

# ADMINISTRACIÓN DE SISTEMAS

*PRÁCTICA FINAL*



VNiVERSIDAD  
D SALAMANCA

---

CAMPUS DE EXCELENCIA INTERNACIONAL

**Enrique Mesonero Ronco 52417500V**  
**Guillermo Pascual Mangas 70911551K**



## ÍNDICE

<b>ÍNDICE.....</b>	<b>2</b>
<b>1. INTRODUCCIÓN.....</b>	<b>3</b>
<b>2. INSTALACIÓN DE LA IMAGEN DEL SERVIDOR.....</b>	<b>3</b>
<b>3. PRIMERA CONFIGURACIÓN.....</b>	<b>4</b>
<b>4. CONFIGURACIÓN DEL SERVIDOR SSH.....</b>	<b>6</b>
<b>5. SERVIDOR APACHE.....</b>	<b>8</b>
<b>6. PHP.....</b>	<b>13</b>
<b>7. PERL Y CPAN.....</b>	<b>14</b>
<b>8. GESTIÓN DE USUARIOS.....</b>	<b>14</b>
<b>9. ELIMINACIÓN DE USUARIOS QUE NO CONFIRMEN.....</b>	<b>16</b>
<b>10. DIRECTORIO /etc/skel.....</b>	<b>17</b>
<b>11. BASE DE DATOS.....</b>	<b>17</b>
<b>12. QUOTAS.....</b>	<b>20</b>
<b>13. COPIAS DE SEGURIDAD.....</b>	<b>20</b>
<b>14. MONITORIZACIÓN.....</b>	<b>22</b>
<b>15. TRIPWIRE.....</b>	<b>26</b>
<b>16. SFTP (USUARIOS).....</b>	<b>27</b>
<b>17. CORREO ELECTRÓNICO.....</b>	<b>27</b>
<b>18. BLOG WORDPRESS.....</b>	<b>29</b>
<b>19. CONFIGURACIÓN DE UN DOMINIO.....</b>	<b>31</b>
<b>20. TIENDA ONLINE.....</b>	<b>32</b>
<b>21. ALGUNAS PROPUESTAS DE MEJORA DEL SERVIDOR.....</b>	<b>33</b>

## 1. INTRODUCCIÓN

Se pide a los alumnos el diseño, la creación y configuración de un servidor LINUX, el cual deberá proporcionar infraestructura informática que permita el correcto funcionamiento de una Empresa Emergente del Sector Informático. La empresa debe ofrecer soporte técnico integral a todos los clientes, provisión de servicios de Internet, proporcionando a su vez la mayor seguridad, rendimiento y eficiencia posible.

En este informe se documentará detallada, clara y adecuadamente todo el proceso empleado para la creación del servidor. Este servidor permite realizar funciones básicas que como todo buen administrador de sistemas tiene que llevar a cabo.

## 2. INSTALACIÓN DE LA IMAGEN DEL SERVIDOR

Para la realización de esta práctica hemos decidido usar la versión 10 de Debian, la cual es estable en una máquina virtual. Hemos optado por la realización de 3 particiones para la instalación de la imagen. Estas particiones nos servirán para separar /home, /var y /tmp. Así logramos dividir por un lado la funcionalidad del servidor y por el otro la información de los usuarios.

Para conseguir que el sistema funcione por defecto en runlevel 3 (multiusuario con red) hemos utilizado el comando:

- **<systemctl set-default multi-user.target>**

Queremos asegurarnos que toda la instalación y configuración de nuestro servidor se complete correctamente, por lo cual, necesitamos convertirnos en superusuarios (root) el comando:

- **<su - >**

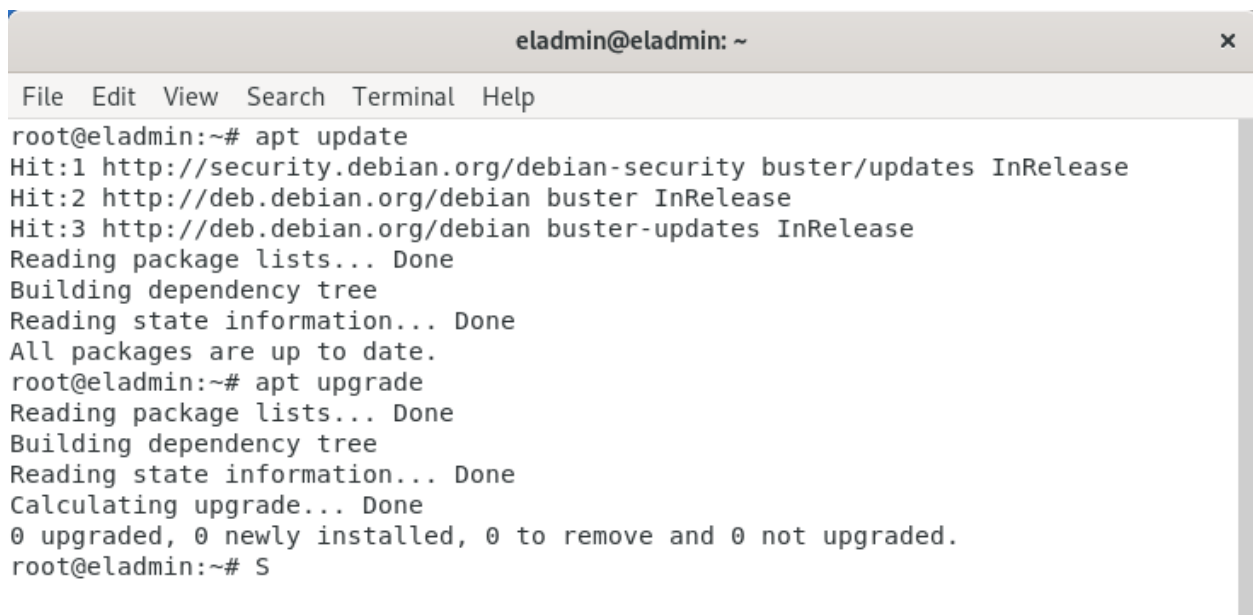
### 3. PRIMERA CONFIGURACIÓN

Una vez instalada la máquina virtual procederemos a realizar las primeras configuraciones del sistema, la configuración inicial. Para esto, empezaremos por realizar las actualizaciones pertinentes sobre todos los paquetes y acto seguido sobre el sistema operativo. Utilizaremos los siguientes comandos:

<apt update> : Nos permite actualizar los paquetes del sistema.

<apt upgrade> : Nos permite actualizar el sistema operativo.

A continuación se adjunta una imagen de la ejecución de los dos comandos.

A screenshot of a terminal window titled 'eladmin@eladmin: ~'. The terminal shows the execution of two commands: 'apt update' and 'apt upgrade'. The output for 'apt update' shows three hits from Debian security and buster repositories, followed by reading package lists and building a dependency tree, concluding that all packages are up to date. The output for 'apt upgrade' shows reading package lists, building a dependency tree, and calculating the upgrade, resulting in 0 packages being upgraded, installed, removed, or not upgraded.

```
eladmin@eladmin: ~
File Edit View Search Terminal Help
root@eladmin:~# apt update
Hit:1 http://security.debian.org/debian-security buster/updates InRelease
Hit:2 http://deb.debian.org/debian buster InRelease
Hit:3 http://deb.debian.org/debian buster-updates InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
root@eladmin:~# apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@eladmin:~# S
```

Figura 1

Para seguir con nuestra práctica necesitamos crear paquetes Debian. Por eso vamos a instalar el paquete “build-essential”. Para instalar este paquete utilizaremos el siguiente comando:

- **<apt-get install build-essential>**

Este paquete contiene una lista informativa de los paquetes considerados esenciales para la creación de paquetes Debian. Este paquete también depende de los paquetes que se encuentran en esa misma lista. Entre los paquetes de la lista destacaremos dos, gcc y g++. Estos son compiladores para c (gcc) y para c++ (g++).

