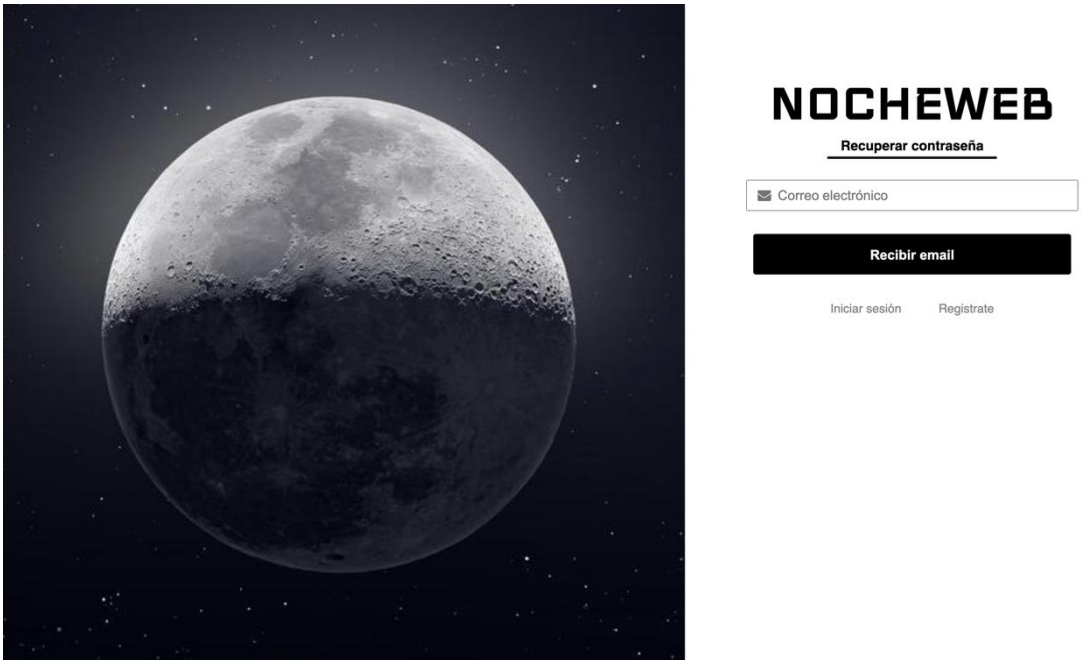
HTML

Hemos diseñado las siguientes vistas gráfica para que los usuarios interactúen con el sistema de forma simple y cómoda:

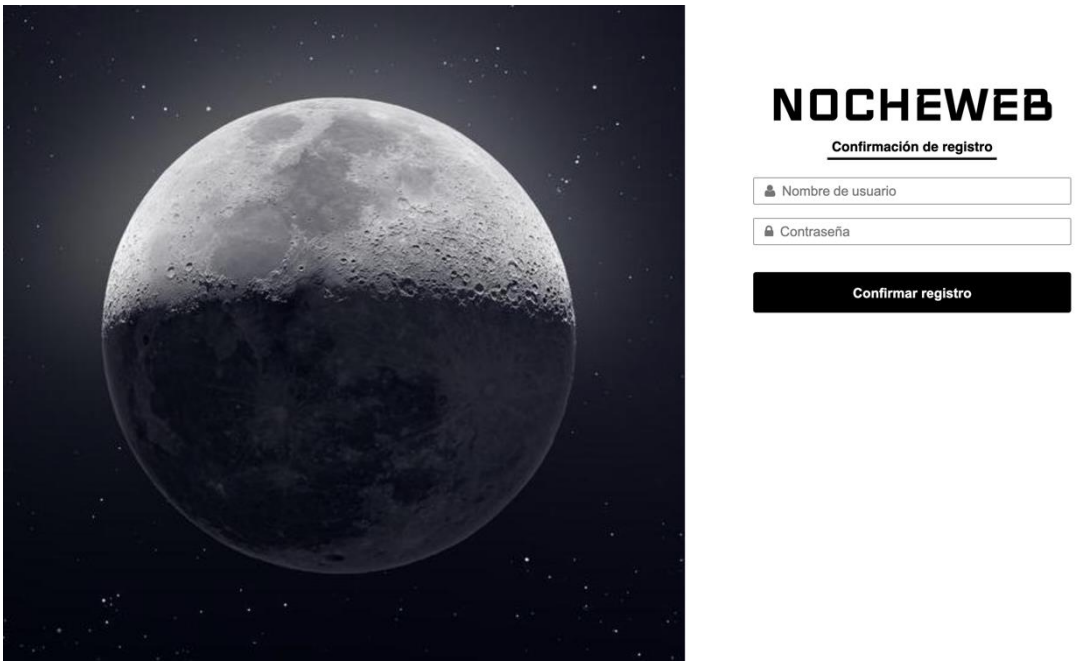
- Vista para iniciar sesión: ‘index.html’

