

# CREACIÓN DE UN SERVIDOR EN DEBIAN 10



**Administración de Sistemas**  
**Universidad de Salamanca**  
ANDRÉS LARA ROMERO DE ÁVILA  
CARLOS GONZÁLEZ GÓMEZ

# Índice

<b>Introducción</b>	<b>3</b>
<b>Primeros pasos</b>	<b>4</b>
<b>Crear la máquina virtual en google cloud</b>	<b>5</b>
Conexión al servidor	6
<b>Pasos esenciales</b>	<b>9</b>
Configuración de apache2	10
Configuración de postfix	12
Configuración de perl	14
<b>Construcción servidor web</b>	<b>16</b>
Primera fase	16
Ventana principal	16
Ventana contacto	17
Ventana registro	17
Ventana aceptación de registro	18
Procesos adicionales	19
Segunda fase	21
Ventana blog	22
Configuración de servidor DNS	23
Ventana correo electrónico	24
Ventana archivos	28
Ventana moodle	30
Ventana de formularios	31
Configuraciones adicionales	33
Copia de seguridad	33
Tareas del sistema	34
<b>Bibliografía</b>	<b>36</b>

# Introducción

En el mundo actual la mayoría de las personas de la población mundial utiliza internet y sus ventajas, es por ello que cada vez más empresas se lanzan a la aventura de brindar una serie de servicios a los individuos que utilizan la red. Para brindar esos servicios es necesario un ordenador que funcione como servidor y que permita a todos sus usuarios utilizar los servicios que tiene disponibles. Hay muchos tipos de servidores, existen los servidores web, los servidores de bases de datos, los servidores de aplicaciones, etc. Pero lo que todos mantienen en común es que se dedican a brindar servicios a sus usuarios.

La facultad de ciencias de la universidad de Salamanca ha convocado una oferta pública para que cualquiera que esté interesado en construir y gestionar un servidor se ofrezca a ello. En este caso el servidor debe estar construido en un sistema operativo Debian 10, pues es el utilizado en clases de práctica en el aula de informática. Este sistema debe brindar unos servicios mínimos, los cuales iremos configurando a lo largo de este documento y explicando cada paso realizado. En este documento por tanto vamos a mostrar el proceso de creación del servidor y todo su posterior desarrollo para dejarlo en funcionamiento para el día 20 de Mayo de 2024.

La dirección a la página inicial de nuestro servidor es:  
<https://34.16.182.84/index.html> .

# Primeros pasos

Para comenzar nuestro proyecto necesitaremos planificar una serie de datos de partida para luego no tener fallos en pasos posteriores. Para poder montar un servidor será necesario determinar la cantidad de memoria que necesita la máquina para soportar el número de usuarios del sistema, es por ello, que antes de establecer la memoria del sistema hay que planificar el número de usuarios futuros y de cuanta memoria va a disponer cada usuario.

En nuestro sistema, cada usuario dispondrá de un tamaño de 80 Mb para almacenamiento de archivos. Si el sistema está pensado para utilizarlo en el grado en ingeniería informática supongamos que hay 100 alumnos por curso y hay cuatro cursos, esto hace un total de 400 alumnos, que multiplicado por los 80 Mb de memoria de la que dispone cada uno son 32000 Mb de memoria, o lo que es lo mismo 32 Gb de memoria. Además, hay que tener en cuenta todos los programas que va a almacenar el servidor y las bases de datos de los clientes. Para ir con memoria de sobra vamos a utilizar 40 Gb de memoria.

En nuestro caso vamos a utilizar la plataforma google cloud. Esta plataforma nos permite albergar una máquina virtual con las características deseadas en un servidor de google. La decisión de realizarlo de esta manera es que al ser un servidor con una ip pública seremos capaces de conectarnos desde cualquier parte, sin necesidad de cargar siempre con el equipo que contenga la máquina virtual. Además, esta solución nos permite trabajar de forma simultánea a los dos compañeros como si estuviéramos trabajando en terminales independientes. El único inconveniente que presenta google cloud es que es una plataforma de pago, pero con la cuenta de la usal disponemos de tres meses gratis, que es tiempo más que suficiente para realizar nuestro proyecto. La máquina virtual encendida durante todo un día consume alrededor de 1 euro, y nosotros disponemos de 400 euros por lo que no resulta un problema.

# Crear la máquina virtual en google cloud

La creación de la máquina virtual es muy sencilla. En primer lugar nos encontramos con una especie de formulario donde ir rellenando campos y escogiendo opciones. Este formulario es bastante extenso, pero iremos comentando los aspectos de mayor interés para nosotros.