Adjuntamos imagen de la instalación de los paquetes gcc y g++ dentro de la instalación de “build-essential”.

```
Setting up libc6-dev:amd64 (2.28-10+deb10u2) ...
Setting up libstdc++-8-dev:amd64 (8.3.0-6) ...
Setting up gcc-8 (8.3.0-6) ...
Setting up gcc (4:8.3.0-1) ...
Setting up g++-8 (8.3.0-6) ...
Setting up g++ (4:8.3.0-1) ...
```

Figura 2

También usaremos dpkg-dev, este nos proporciona las herramientas de desarrollo (entre ellas, dpkg-source) necesarias para desempaquetar y subir paquetes fuente de Debian.

Para que el servidor nos muestre un mensaje, cada vez que se inicie, deberemos modificar el fichero “/etc/motd” (message of the day). Para esto necesitamos instalar el paquete “toilet”, lo haremos con el comando:

- **<apt install toilet>**

Una vez instalado, para poder modificar el fichero usaremos el comando:

- **<toilet -metal -f future.tlf Hola admin! > /etc/motd>**

Por último, para finalizar con la configuración inicial, debemos establecer nuestra política de permisos. Solo deben poseer permisos los usuarios que se encuentren en el grupo de root. Para ello, modificaremos el fichero “/etc/sudoers” y añadiremos la línea:

**“root ALL=(ALL:ALL) ALL”**

A continuación pasaremos a la configuración del servidor ssh.

## 4. CONFIGURACIÓN DEL SERVIDOR SSH

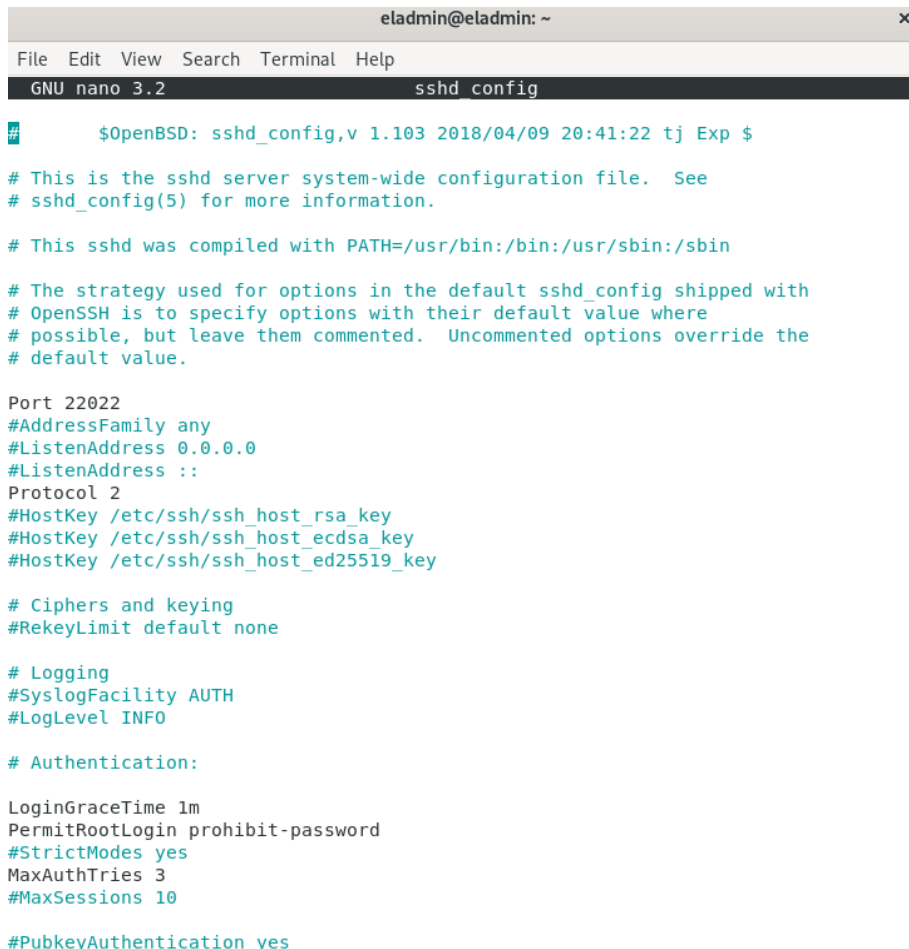
Para comenzar, debemos instalar el paquete net-tools. Este paquete nos ofrece una serie de herramientas útiles para la administración de redes, como puede ser los comandos netstat o ifconfig. Para instalar el paquete usaremos el comando:

- **<apt install net-tools>**

Una vez instalado el paquete, procederemos a instalar el servidor ssh. Este es utilizado para conexiones en remoto desde otros dispositivos. Este servidor nos permitirá autenticar los usuarios que se conectan a este gracias a una clave. El comando para instalarlo será:

- **<apt install openssh-server>**

Después, modificamos el fichero “/etc/ssh/sshd\_config” para brindar mayor seguridad a nuestro servidor. Adjuntamos foto del contenido del fichero:



```
eladmin@eladmin: ~
File Edit View Search Terminal Help
GNU nano 3.2 sshd_config

# $OpenBSD: sshd_config,v 1.103 2018/04/09 20:41:22 tj Exp $

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

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

# 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.

Port 22022
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
Protocol 2
#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
#MaxSessions 10

#PubkeyAuthentication yes
```

Figura 3

Las modificaciones que introduciremos serán las siguientes:

- Primero de todo, cambiaremos el *puerto* que por defecto viene (el 22). El puerto que usaremos es 22022. Usaremos este debido a que el puerto debe de ser mayor a 1024, puesto que los servicios corren en puertos menores a este y no debemos de causar conflictos entre ellos.
- Establecemos el *protocolo* SSH 2, ya que es más seguro que el primero, al tener en consideración las vulnerabilidades del protocolo ssh 1.
- Lo siguiente que cambiaremos será establecer *LoginGraceTime* a 1 minuto o 60 segundos. LoginGraceTime será el tiempo que tendrá el usuario para poder hacer login, en caso de que no consiga hacer login se le cerrará la conexión.
- Otro cambio que deberemos cambiar será *MaxAuthTries* a 3, este parámetro se refiere al número máximo de intentos que tiene el usuario para hacer login.
- Por último, descomentamos *PermitRootLogin* y le damos el valor yes. Esto nos permite registrarnos como root desde otro ordenador y si queremos, podemos subir algún fichero a través del protocolo sftp.

Una vez hechas las modificaciones, reiniciamos el servicio ssh con el comando:

- **<systemctl restart ssh>**

Adjuntamos imagen de cuando reiniciamos el servicio:

```
systemd[1]: Starting OpenBSD Secure Shell server...  
sshd[10708]: Server listening on 0.0.0.0 port 22022.  
sshd[10708]: Server listening on :: port 22022.  
systemd[1]: Started OpenBSD Secure Shell server.
```

Figura 4

A continuación, adjuntamos imagen de cómo nos conectaremos al servidor.

```
C:\Users\Enrique>sftp -oPort=22022 root@127.0.0.1  
root@127.0.0.1's password:   
Connected to 127.0.0.1.  
sftp>   

```

Figura 5



## 5. SERVIDOR APACHE

El servidor Apache nos permite exponer nuestros ficheros en internet, o dicho de otra manera, sirve las webs alojadas en el servidor a diversos navegadores como Chrome, Firefox o Safari.