- Vista para registrarse: ‘registro.html’

- Vista para recordar contraseña: ‘rem_pwd.html’



- Vista para confirmar creación de cuenta: ‘activacion.html’



- Vista bienvenida: ‘profile.html’

- Vista de administrador: ‘admin.html’

- Vista de profesor y estudiante: ‘profesor.html’ y ‘estudiante.html’

- Vista de configuración: ‘settings.html’

WEB PERSONAL

Usaremos la herramienta WordPress de código abierto para crear los blogs de cada usuario.

Lo primero es descargar el archivo con el comando:

wget https://wordpress.org/latest.tar.gz

Lo 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

Una vez Wordpress está correctamente instalado configuramos su base de datos, accedemos a MariaDB y creamos la base de datos 'wordpress':

CREATE DATABASE wpdb DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

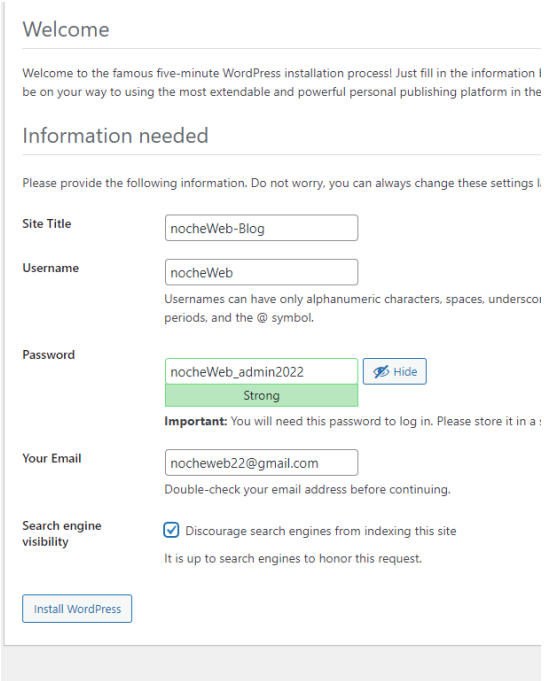
Después creamos al usuario 'wpuser' y le damos permisos sobre la nueva base de datos:

CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'root';

GRANT ALL PRIVILEGES ON wpdb.* TO 'wpuser'@'localhost';

quit;

Finalmente accedemos a 'direccionIP/wordpress' para visualizar la interfaz de la web de WordPress verificando que la instalación y configuración se han realizado correctamente.