Estado de prueba gratuita: crédito por €276.98 y 88 días restantes. Activa tu cuenta completa para obtener acceso ilimitado a todas las funciones de Google Cloud. Usa los créditos restantes [DESCARTAR](#) [ACTIVAR](#)

Google Cloud My First Project Buscar (/) recursos, documentos, productos y más [Buscar](#)

← Crear una instancia [CÓDIGO EQUIVALENTE](#)

**Nueva instancia de VM**  
Crea una instancia de VM única desde cero

**Nueva instancia de VM a partir de una plantilla**  
Crea una instancia de VM única a partir de una plantilla existente

**Instancia nueva de VM a partir de una imagen de máquina**  
Crea una instancia de VM única a partir de una imagen de máquina existente

**Marketplace**

Nombre \*  
instance-20240425-152519

**ADMINISTRAR ETIQUETAS DE INSTANCIA Y ETIQUETAS DE RECURSO**

Región \*  
us-west4 (Las Vegas)  
La región es permanente

Zona \*  
us-west4-b  
La zona es permanente

**Configuración de la máquina**

☒ De uso general ☐ Optimizado para procesamiento ☐ Con optimización de memoria

☐ Optimizada para almacenamiento ☒ NUEVO ☐ GPU

Tipos de máquinas para cargas de trabajo comunes, optimizados en función del costo y la flexibilidad

**Estimación mensual**  
USD28.65  
Equivale a alrededor de USD0.04 por hora  
Paga por lo que usas, con facturación por segundo y sin pagos por adelantado

Elemento	Estimación mensual
2 vCPU + 4 GB memory	USD27.55
Disco persistente balanceado de 10 GB	USD1.10
<b>Total</b>	<b>USD28.65</b>

[Precios de Compute Engine](#)

[CREAR](#) [CANCELAR](#) [CÓDIGO EQUIVALENTE](#)

Figura 1 Formulario para la creación de máquina virtual

En primer lugar tenemos que escoger una región donde albergar nuestra máquina. Está decisión depende en nuestro caso del precio mensual de la máquina, ya que hay lugares más baratos debido a la localización, pero si estuviéramos realizando un servidor real para nuestro uso, elegiríamos una región que se encuentre lo más cercana posible a nuestros usuarios, y que además se encuentre en un lugar lo más seguro posible (alejado de catástrofes naturales).

Lo siguiente es escoger un tipo de máquina que se ajuste a nuestras necesidades. A nosotros como la potencia de cálculo no nos interesa, escogemos la más sencilla (de nuevo fijándonos en el precio). Elegimos una máquina de la serie E2 (procesamiento diario de bajo costo), y dentro de ella una médium, que contiene 2 CPU virtuales, 1 núcleo y 4Gb de memoria RAM.

Ahora tendremos que ajustar la memoria necesaria. En nuestro caso, en lugar de usar 40Gb de memoria, vamos a utilizar 20Gb, por el simple hecho de que la opción de 40Gb de memoria excede mucho el presupuesto, y con 20Gb para probar el proyecto será suficiente (Figura 2).

## Disco de arranque

Nombre ↑	Imagen	Tipo de interfaz	Tamaño (GB)	Nombre del dispositivo	Tipo	Arquitectura	Encriptación
<a href="#">carldres</a>	<a href="#">debian-10-buster-v20240417</a>	SCSI	20	carldres	Disco SSD persistente	x86-64	Administrada por Google

Figura 2 Información sobre el disco de la máquina virtual

Por último existen una serie de configuraciones avanzadas y de red que no vamos a cambiar porque no son necesarias. Lo que sí hay que hacer, muy importante, es permitir el tráfico http y el https. Es importante mencionar que otra de las ventajas de la máquina virtual alojada en un servidor de google, es que nos permite realizar cambios sin perder el estado de la máquina, como por ejemplo de tamaño de memoria, que si tuviéramos máquina virtual local no sería posible y podría llevarnos a problemas.

Una vez tenemos todo el proceso hecho, aceptamos y ya tenemos la máquina virtual creada.

## Conexión al servidor

Para nosotros (los administradores del servidor) el acceso al servidor será por medio de una conexión ssh (como lo haríamos para encina). Al disponer el servidor de una ip pública, podemos conectarnos desde nuestros equipos ejecutando la orden: **ssh nombre\_de\_usuario@ip\_servidor**. Antes de conectarnos es necesario crear una clave pública desde la máquina local desde la que queramos acceder a nuestra máquina virtual, y luego guardar esa clave pública en la configuración de nuestra máquina virtual en google cloud (Figura 3).