Primero de todo, necesitamos instalar apache en nuestro servidor, para ello usaremos el comando:

- **<apt-get install apache2>**

Simultáneamente para añadir más seguridad tendremos que instalar el plugin de suExec custom de Apache, el comando necesario para instalarlo es:

- **<apt install apache2-suexec-custom>**

También procederemos a instalar openssl. Esto nos permitirá crear una comunicación segura a través del protocolo HTTPS. Para instalarlo seguiremos los siguientes comandos:

- **<apt install openssl>**
- **<a2enmod ssl>** nos permite comprobar que el soporte SSL está activado.
- **<a2enmod rewrite>** nos permite realizar un DNS.

Una vez realizados estos comandos tenemos que reiniciar el servicio de apache para que los cambios se guarden correctamente. El comando para esto será:

- **<systemctl restart apache2>**

Acto seguido deberemos modificar el fichero de configuración de apache, el fichero **“/etc/apache2/apache2.conf”**. A éste habrá que añadirle tres líneas, indicando el directorio **“/var/www/html”** y dentro modificando el apartado **“AllowOverride All”**.

Ahora crearemos un directorio para el certificado ssl, este directorio lo crearemos mediante el comando:

- **<mkdir /etc/apache2/certificates>**

En este directorio crearemos, dentro de este directorio, una clave para que se pueda validar el certificado. Esta clave se hará mediante el comando:

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

Adjuntamos imagen de demostración de los pasos anteriores, junto con el certificado.

```
root@eladmin:/etc/apache2# mkdir certificates
root@eladmin:/etc/apache2# cd certificates/
root@eladmin:/etc/apache2/certificates# openssl req -new -newkey rsa:4096 -x509 -sha256 -days 365 -nodes -out apache-certificate.crt -keyout apache.key
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'apache.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Salamanca
Locality Name (eg, city) []:Salamanca
Organization Name (eg, company) [Internet Widgits Pty Ltd]:eladmin
Organizational Unit Name (eg, section) []:administracion
Common Name (e.g. server FQDN or YOUR name) []:Enrique
Email Address []:id00779874@usal.es
root@eladmin:/etc/apache2/certificates#
```

Figura 6

Acto seguido modificaremos el fichero “/etc/apache2/sites-enabled/000-default.conf”. En este fichero haremos que nuestro servidor funcione gracias al certificado que acabamos de crear.

En la foto siguiente mostramos como debe quedar:

```
<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/certificates/apache-certificate.crt
    SSLCertificateKeyFile /etc/apache2/certificates/apache.key
</VirtualHost>
```

Figura 7

Ahora accederemos al servidor, buscamos en el navegador la dirección: “<https://eladmin>”. Esta dirección está creada con el nombre establecido en SSL pero también se podría buscar buscando con la IP. Esta página web es la creada por defecto por Apache.

Incluimos imagen mostrando la página:

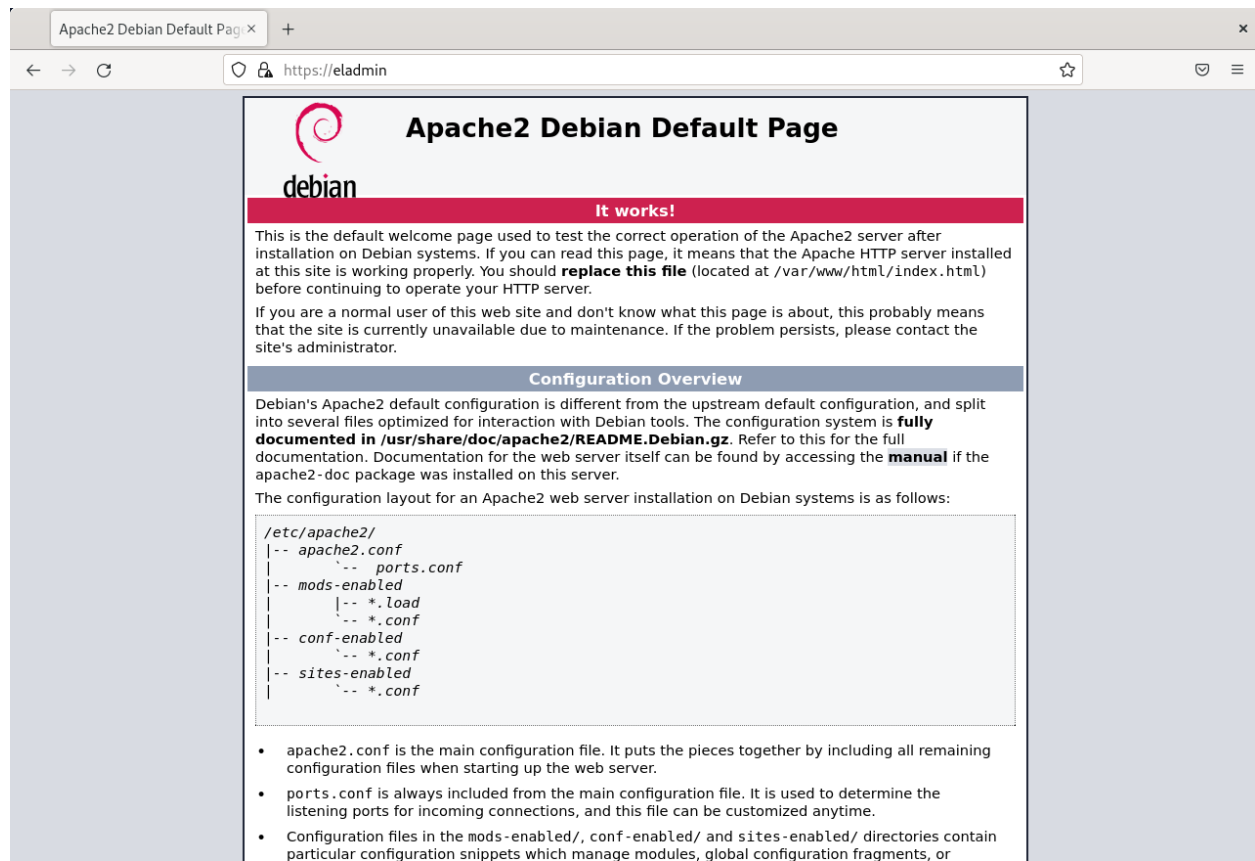


Figura 8

Debemos activar los siguientes módulos:

- “**a2enmod cgi**”: nos habilita la ejecución de scripts en el servidor.
- “**a2enmod userdir**”: nos habilita las páginas personales en el directorio home de cada usuario.
- “**a2enmod suexec**”: esto nos permite asegurar que algunos scripts solo puedan ser ejecutados por ciertos usuarios.

Apache administra las peticiones que se realizan mediante el usuario www-data, para obtener mayor seguridad los permisos de ejecución de los CGI no los tendrá www-data. En cambio los tendrá un nuevo usuario, será el que disponga de los permisos para llevar a cabo esta tarea. Por eso debemos restringir a este usuario del sistema para que no tenga directorio propio, ni tampoco opción de login.

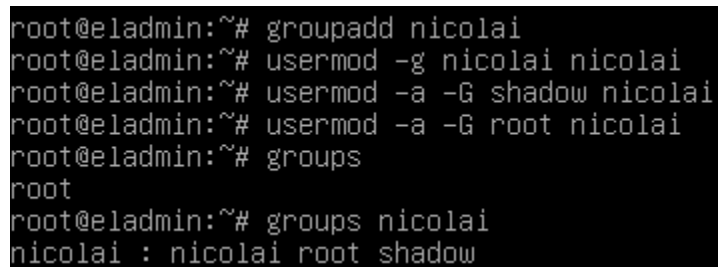
Para conseguir esto usaremos los siguientes comandos:

- **<adduser -system -home /empty nicolai -shell=/bin/false>** creamos un usuario llamado nicolai.
- **<passwd nicolai adminPass>** establecemos la contraseña de este usuario.