MOODLE

Lo primero es descargar el archivo con el comando:

wget https://download.moodle.org/stable38/moodle-latest-38.tgz

Lo descomprimimos:

tar -zxvf moodle-latest-38.tgz

Copiamos el directorio moodle en *'/var/www/htm'l* de forma recursiva:

cp /downloads/moodle /var/www/html/ -R

Cambiamos el propietario y los permisos para que www-data pueda acceder sin problemas:

chown www-data.www-data /var/www/html/moodle -R

chmod 0755 /var/www/html/moodle -R

Creamos el directorio *'/moodledata'* y volvemos a cambiar el propietario y los permisos:

mkdir /var/www/moodledata

chown www-data /var/www/moodledata -R

chmod 0770 /var/www/moodledata -R

Una vez Moodle está correctamente instalado tenemos que configurar su base de datos, accedemos a MariaDB y creamos la base de datos 'moodle':

CREATE DATABASE moodle DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

Después creamos al usuario 'moodle' y le damos permisos sobre la nueva base de datos:

CREATE USER 'moodle'@'localhost' IDENTIFIED WITH IDENTIFIED BY 'root';

GRANT ALL PRIVILEGES ON moodle.* TO 'moodle'@'localhost';

quit;

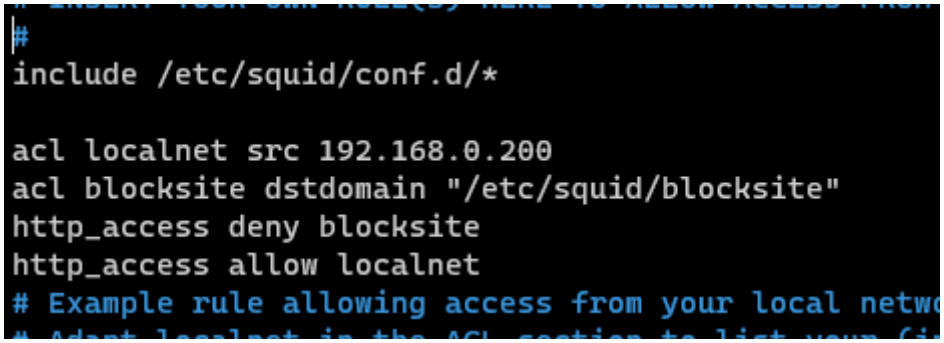
Finalmente accedemos a ‘direccionIP/moodle’ para visualizar la interfaz de la web de Moodle verificando que la instalación y configuración se han realizado correctamente.

SQUID

Para satisfacer las necesidades del profesor Fernando de bloquear el acceso a Facebook y Youtube haremos uso de Squid que permite cachear los datos que solicitan los usuarios y bloquear las direcciones deseadas.

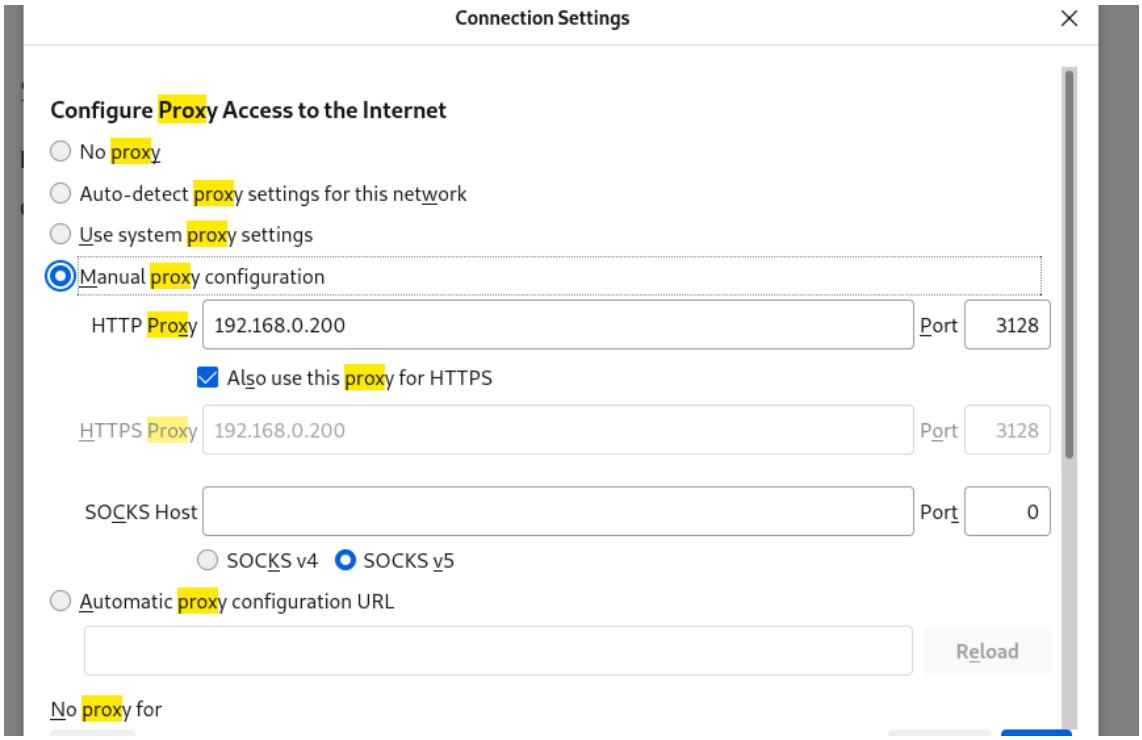
Lo instalamos con: **apt install squid**