Clave SSH 1 \*

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQCdWxJfPGmT+Un0ntihsoal

Ingresar la clave pública SSH

Clave SSH 2 \*

ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHA

Ingresar la clave pública SSH

Clave SSH 3 \*

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCC7gCx4qtf5Q3QU2zdsE.

Ingresar la clave pública SSH

+ AGREGAR ELEMENTO

**Figura 3** Añadir claves ssh a la máquina virtual

Una vez hecho esto podemos acceder desde nuestra terminal como se ve en la figura 4. Con el usuario para el que hemos creado la clave, en este caso id00805760, que además debe ser un usuario que exista en la máquina virtual, entramos utilizando el nombre de este usuario y la ip pública del servidor.

```

Microsoft Windows [Versión 10.0.22631.3447]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\andre>ssh id00805760@34.125.128.12
The authenticity of host '34.125.128.12 (34.125.128.12)' can't be established.
ED25519 key fingerprint is SHA256:w04q18vCxlnCA5gHglU4kk7LbionNwZ+10JyoqA/QKk.
This host key is known by the following other names/addresses:
  C:\Users\andre/.ssh/known_hosts:9: 34.125.104.120
  C:\Users\andre/.ssh/known_hosts:10: 34.16.164.5
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.125.128.12' (ED25519) to the list of known hosts.
Enter passphrase for key 'C:\Users\andre/.ssh/id_rsa':
Linux carldres 4.19.0-26-cloud-amd64 #1 SMP Debian 4.19.304-1 (2024-01-09) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Apr 25 14:31:29 2024 from 35.235.245.96
id00805760@carldres:~$

```

Figura 4 Conexión vía ssh a nuestro servidor remoto

Otra forma de conectarnos al servidor puede ser a través de la plataforma google cloud. Esta página tiene una opción para conexiones vía ssh a través de una nueva ventana del navegador. Esto lo podemos ver en la Figura 5 donde encontramos el botón para acceder a terminal ssh, y en la Figura 6 donde ya aparece la terminal conectada.

The screenshot shows the Google Cloud Platform console interface. At the top, there are tabs for 'INSTANCIAS', 'OBSERVABILIDAD', and 'PROGRAMAS DE LAS INSTANCIAS'. The 'INSTANCIAS' tab is selected. Below the tabs, there's a section titled 'Instancias de VM'. A table lists the instances. The first instance is 'carldres' with status 'On', zone 'us-west4-b', and external IP '34.125.128.12'. The 'Conectar' button for this instance is circled in orange. Below the table, there are several action cards for related tasks like 'Explorar copias de seguridad y DR', 'Consulta el informe de facturación', 'Supervisa VMs', 'Explora los registros de VM', 'Configura reglas de firewall', and 'Administración de parches'.

Estado	Nombre	Zona	Recomendaciones	En uso por	IP interna	IP externa	Conectar
On	carldres	us-west4-b			10.182.0.2 (nic0)	34.125.128.12 (nic0)	SSH

Figura 5 Botón para acceder al terminal

The screenshot shows a web browser window with the URL 'https://ssh.cloud.google.com/v2/ssh/projects/totemic-fact-421117/zones/us-west4-b/instances/carldres?authuser=1&hl=es\_419&projectNumber=...'. The browser title is 'SSH en el navegador'. Below the browser window, there's a terminal window showing the Debian GNU/Linux login screen for the 'carldres' instance. The terminal output is identical to the one in Figure 4.

```

Linux carldres 4.19.0-26-cloud-amd64 #1 SMP Debian 4.19.304-1 (2024-01-09) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Apr 26 08:30:48 2024 from 212.128.135.52
id00805760@carldres:~$

```

**Figura 6** Terminal que nos abre el navegador

Esta opción tiene otra ventaja y es que la máquina virtual puede estar vinculada a varias cuentas de google, por lo que podemos tener acceso los dos administradores sin molestarnos uno a otro, es decir, que ambos estaremos trabajando en la misma máquina, pero desde diferentes terminales, y aunque estemos utilizando el mismo usuario no tendremos interferencias.