Una vez creado este usuario y su contraseña, hay que añadirlo a un grupo, introducirlo en shadow y en root para que tenga suficientes permisos:

- **<groupadd nicolai>** creamos el grupo.
- **<usermod -g nicolai nicolai>** le asignamos este grupo al usuario como grupo principal.
- **<usermod -a -G shadow nicolai>** 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 nicolai>** añadimos nicolai al grupo root para que disponga de los permisos necesarios para la ejecución.

Introducimos imagen de todos los comandos:



```
root@eladmin:~# groupadd nicolai
root@eladmin:~# usermod -g nicolai nicolai
root@eladmin:~# usermod -a -G shadow nicolai
root@eladmin:~# usermod -a -G root nicolai
root@eladmin:~# groups
root
root@eladmin:~# groups nicolai
nicolai : nicolai root shadow
```

Figura 9

Ahora añadiremos nuestro usuario nicolai al archivo “**/etc/sudoers**” y modificaremos los permisos a los ficheros necesarios para la creación de usuarios y sus directorios propios. Incluimos una imagen mostrando el fichero:

```
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
nicolai 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_dir /etc/sudoers.d
root ALL=(ALL:ALL) ALL
```

Figura 10

Damos los permisos de lectura y escritura al grupo del creador de los ficheros passwd, shadow, gshadow, groups, home y dentro de este último el de ejecución. Insertamos foto de este paso:

```
root@eladmin:~# chmod g+w /etc/group
root@eladmin:~# chmod g+w /etc/shadow
root@eladmin:~# chmod g+w /etc/gshadow
root@eladmin:~# chmod g+w /etc/passwd
root@eladmin:~# chmod g+w /home
root@eladmin:~# chmod g+x /home
```

Figura 11

A continuación, modificaremos el fichero “**www-data**” añadiendo la dirección “/usr/lib/cgi-bin” para que nuestro usuario pueda localizar los scripts.

```
GNU nano 3.2                               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.
```

Figura 12

Por último, en el fichero “/000-default.conf” debemos añadir la directiva SuExecUserGroup para indicar que el usuario va a ejecutar los scripts, debemos indicar también el directorio en el que se van a almacenar los ficheros cgi. A parte también se almacenarán algunas órdenes que reconozca de manera exitosa los tipos de archivos y de esta manera no se produzca error alguno.

```
<VirtualHost *:443>
    ServerName www.eladmin.es
    ServerAdmin eladmin@eladmin

    SuexecUserGroup nicolai nicolai
    <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
```

Figura 13

## 6. PHP

Instalaremos php para el funcionamiento de nuestra página web, buscamos la correcta interacción con nuestra base de datos. Por ellos instalaremos los siguientes comandos:

- **<apt-get install php>**
- **<apt-get install php-common>**
- **<apt-get install php-ssh2>**

## 7. PERL Y CPAN

Instalaremos tanto PERL como CPAN. Necesitamos PERL ya que lo usaremos en los diferentes scripts y CPAN nos brindará diversas librerías útiles para nosotros:

- **<apt-get install perl build-essential curl>**
- **<apt-get install cpan>**

Por último instalaremos las librerías:

- **<cpanm Email::Send::SMTP::Gmail>**
- **use strict;**
- **use warnings;**
- **use DBI;**
- **use Excel::Writer::XLSX;**
- **use Email::Sender::Simple qw(sendmail);**
- **use Email::Sender::Transport::SMTP::TLS;**
- **use MIME::Base64;**
- **use File::Slurp;**
- **use MIME::Lite;**
- **use Net::SMTP;**
- **use File::Copy;**
- **use File::Rotate::Backup;**

## 8. GESTIÓN DE USUARIOS

En cuanto a la gestión de usuarios, crearemos dos grupos. Uno de encargados y otro de clientes. Para ellos utilizaremos dos comandos:

- **<groupadd -g 1020 encargados>**
- **<groupadd -g 1021 clientes>**

Incluimos imagen de estos grupos juntos con sus respectivos gid:

```
encargados:x:1020:  
clientes:x:1021:
```

Figura 14

Acto seguido, crearemos un directorio “/var/www/html/servicios” con:

- **<mkdir /var/www/html/servicios>**

Este directorio se podrá acceder desde la página web del servidor, del cual será propietario el grupo de encargados. El comando para hacer propietario al grupo de encargados es:

- **<chown :encargados /var/www/html/servicios>**

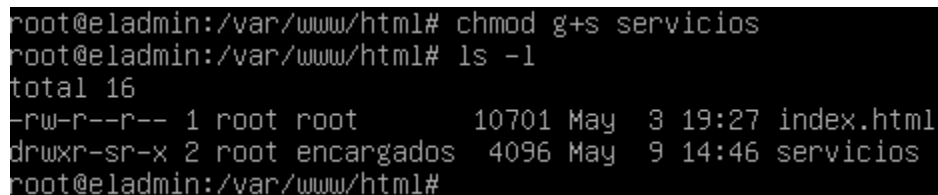
Asignamos los permisos necesarios para los ficheros que se creen dentro del directorio con el comando:

- **<chmod 755 /var/www/html/servicios>**

Además añadimos el bit setid para el grupo que los usuarios pertenecientes al mismo grupo puedan interactuar sobre los archivos sin problemas. Lo hacemos con el comando:

- **<chmod g+s servicios>**

Adjuntamos imagen:



```
root@eladmin:/var/www/html# chmod g+s servicios
root@eladmin:/var/www/html# ls -l
total 16
-rw-r--r-- 1 root root      10701 May  3 19:27 index.html
drwxr-sr-x 2 root encargados 4096 May  9 14:46 servicios
root@eladmin:/var/www/html#
```

Figura 15

Por último, crearemos un enlace simbólico al fichero de los accesos al servidor de apache. Hacemos esto para poder visualizar el log de los accesos de los usuarios. Enlace el cuál solo podrá ser utilizado por el usuario root. Este lo verá usando:

- **<ln -s /var/log/apache2/access.log /access.log>**



## 9. ELIMINACIÓN DE USUARIOS QUE NO CONFIRMEN

Otro apartado muy importante del cual se debe hablar en la gestión de usuario es saber eliminar aquellos usuarios que no hayan confirmado su cuenta. Esto tiene el objetivo de mantener el sistema actualizado y optimizado. Para ello, crearemos un archivo crontab con el comando:

- **<crontab -u root -e>**

En este archivo introducimos la siguiente información:

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
0 4 * * * /usr/bin/perl /root/del_user.pl
```

Figura 16

## 10.DIRECTORIO /etc/skel

Necesitamos los archivos necesarios para que se creen correctamente los archivos de los diferentes usuarios, estos archivos deben tener una estructura común, teniendo el directorio home de cada uno de los usuarios que se creen. En este directorio “/etc/skel” introduciremos los siguientes archivos:

- Enlace simbólico a la carpeta de servicios.
- Documento de texto con información de nuestro servidor.
- Posteriormente, un directorio con la página principal (una vez se haya creado).

## 11. BASE DE DATOS

En cuanto a la base de datos, hemos seleccionado la propuesta vista en clase, MariaDB. Esta es una base open source en MySQL.

Primero de todo, debemos instalar los paquetes necesarios. Estos serán mariadb-server y mariadb-client, para instalar estos paquete usaremos los comandos:

- **<apt install mariadb-server>**
- **<apt install mariadb-client>**

Una vez instalados, debemos ejecutar el script “mysql\_secure\_installation”. Este script nos permite:

- Establecer una contraseña para el root.
- Eliminar cuentas de usuarios anónimos.
- Eliminar bases de datos de prueba.
- Eliminar las cuentas root con acceso fuera del localhost.

Para acceder a la base de datos simplemente deberemos introducir el comando:

- **<mysql>**

Una vez dentro, creamos un usuario:

- **<CREATE USER ‘admin’@’localhost’ identified by ‘admin’>**

Acto seguido crearemos una base de datos:

- **<create database eladminDB>**

Debemos otorgar al user 'admin' los permisos para que este tenga acceso a todas las tablas de la base de datos. Lo haremos de la siguiente manera:

- **<grant all privileges on eladminDB.\* to 'admin'@'localhost'>**

Una vez hecho todos estos pasos procederemos a crear la tabla donde se almacenará la información de nuestros usuarios:

- **<CREATE TABLE IF NOT EXISTS users(  
id varchar(255) not null primary key,  
user varchar(255) not null,  
password varchar(255) not null  
nombre varchar(255) not null,  
apellidos varchar(255) not null,  
correo varchar(255) not null,  
cPostal varchar(255) not null  
telefono varchar(255) not null,  
ciudad varchar(255) not null  
is\_admin boolean not null default false);>**

Adjuntamos imagen de como se verá la tabla en mariaDB