Modificamos su fichero de configuración, este se encuentra en *‘/etc/squid/squid.conf’.*



Reiniciamos el servicio: **service squid restart**

Finalmente en el navegador modificamos la configuración del proxy del siguiente modo:


Connection Settings
Configure Proxy Access to the Internet
No proxy
Auto-detect proxy settings for this network
Use system proxy settings
Manual proxy configuration
HTTP Proxy
192.168.0.200
Port
3128
Also use this proxy for HTTPS
HTTPS Proxy
192.168.0.200
Port
3128
SOCKS Host
Port
0
SOCKS v4
SOCKS v5
Automatic proxy configuration URL
Reload
No proxy for

BIBLIOGRAFIA

https://codepre.com/en/que-es-build-essential-package-en-ubuntu-como-instalarlo.html

https://www.redhat.com/sysadmin/manage-permissions

https://linux.die.net/man/1/chmod

https://tipstricks.itmatrix.eu/customizing-apache2-suexec/

https://techexpert.tips/apache/enable-https-apache/

https://runebook.dev/es/docs/apache_http_server/suexec

https://blog.carreralinux.com.ar/2016/12/correr-apache-bajo-un-usuario-y-grupo-diferente/

https://blog.desdelinux.net/permisos-y-derechos-en-linux/

https://www.digitalocean.com/community/tutorials/how-to-use-suexec-in-apache-to-run-cgi-scripts-on-an-ubuntu-vps

https://www.w3docs.com/snippets/html/how-to-redirect-a-web-page-in-html.html

https://askubuntu.com/questions/481698/installing-authensimplepam-module-with-cpan-in-ubuntu-fails

https://unix.stackexchange.com/questions/157426/what-is-the-regex-to-validate-linux-users

https://www.howtoforge.com/tutorial/how-to-install-and-configure-vsftpd/

https://www.monitorix.org/doc-debian.html

https://techexpert.tips/es/moodle-es/instalacion-de-moodle-en-ubuntu-linux/

https://wordpress.org/support/article/how-to-install-wordpress/

https://metacpan.org/dist/App-cpanminus/view/bin/cpanm

https://metacpan.org/pod/Passwd::Unix

https://metacpan.org/pod/Sudo

https://metacpan.org/dist/PerlPowerTools/view/bin/mkdir

https://metacpan.org/pod/Linux::usermod

https://metacpan.org/pod/File::Copy::Recursive

https://metacpan.org/pod/Module::Rename

https://metacpan.org/pod/File::Path

https://metacpan.org/pod/CGI::Session

https://perldoc.perl.org/Sys::Hostname