# Pasos esenciales

Vamos a realizar una serie de configuraciones dentro del equipo que son necesarias para cualquier servidor de la familia Debian y que nos permitirán no caer en errores en un futuro. Lo primero es actualizar el sistema con la orden **sudo apt update** y **sudo apt upgrade**. Esto hay que realizarlo todo los días cuando entremos de nuevo a la máquina. Estas órdenes nos permiten actualizar nuestro sistema operativo a las últimas versiones disponibles, evitando fallos de seguridad y de otros tipos.

```
id00805760@carldres:~$ sudo apt update
Hit:1 https://deb.debian.org/debian buster InRelease
Hit:2 https://deb.debian.org/debian-security buster/updates InRelease
Hit:3 https://deb.debian.org/debian buster-updates InRelease
Hit:4 https://packages.cloud.google.com/apt google-compute-engine-buster-stable InRelease
Hit:5 https://packages.cloud.google.com/apt cloud-sdk-buster InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
id00805760@carldres:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
  google-cloud-cli-anthoscli
The following packages will be upgraded:
  google-cloud-cli google-cloud-packages-archive-keyring
2 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 106 MB of archives.
After this operation, 653 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 https://packages.cloud.google.com/apt cloud-sdk-buster/main amd64 google-cloud-cli amd64 473.0.0-0 [81.0 MB]
Get:2 https://packages.cloud.google.com/apt cloud-sdk-buster/main amd64 google-cloud-cli-anthoscli amd64 473.0.0-0 [24.7 MB]
Get:3 https://packages.cloud.google.com/apt google-compute-engine-buster-stable/main amd64 google-cloud-packages-archive-keyring all 1.2-627078515 [2886 B]
```

Figura 10 Proceso de actualización del sistema

Para trabajar en la máquina vamos a crear un usuario específico para los administradores del servidor. Este usuario se llamará igual que el nombre de la empresa, esto es, **carldres**. Para poder ejecutar las órdenes privilegiadas de superusuario con el usuario **carldres** que va a ser nuestro administrador del sistema, es necesario incluirlo en el grupo de root con la orden **sudo usermod -aG sudo carldres**. Además también utilizamos la orden **id carldres** para comprobar información sobre el identificador de usuario y los grupos a los que pertenece carldres.

```
id00805760@carldres:~$ sudo usermod -aG sudo carldres
id00805760@carldres:~$ id carldres
uid=1001(carldres) gid=1002(carldres) groups=1002(carldres),27(sudo)
```

Figura 11 Cambio grupo de carldres

Una vez tenemos nuestro usuario vamos a comenzar a instalar los servicios primordiales que vamos a necesitar en nuestro servidor. Los servicios primordiales son aquellos nos van a permitir desarrollar completamente el servidor. En Debian 10, y en todas las distribuciones de sistemas operativos de software libre, hay una comunidad muy grande que se dedica a mejorar el software existente cada día. Es por ello, que las distribuciones basadas en Debian, son unas de las más utilizadas para la creación de servidores, debido a su bajo coste, bajo consumo, eficiencia y sobre todo gracias a la gran cantidad de software gratis que existe para automatizar cualquier proceso.

En este momento vamos a descargar el software de necesidad casi inmediata para la construcción de las funciones del servidor, lo cual no excluye que con el desarrollo del proyecto, esta lista de software necesario se haga cada vez más amplia, pero no hay problema porque iremos indicando todo el software que requiera de instalación. Para empezar necesitamos instalar estos servicios:

- Apache2
- Postfix
- perl
- Nettools
- Php

Para instalar estos servicios simplemente usamos la orden **`sudo apt-get install [nombre del servicio]`**. Una vez instalados estos servicios vamos a configurar el servicio web apache2.

## Configuración de apache2

Para configurar apache es necesario que entremos en sus archivos de configuración, los cuales se encuentran en el directorio **`/etc/apache2/`**.

```
carldres@carldres:/home/id00805760$ ls -l /etc/apache2/
total 80
-rw-r--r-- 1 root root 7404 May 13 09:41 apache2.conf
drwxr-xr-x 2 root root 4096 May 12 12:07 conf-available
drwxr-xr-x 2 root root 4096 May 12 12:07 conf-enabled
-rw-r--r-- 1 root root 1782 Apr 21 2023 envvars
-rw-r--r-- 1 root root 31063 Jul 23 2022 magic
drwxr-xr-x 2 root root 12288 May 12 12:16 mods-available
drwxr-xr-x 2 root root 4096 May 13 09:43 mods-enabled
-rw-r--r-- 1 root root 320 Jul 23 2022 ports.conf
drwxr-xr-x 2 root root 4096 May 13 09:39 sites-available
drwxr-xr-x 2 root root 4096 May 12 12:44 sites-enabled
```

**Figura 12** Archivos y directorios de configuración de apache

El archivos que vamos a configurar es `apache2.conf`. Este archivo contiene todos los parámetros necesarios para el funcionamiento de apache. En nuestro caso lo más importante va a ser habilitar el directorio `/var/www/` que es donde se encuentra el directorio html, donde se encuentran los archivos html utilizados en el servidor web. También muy importante para la ejecución de los archivos perl, introducir un alias al directorio

/usr/lib/cgi-bin/ que sea /cgi-bin/ para que las redirecciones desde los archivos html sean comprendidas por el servidor.