```
MariaDB [(none)]> describe eladminDB.users;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id     | varchar(255)  | NO   | PRI | NULL    |       |
| user   | varchar(255)  | NO   |     | NULL    |       |
| password | varchar(255) | NO   |     | NULL    |       |
| nombre | varchar(255)  | NO   |     | NULL    |       |
| apellidos | varchar(255) | NO   |     | NULL    |       |
| correo | varchar(255)  | NO   |     | NULL    |       |
| cPostal | varchar(255) | NO   |     | NULL    |       |
| is_admin | tinyint(1)   | NO   |     | 0       |       |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.001 sec)

MariaDB [(none)]> select * from eladminDB.users;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | user      | password      | nombre | apellidos | correo                                     | cPostal | is_admin |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  | adminTest | adminTestPass | German | Palomares | germanP@eladminmail.com | 37001   | 1        |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [(none)]>
```

Figura 18

Como este proceso desde la terminal podría llegar a ser muy tedioso crearemos un fichero “**dump.sql**”. Este nos ayudará a crear las tablas y un usuario de prueba de manera sencilla.

Introducimos la tabla con el comando:

- **<mysql -p eladminDB < dump.sql>**

Ahora creamos el usuario, insertandolo en la tabla:

- **<INSERT TO users(id, user, password, nombre, apellidos, correo, cPostal, is\_admin) VALUES (“1”, “adminTest”, “adminTestPass”, “German”, “Palomares”, “[germanP@eladminmail.com](mailto:germanP@eladminmail.com)”, “37001”, 1)>**

## 12. QUOTAS

Para las Quotas necesitamos instalar el paquete de quota. Lo haremos con el comando:

- **<apt-get install quota>**

Después editamos el fichero “**fstab**” dentro del directorio “/etc”. Dentro de este fichero, incluiremos **usrquota** y **grpquota** en el apartado del sistema de ficheros **/home**.

Una vez instalado, remontamos el sistema de ficheros con el comando:

- **<mount -o remount /home>**

Activamos las cuotas de la partición con el comando:

- **<quotaon -ugv /home>**

Haremos una comprobación con:

- **<quotacheck -ugmv /home>**

El resultado nos saldrá tal y como en la imagen:

```
/dev/sda8 [/home]: group quotas turned on  
/dev/sda8 [/home]: user quotas turned on
```

Figura 19

## 13. COPIAS DE SEGURIDAD

Uno de los apartados más importantes en la administración de sistemas en el que nos hemos enfocado durante el curso han sido las copias de seguridad.

Estas copias de seguridad tienen varios cometidos, entre ellos, servir como backup en caso de destrucción del servidor. Realizaremos copias de seguridad periódicamente, estas se guardarán en nuestro sistema y en una nube. Para realizar dichas copias, utilizaremos la herramienta **rsync**.

```
0 4 * * * /bin/bash /sbin/dropbox.sh
```

Figura 20

Necesitamos añadir al fichero de tareas periódicas la ejecución del .sh (como hemos hecho previamente en eliminar usuarios no confirmados). Para ello usamos el comando:

- **<crontab -u root -e>**

Para que nuestras copias no solo se encuentren en nuestro sistema también utilizaremos Dropbox. Dropbox es un servicio de almacenamiento en la nube y que ofrece múltiples posibilidades para alojar nuestros archivos.

Hemos decidido realizar una configuración usando rsync y Dropbox. De esta manera, como hemos dicho antes, podremos tener nuestras copias de seguridad en la nube.

Primero instalaremos Dropbox en el sistema con:

- **<cd ~ && wget -O - "[https://www.dropbox.com/download?plat=lnx.x86\\_64](https://www.dropbox.com/download?plat=lnx.x86_64)" | tar xzf ->**
- **<~/dropbox-dist/dropboxd>** con esto lo linkeamos a nuestro servidor.

A continuación, debemos crear un perl para hacer las copias de seguridad, la ejecución periódica de este la añadimos a nuestro cron. Esto se hará con el script **"dropbox.sh"**.

Incluimos imagen de una copia de seguridad en Dropbox:

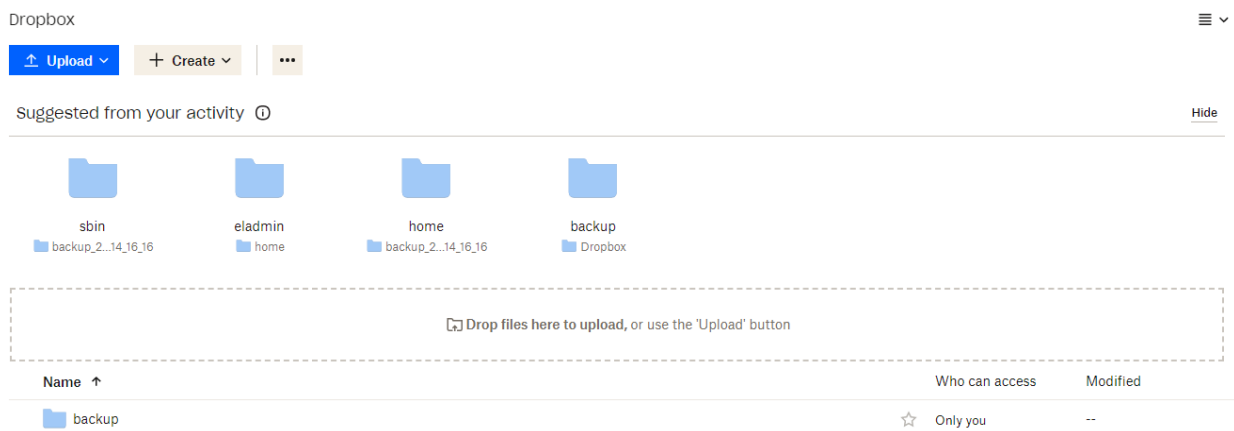


Figura 21

## 14.MONITORIZACIÓN

Para la monitorización de usuarios, utilizamos la herramienta acct. Esta permite generar informes de los tiempos de conexión de los usuarios. Para instalarla será tan sencillo como escribir el comando:

- **<apt install acct>**

Lo activamos con el comando:

- **<accton on>**

Creamos un directorio:

- **<mkdir /root/acct>** en este directorio es donde se almacenarán los ficheros relacionados a la monitorización.

También añadiremos dos ficheros al directorio, estos son “**acct.sh**” y “**acct.pl**”.

El .sh también los añadimos al fichero de actividades periódicas mediante:

- **<crontab -u root -e>**

Para la monitorización necesitamos instalar el “intérprete para humanos”, este es monitorix. Para instalarlo lo haremos con:

- **<apt install monitorix>**

Para acceder a el intérprete utilizaremos:

- **<ip:8080/monitorix>**

En la imagen siguiente adjuntamos como debe de realizarse la inclusión de los ficheros .sh y .pl:

```

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 4 * * * /usr/bin/perl /root/del_user.pl
0 4 * * * /bin/bash /sec_back/sec_back.sh
0 4 * * * /bin/bash /root/acct/acct.sh

crontab: installing new crontab
root@eladmin:~/acct#

```

Figura 22

Por otra parte, otro aspecto importante al monitorizar el sistema es que debemos avisar al administrador (root). Esto lo haremos incluyendo el script “avisar.pl” al fichero “.bashrc”, dentro del cual añadiremos la línea “**perl /root/avisar.pl**”.

Adjuntamos imagen del fichero “**.bashrc**”



```
GNU nano 3.2                                .bashrc

# ~/.bashrc: executed by bash(1) for non-login shells.

# Note: PS1 and umask are already set in /etc/profile. You should not
# need this unless you want different defaults for root.
# PS1='${debian_chroot:+($debian_chroot)}\h:\w\$ '
# umask 022

# You may uncomment the following lines if you want `ls' to be colorized:
# export LS_OPTIONS='--color=auto'
# eval "`dircolors`"
# alias ls='ls $LS_OPTIONS'
# alias ll='ls $LS_OPTIONS -l'
# alias l='ls $LS_OPTIONS -lA'
#
# Some more alias to avoid making mistakes:
# alias rm='rm -i'
# alias cp='cp -i'
# alias mv='mv -i'
perl /root/avisar.pl

root@eladmin:~#
```

Figura 23

También utilizaremos la herramienta “Logwatch”, la cuál se ejecuta diariamente, analiza los logs y envía un correo al administrador del sistema. Es útil monitorizar la actividad de los servidores y detectar posibles intentos de intrusión o consumo de recursos (entre otras posibles funciones). Instalaremos esta herramienta con:

- **<apt-get install logwatch>**

Configuraremos el mail al que queramos enviar la información (en este caso, el mail del administrador) modificando el archivo

“**/usr/share/logwatch/default.conf/logwatch.conf**”.

Por último, instalaremos auditd, un daemon encargado de monitorizar las llamadas al sistema, modificaciones de archivos, detección de anomalía e intrusos, etc. Para ello ejecutaremos:

- `<apt-get install auditd>`
- `<aureport>`

Adjuntamos imagen de auditd :

```
Summary Report
=====
Range of time in logs: 01/01/1970 01:00:00.000 - 05/18/2023 20:08:14.047
Selected time for report: 01/01/1970 01:00:00 - 05/18/2023 20:08:14.047
Number of changes in configuration: 3
Number of changes to accounts, groups, or roles: 0
Number of logins: 0
Number of failed logins: 0
Number of authentications: 0
Number of failed authentications: 0
Number of users: 1
Number of terminals: 1
Number of host names: 1
Number of executables: 1
Number of commands: 1
Number of files: 0
Number of AVC's: 0
Number of MAC events: 0
Number of failed syscalls: 0
Number of anomaly events: 0
Number of responses to anomaly events: 0
Number of crypto events: 0
Number of integrity events: 0
Number of virt events: 0
Number of keys: 0
Number of process IDs: 2
Number of events: 8
```

Figura 24

## 15. TRIPWIRE

“Tripwire es una herramienta cuya función específica es la detección de intrusos”. Tripwire es un programa de computador basado en Open Source, consistente en una herramienta de seguridad e integridad de los datos. Es útil para monitorizar y alertar de cambios en los ficheros de un sistema de ficheros. Análisis rutinarios de la integridad de una gran cantidad de archivos. Esta herramienta asume que todos los controles de seguridad han fallado, y que el sistema ya ha sido afectado. De esta manera se conseguirá alertar al administrador y tomar acciones con rapidez.

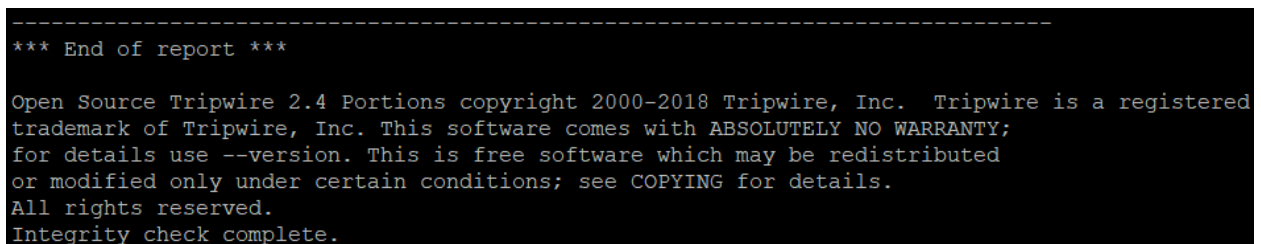
Para instalarla usaremos:

- `<apt-get install tripwire>`
- `<apt-get install mailutils>`

Para la configuración utilizaremos:

- `<twadmin -m P /etc/tripwire/twol.txt>`
- `<tripwire -init>`
- `<tripwire -check>` para comprobar que funciona correctamente.

Adjuntamos imagen:



```
-----  
*** End of report ***  
  
Open Source Tripwire 2.4 Portions copyright 2000-2018 Tripwire, Inc. Tripwire is a registered  
trademark of Tripwire, Inc. This software comes with ABSOLUTELY NO WARRANTY;  
for details use --version. This is free software which may be redistributed  
or modified only under certain conditions; see COPYING for details.  
All rights reserved.  
Integrity check complete.
```

Figura 25

## 16.SFTP (USUARIOS)

Otro aspecto importante de la gestión de usuarios es el enjaulado de los usuarios en su directorio /home cuando acceden a su directorio a través de sftp. Para ello, usaremos el servidor vsftpd. Este servidor los instalaremos con:

- **<apt install vsftpd>**

Modificaremos su archivo de configuración (“/etc/vsftpd.conf”) y reiniciamos el servicio con el comando:

- **<systemctl restart vsftpd.service>**

Crearemos también un enlace simbólico hacia el fichero “vsftpd.log” que genera el propio software.

## 17. CORREO ELECTRÓNICO

Para la realización del servidor del correo electrónico utilizaremos Postfix para el envío. En cuanto a la entrega usaremos Dovecot y Roundcube para visualizar los mensajes.

Instalaremos los programas con:

- **<apt-get install postfix>**
- **<apt-get install dovecot-imapd>**
- **<apt-get install roundcube>**

Para la configuración realizaremos:

- **/etc/postfix/main.cf** lo editaremos para el tamaño de los mensajes y del buzón
- **/etc/roundcube/config.inc.php** para el host (en nuestro caso, localhost) y envíos desde la propia plataforma.
- Crearemos enlace simbólico para acceder desde la web “**ln -s /usr/share/roundcube /var/www.html/webMailService**”.

## Imagen del fichero “/etc/postfix/main.cf”:

```
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 ESMTF $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 = eladmin.home
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = $myhostname, eladmin, 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 = +
```

Figura 26

## Imagen del fichero “/etc/roundcube/config.inc.php”:

```
GNU nano 3.2 /etc/roundcube/config.inc.php

// %s - domain name after the '@' from e-mail address provided at login screen
// For example %n = mail.domain.tld, %t = domain.tld
$config['default_host'] = 'localhost';

// SMTP server host (for sending mails).
// Enter hostname with prefix tls:// to use STARTTLS, or use
// prefix ssl:// to use the deprecated SSL over SMTP (aka SMTPS)
// Supported replacement variables:
// %h - user's IMAP hostname
// %n - hostname ($SERVER['SERVER_NAME'])
// %t - hostname without the first part
// %d - domain (http hostname $SERVER['HTTP_HOST'] without the first part)
// %z - IMAP domain (IMAP hostname without the first part)
// For example %n = mail.domain.tld, %t = domain.tld
$config['smtp_server'] = 'localhost';

// SMTP port (default is 25; use 587 for STARTTLS or 465 for the
// deprecated SSL over SMTP (aka SMTPS))
$config['smtp_port'] = 25;

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

// SMTP password (if required) if you use %p as the password Roundcube
// will use the current user's password for login
$config['smtp_pass'] = '%p';