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>

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

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

ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin">
    AddHandler cgi-script .cgi .pl
    Options FollowSymLinks ExecCGI
    AllowOverride none
</Directory>
```

**Figura 13** Archivo de configuración apache2.conf

Para el caso de nuestro servidor vamos a utilizar el protocolo https que utiliza el puerto 443. Tendremos que modificar el archivo apache.conf para incluir toda la información necesaria para el tráfico https. La siguiente Figura 14 muestra la información que contiene apache.conf sobre los directorios. Podemos ver como encontramos la redirección, por medio de la sentencia alias, a los diferentes programas de software libre utilizados (nextcloud, roundcube y wordpress). Es importante tener en cuenta que para utilizar el protocolo https es necesario tener un certificado expedido por una institución autorizada para validar los certificados SSL, y para ello es necesario tener un dominio comprado que sea visible públicamente. Los certificados se autorizan y validan para confirmar que son seguros, y es por ello que algunas veces cuando entramos en páginas web nos indica que no es segura, pero aún así nos permite acceder corriendo el riesgo de que sea una web peligrosa. Aunque nos indique que no es una web segura, no tiene porqué ser peligrosa, simplemente es que no tiene el certificado validado por la institución correspondiente. Hay otra forma de poder utilizar el tráfico https, y es emitiendo un certificado SSL local, esto es, que se configura y autentifica en nuestra propia máquina. Este método se recomienda para realizar pruebas en local, pero puede ser utilizado igualmente de forma pública. Nosotros al no tener un dominio comprado, vamos a utilizar este segundo método, por ello cada vez que entremos en las páginas del servidor nos aparecerá como web no segura, pero es simplemente por eso.

```

<VirtualHost *:443>
    ServerName 34.16.182.84
    DocumentRoot /var/www/html

    SSLEngine on
    SSLCertificateFile /var/www/html/ssl/server.crt
    SSLCertificateKeyFile /var/www/html/ssl/server.key

    <Directory /var/www/html>
        Options FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    Alias /correo /var/lib/roundcube
    <Directory /var/lib/roundcube>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    Alias /wordpress /var/www/wordpress
    <Directory /var/www/wordpress>
        Options FollowSymLinks
        AllowOverride Limit Options FileInfo
        DirectoryIndex index.php
        Require all granted
    </Directory>
    <Directory /var/www/wordpress/wp-content>
        Options FollowSymLinks
        Require all granted
    </Directory>
</VirtualHost>

```

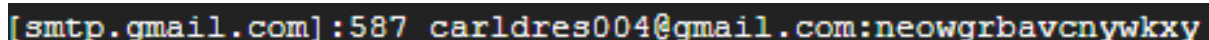
**Figura 14** Archivo de configuración apache2.conf para el tráfico https

## Configuración de postfix

Para configurar postfix necesitamos en primer lugar instalarlo (si no lo hemos hecho antes). En la instalación de postfix nos va a solicitar el propósito de su instalación, que en nuestro caso es el de Internet conexión, esto es, queremos que se envíen correos electrónicos a otras direcciones a parte de las locales. Además, nos solicitará un nombre de dominio que se utilizará para el envío de mensajes locales.

Una vez instalado postfix, lo siguiente será configurar una serie de ficheros. Primero debemos crear un fichero que se utilizará como conexión a nuestra cuenta de correo electrónico de gmail, para poder enviar correos electrónicos al exterior. Los correos que se envían al exterior se van a utilizar principalmente para mandar confirmaciones de registros y para mandar correos con información del servidor al administrador del sistema carldres004@gmail.com. Para que postfix pueda conectarse a nuestra cuenta de correo gmail, necesitamos autorizarlo por medio de un archivo de tipo sasl, en el que le indicaremos la cuenta de correo electrónico que queremos utilizar, la dirección del servidor

(en este caso del de gmail, smtp.gmail.com), el puerto de escucha del protocolo smtp, que es 587 y una contraseña única que creamos para este propósito. Esta contraseña una vez se ha utilizado, por mucho que intentemos volver a utilizar solo funcionará en el primer lugar que se utilizó, es decir, si intentamos iniciar sesión con esta contraseña en gmail no nos dejará hacerlo, pero en cambio, cuando envía correos el servidor sí que la puede utilizar, porque es el primer lugar donde se utilizó la contraseña. Con este mecanismo nos aseguramos de no utilizar nuestra contraseña real de correo electrónico, elevando la seguridad del servidor. Este archivo se encuentra en el directorio /etc/postfix/sasl, y su nombre es sasl\_passwd. El contenido de este archivo es el de la figura 15.



```
[smtp.gmail.com]:587 carldres004@gmail.com:neowgrbavcnywkxy
```

**Figura 15** Contenido del archivo sasl\_passwd

Lo siguiente es manipular a nuestro gusto el archivo de configuración de postfix. Este es el archivo main.cf y se encuentra en el directorio principal de postfix /etc/postfix/. La figura 16 muestra el contenido del archivo ya configurado. Las configuraciones más importantes realizadas en el archivo son las siguientes.

El uso del dominio local de correo electrónico carldres.local, con este sufijo se enviarán los correos electrónicos. Tuvimos un problema en la máquina virtual de google cloud, y es que es muy complejo el cambio de nombre de dominio puesto que es necesario cambiar el disco duro para cambiar el hostname. Como no queríamos perder el progreso (que ya era muy elevado en ese momento) decidimos que los usuarios tendrían su cuenta de correo electrónico con este formato usuario@carldres, pero los correos los enviaría con este otro formato [usuario@carldres.local](#). Esto se debe a que roundcube no permite el envío de mensajes con un único sufijo, es decir, necesita que haya al menos un punto en el sufijo y por lo tanto no era posible el uso de carldres como dominio de correo. A pesar de esto, los correos le llegan a los correos usuario@carldres, es decir, que hemos configurado los nombres de dominio de tal forma que [usuario@carldres.local](#) es lo mismo a poner usuario@carldres, lo que pasa es que para enviar correos, es necesario que el destinatario tenga el formato usuario@carldres.local. Con este mecanismo hemos solucionado el problema del envío de correos en roundcube.

Lo siguiente que configuramos en el archivo main.cf es myorigin en el que le indicamos localhost, entonces cuando mande un correo, va a utilizar el nombre localhost como sufijo, esto es, usuario@carldres. Vamos a indicarle las redes locales donde puede trabajar, y en este caso incluimos la red local que usa la máquina virtual de google cloud, que es 10.182.0.0. También le vamos a limitar el buzón de correo de cada usuario a 3 Mb de memoria como dice el enunciado.

La última parte corresponde a la configuración de los protocolos sasl para poder enviar correos a través del servidor smtp de google (gmail) con nuestra cuenta de gmail. En general lo que estamos diciendo es que utilice un archivo para poder acceder y que utilice la contraseña del archivo sasl.

```

# 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 = carldres.local
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = localhost
mydestination = $myhostname, carldres, carldres.c.totemic-fact-421117.internal, localhost.c.
relayhost = [smtp.gmail.com]:587
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 10.182.0.0/20
mailbox_size_limit = 3145728
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all

home_mailbox = Maildir/
mailbox_command =

#Enable SASL authentication
smtp_sasl_auth_enable = yes
smtp_sasl_security_options = noanonymous
smtp_sasl_password_maps = hash:/etc/postfix/sasl/sasl_passwd
smtp_tls_security_level = encrypt
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt

```

**Figura 16** Configuración del archivo main.cf de postfix

## Configuración de perl

Para que el usuario que utiliza el servidor web apache en nuestro sistema pueda ejecutar los archivos perl es necesario configurar una serie de parámetros en nuestro servidor. En primer lugar necesitamos instalar el intérprete de perl (si aún no lo hemos hecho). Es importante mencionar que en nuestro caso vamos a almacenar los archivos de perl en el directorio `/usr/lib/cgi-bin/` que es el directorio habitual utilizado para almacenar este tipo de archivos. En nuestro caso todos los archivos perl se almacenarán en el directorio por defecto para ello, esto es `/usr/lib/cgi-bin/` evitando así problemas de permisos. Para que se puedan redireccionar las peticiones realizadas en el buscador web a los archivos perl, es necesario añadir una serie de líneas en el archivo de configuración de apache. Estas líneas son las de la Figura 18. En ellas le indicamos a apache, que la dirección `/cgi-bin/` la traduzca en `/usr/lib/cgi-bin/`.