// provide an URL where a user can get support for this Roundcube installation
// PLEASE DO NOT LINK TO THE ROUND_CUBE.NET WEBSITE HERE!
$config['support_url'] = '';

root@eladmin:~# _
```

Figura 27

## 18.BLOG WORDPRESS

Primero deberemos crear una base de datos, para esto usaremos los comandos:

- **<create database wordpress>**
- **<CREATE USER 'wordpress'@'localhost' IDENTIFIED BY 'wordpress';>** creamos un usuario.
- **<GRANT ALL PRIVILEGES ON \*.\* TO 'wordpress'@'localhost';>**
- **<Flush privileges>** en caso de que los privilegios no se hayan refrescado.

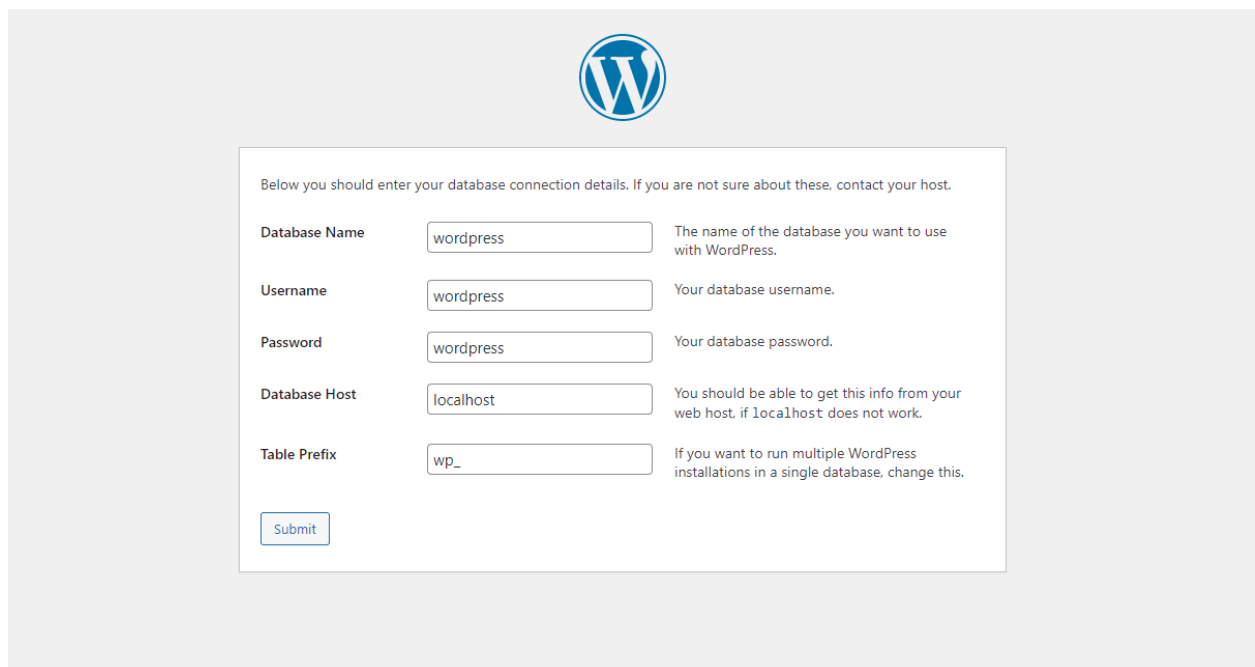
Acto seguido descargamos la última versión de wordpress:

- **<wget https://wordpress.org/latest.tar.gz>**

Este paquete tendremos que descomprimirlo en **“/var/www/html”**. Después debemos cambiar el propietario junto con los permisos para que www-data pueda acceder sin problemas. Para hacer esto escribiremos el siguiente comando:

- **<chown www-data.www-data /var/www/html/wordpress -R>**

Realizamos la configuración de word de wordpress siguiendo los pasos una vez accedemos a **“/var/www/html/wordpress”** en el navegador.



The screenshot shows the WordPress installation database configuration screen. At the top is the WordPress logo. Below it is a text box with the instruction: "Below you should enter your database connection details. If you are not sure about these, contact your host." The form contains five input fields, each with a label and a description:

Field	Value	Description
Database Name	wordpress	The name of the database you want to use with WordPress.
Username	wordpress	Your database username.
Password	wordpress	Your database password.
Database Host	localhost	You should be able to get this info from your web host, if localhost does not work.
Table Prefix	wp_	If you want to run multiple WordPress installations in a single database, change this.

At the bottom left of the form is a "Submit" button.

Figura 28

Creamos el fichero “**wp-config.php**”. Una vez introducido el nombre de la base de datos de wordpress, usuario, contraseña, correo, host y prefijo de la tabla, deberemos pegar el contenido que nos muestra la página de wordpress.

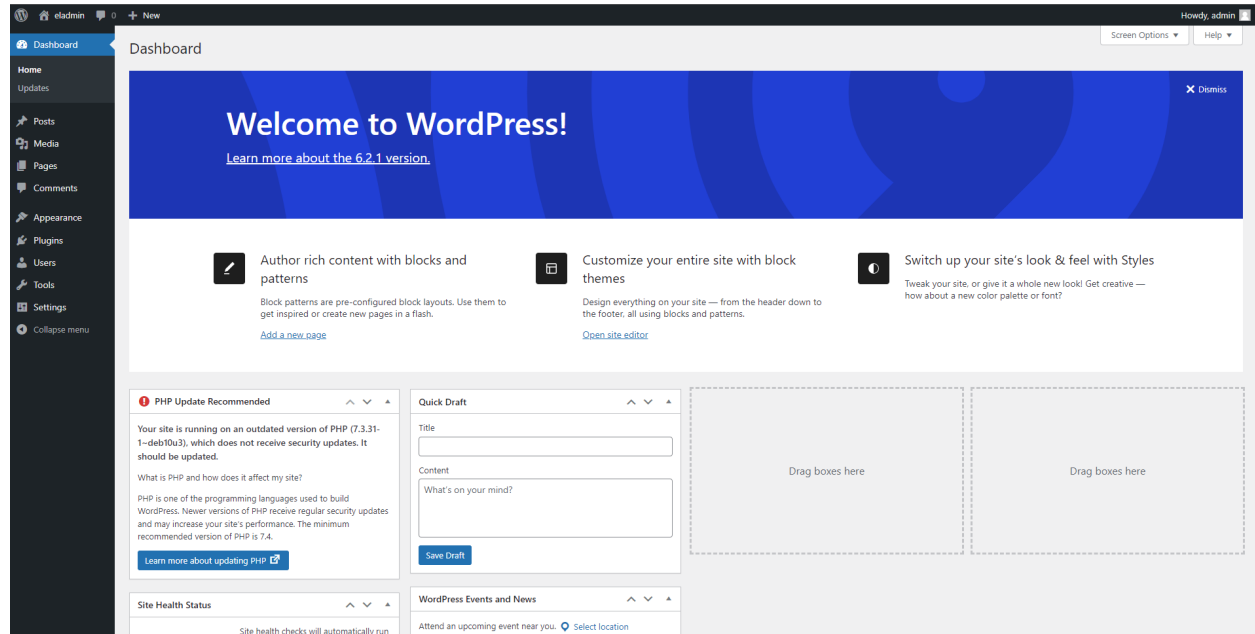


Figura 29

## 19.CONFIGURACIÓN DE UN DOMINIO

Para nuestra página web, hemos decidido utilizar Duck DNS para establecer un dominio y poder acceder a la web sin la necesidad de utilizar la IP, facilitando el acceso a los usuarios.

<https://eladminusal.duckdns.org/>

Incluimos imagen del dominio:

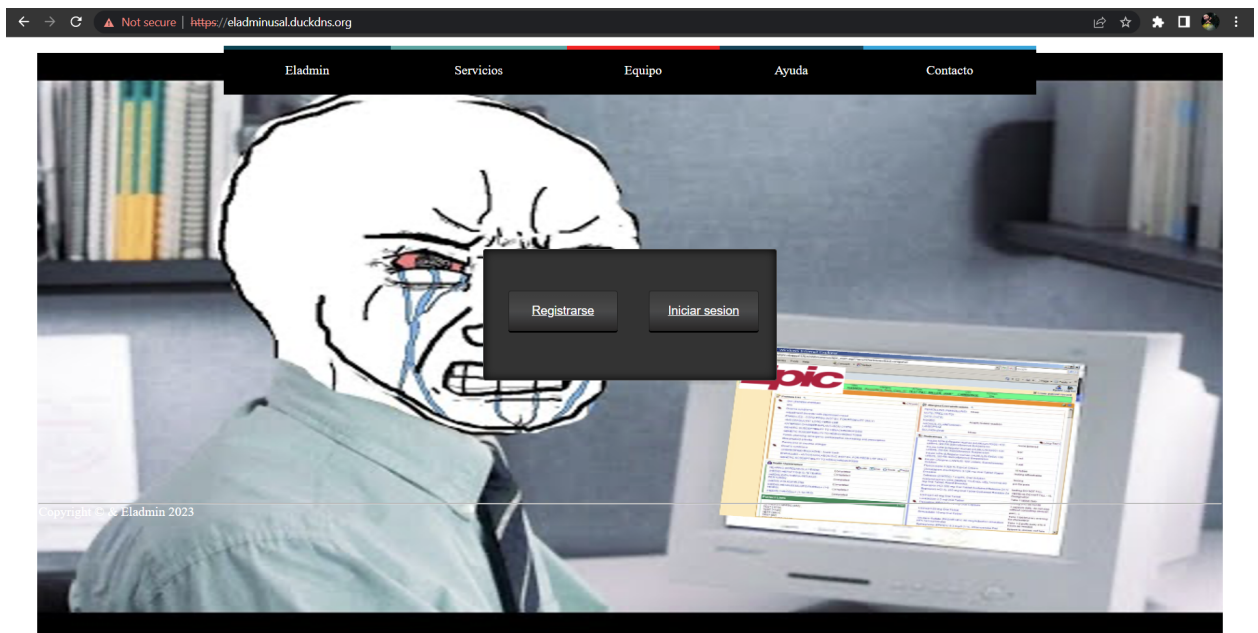


Figura 30



## 20. TIENDA ONLINE

Para la instalación y configuración en caso de que se requiera de una tienda online, utilizaremos Prestashop. Como Prestashop se basa en un servidor apache como nuestra web, y por comodidad de instalación y configuración, vamos a utilizar Docker para ello.

Lo primero de todo, será descargar docker y crear la preconfiguración, siguiendo la guía de la web:

- **<sudo apt-get update>**
- **<sudo apt-get install ca-certificates curl gnupg>**
- **<sudo install -m 0755 -d /etc/apt/keyrings>**
- **<curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg>**
- **<sudo chmod a+r /etc/apt/keyrings/docker.gpg>**
- **<echo \>**
- **<"deb [arch="\$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/debian \>**
- **<"\$(. /etc/os-release && echo "\$VERSION\_CODENAME)" stable" | \>**
- **<sudo tee /etc/apt/sources.list.d/docker.list > /dev/null>**

Una vez realizada la pre-configuración, procedemos con la instalación:

- **<sudo apt-get update>**
- **<sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin>**

Como probablemente tengamos problemas de permisos al ejecutar docker, deberemos añadirnos al grupo de docker:

- **<sudo groupadd docker>**
- **<sudo usermod -aG docker \$(whoami)>**