```
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/  
<Directory "/usr/lib/cgi-bin">  
    AddHandler cgi-script .cgi .pl  
    Options FollowSymLinks ExecCGI  
    AllowOverride none  
</Directory>
```

**Figura 17** Líneas de configuración de los archivos perl

Además, será necesario instalar manualmente cada uno de los módulos de perl utilizados en los script. Para instalar estos módulos utilizaremos la orden ***sudo cpanm [módulo a instalar]***. En nuestro caso utilizamos la librería cpan minux y la cpan normal para instalar los módulos. Además, también es importante que los archivos perl tengan permisos de ejecución al menos para el usuario www-data que es el que utiliza apache en nuestro sistema.

# Construcción servidor web

La construcción del servidor web la vamos a dividir en dos partes. La primera parte será la encargada de construir todo lo que tiene que ver con la fase anterior al registro de usuario, incluyendo el registro y validación del usuario. La segunda parte consistirá en crear la interfaz que va a utilizar el usuario una vez esté logueado en el sistema.

Para construir la primera fase, hemos decidido hacer un menú sencillo donde aparezca una página principal, que muestre información sobre la empresa y los servicios instalados en el sistema. Recordemos que todas las páginas escritas en html se guardan en el directorio `var/www/data/`, y todos los scripts escritos en html se almacenan en el directorio `/usr/lib/cgi-bin/`.

## Primera fase

### Ventana principal

Esta primera pestaña estará representada por el archivo `index.html`, y será lo primero que vea el usuario cuando acceda al portal web. La pestaña de índice estará formada por una breve descripción de los servicios ofrecidos por nuestra empresa `carldres`, una imagen de un pingüino que representa el símbolo de linux, para darle originalidad y por último mostrará los servicios activos e inactivos del sistema en el momento en el que se imprime la página.

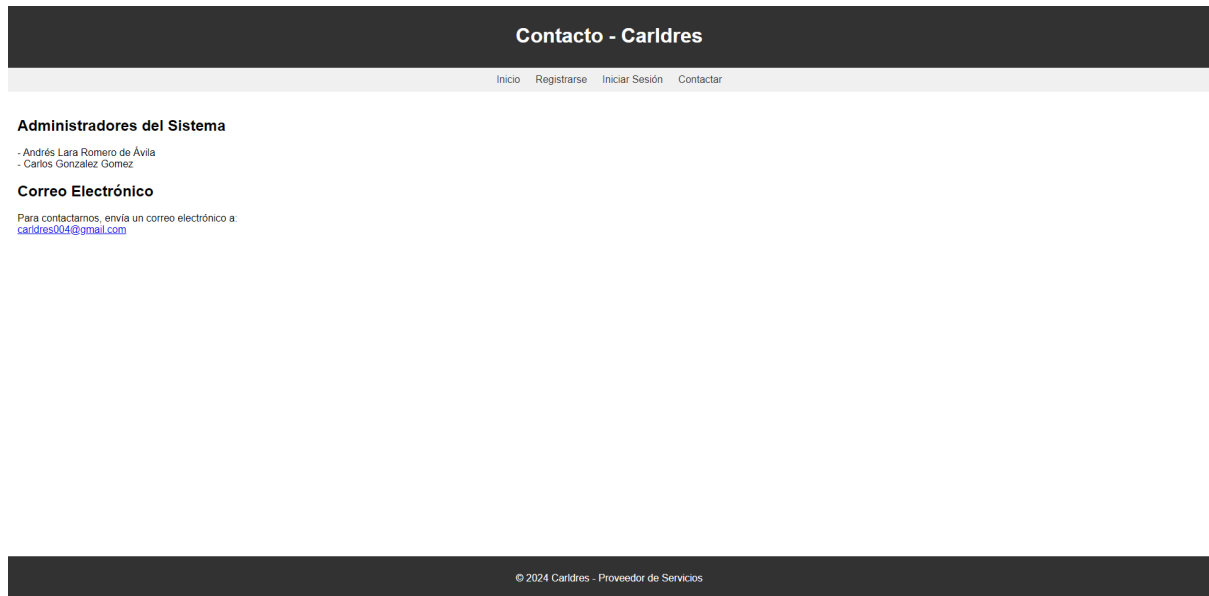


**Figura 18** Vista del menú principal



## Ventana contacto

Para seguir con las ventanas vamos a mostrar la de contacto, que simplemente contiene el nombre de los integrantes de la empresa y el correo electrónico de contacto.



**Figura 19** Vista de la ventana de contacto

## Ventana registro

La siguiente ventana es la de registro. Esta ventana llamada registro.html, es especial debido a que es un formulario para ingresar como usuario en el sistema. En el formulario se solicita el nombre de usuario, el nombre, el apellido, la contraseña y el email. Una vez introducidos los datos, cuando se pulsa el botón de registro, se envía un correo electrónico a la dirección proporcionada en el formulario. Esto se realiza por medio de un script en perl llamado procesar\_registro.pl. En este script de perl, se comprueba si el usuario existe en la base de datos de usuarios, y de ser así, si es un usuario confirmado (autenticado). Tanto si existe sin autenticar, como si no existe, se manda el correo electrónico con un token único y un enlace para introducir ese token, como se ve en la figura 22.

**Figura 20** Ventana de registro (formulario)

La pestaña de la figura 21 aparece si el registro ha sido exitoso.

### Actualizacion Exitosa

La informacion del usuario ha sido actualizada correctamente. Se ha enviado un correo electronico de confirmacion a tu direccion de correo electronico.