Una vez hecho esto, creamos un docker-compose.yml, ya que para facilidad de configuración y lanzamiento, ejecutaremos todo con docker compose. En este compose, crearemos, además, una base de datos de mysql dockerizada también (necesaria para el servidor apache de la tienda). Una vez hecho esto, seguiremos las instrucciones de prestashop para la instalación en un servidor.

## 21.ALGUNAS PROPUESTAS DE MEJORA DEL SERVIDOR

Debido a la falta de tiempo y para evitar posibles problemas de cara a la defensa del proyecto, se van a especificar aquí algunas propuestas para la mejora de este:

Dockerización de las bases de datos: Docker nos ofrece imágenes con bases de datos ya listas para su uso, lo que nos puede permitir gran comodidad a la hora de gestionar estas, y gracias a los volúmenes de Docker, podemos hacer que estas sean persistentes.

Dockerización del servidor Apache: Además de lo anteriormente mencionado, también podemos hacer uso de imágenes de apache (como en el caso de la tienda), para configurar más fácilmente el servidor (puertos, etc...) y lanzar todo de un solo comando con docker compose.

Integración continua: Utilizando herramientas de CI (Continuous Integration) como GitHub Actions, podemos actualizar con un solo commit toda nuestra infraestructura web, además, si combinamos esto con docker, podremos instalar de forma mucho más rápida y sencilla nuestra página en cualquier sistema, instalando solo en este un runner de GitHub Actions y el Docker Engine.

Otra posible mejora, sería, en el caso de que el cliente lo requiriese, que el servidor bloquee el acceso a diferentes webs, esto lo podríamos hacer gracias a la herramienta squid, que nos permite cachear datos solicitados por los usuarios y además bloquear las direcciones que queramos.

## 22. REFERENCIAS

- Apuntes proporcionados por la universidad  
<https://packages.ubuntu.com/search?keywords=net-tools>
- Bibliotecas de PERL:  
<https://www.cpan.org/>  
<https://www.geeksforgeeks.org/accton-command-in-linux-with-examples/>  
<https://www.linode.com/docs/guides/logwatch-monitor-system-logs/>  
[https://access.redhat.com/documentation/es-es/red\\_hat\\_enterprise\\_linux/8/html/security\\_hardening/configuring-auditd-for-a-secure-environment\\_auditing-the-system](https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/8/html/security_hardening/configuring-auditd-for-a-secure-environment_auditing-the-system)
- Definición para introducir Apache:  
<https://www.webempresa.com/hosting/que-es-servidor-apache.html>
- Aumentar el límite de tamaño de mensajes/archivos adjuntos:  
<https://rm-rf.es/postfix-aumentar-el-limite-de-tamano-de-mensajes-archivos-adjuntos/>
- Para qué sirve mysql\_secure\_installation:  
[https://phoenixnap.com/kb/mysql-secure-installation#:~:text=mysql secure installation%20is%20a%20shell%20script,accessible%20from%20outside%20the%20localhost.](https://phoenixnap.com/kb/mysql-secure-installation#:~:text=mysql%20secure%20installation%20is%20a%20shell%20script,accessible%20from%20outside%20the%20localhost.)
- Roundcube:  
<https://ticket.cdmon.com/es/support/solutions/articles/7000006301-c%C3%B3mo-acceder-y-gestionar-roundcube/#:~:text=Para%20acceder%20a%20Roundcube%20debes,introducen%20los%20datos%20que%20solicitan>
- Qué es y para qué sirve dropbox (meramente una pequeña introducción a qué es dropbox):  
<https://www.geeknetic.es/Dropbox/que-es-y-para-que-sirve>

- Para la definición del dominio:

<https://www.duckdns.org/domains>

- Instalación del Docker Engine

<https://docs.docker.com/engine/install/debian/>

- GitHub Actions:

<https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions>

- Que es Tripwire y como instalarlo:

<https://www.solvetic.com/tutoriales/article/3153-como-instalar-tripwire-sistema-de-teccion-intrusos-linux/>

- Cómo configurar Tripwire:

<https://www.etl.it.uc3m.es/Manual de configuraci%C3%B3n y uso de Tripwire>