**Figura 21** Ventana que nos indica que el envío de correo electrónico ha sido exitoso

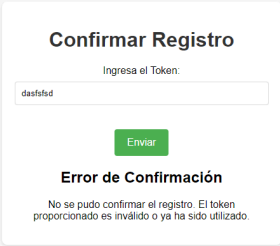


**Figura 22** Correo electrónico recibido tras el registro

## Ventana aceptación de registro

La siguiente ventana será la encargada de recoger el token enviado por correo electrónico. Esta ventana será la ventana confirmar\_registro.html. Además, estará respaldada por otro script escrito en perl llamado confirmar\_registro.pl, que será el encargado de confirmar que ese token es correcto para el usuario, y darlo de alta (marcando su casilla en la base de datos con 1). En este momento, el usuario sería capaz de entrar al sistema, aunque no disponga de un usuario en el servidor, pero de lo que sí dispone es de una fila en la base de datos con su información. Será un script, llamado crea\_usuarios.pl, el cual será el

encargado de crear todo lo que tiene que ver con el usuario, esto es, su usuario en el sistema, su carpeta de directorios, y su correspondiente usuario en el correo electrónico y en los archivos del sistema. Este script de crea\_usuarios se ejecutará una vez cada hora, para lograr que los usuarios esperen lo menos posible para acceder a los servicios esenciales. Hemos configurado el login de usuario de esta manera, para evitar que el usuario www-data cree los usuarios del sistema y ejecute instrucciones que son privilegiadas. Al ser un servicio, el que ejecuta el script, no habrá problemas de seguridad, porque los servicios los ejecuta el system y solo tiene acceso él.



**Figura 23** Ventana para confirmar el registro de usuario

## Procesos adicionales

Aparte de todos estos archivos para que funcione la aplicación web, vamos a programar una serie de tareas que nos faciliten la limpieza del servidor. Una de estas tareas es la de eliminar usuarios que no están confirmados todas las noches. Nuestro sistema está construido para que los usuarios que se quieren registrar rellenen el formulario. Si hay alguien que quiere entrar en nuestro sistema utilizando un correo falso, que no es suyo, no podrá autenticarse, porque no le llegará el correo con el token correspondiente. Todos estos intentos de acceso que no se confirman pueden llegar a saturar nuestra base de datos, lo cual no es algo conveniente teniendo en cuenta que es una fuente muy importante de información. Por ello vamos a programar una tarea que se ejecute todas las noches para eliminar los usuarios no confirmados de nuestro sistema.

Para conseguir nuestro objetivo, es necesario crear un archivo perl para que sea ejecutado por un servicio con temporizador. El archivo perl se va a llamar eliminar\_usuarios\_sin\_confirmacion.pl y se encontrará en el directorio donde guardamos todos los archivos perl esto es /usr/lib/cgi-bin/.

```
carldres@carldres:/var/www/html$ sudo systemctl list-timers -all
```

NEXT	LEFT	LAST	PASSED	UNIT	ACTIVATES
Tue 2024-05-14 14:32:15 UTC	4min 31s left	Tue 2024-05-14 14:22:15 UTC	5min ago	gce-workload-cert-refresh.timer	gce-workload-cert-refresh.service
Tue 2024-05-14 14:39:00 UTC	11min left	Tue 2024-05-14 14:09:05 UTC	18min ago	phpsessionclean.timer	phpsessionclean.service
Tue 2024-05-14 17:17:05 UTC	2h 49min left	Tue 2024-05-14 11:17:05 UTC	3h 10min ago	google-oslogin-cache.timer	google-oslogin-cache.service
Tue 2024-05-14 17:31:05 UTC	3h 3min left	Mon 2024-05-13 17:31:05 UTC	20h ago	systemd-tmpfiles-clean.timer	systemd-tmpfiles-clean.service
Wed 2024-05-15 00:00:00 UTC	9h left	n/a	n/a	eliminar_usuarios_sin_confirmacion.timer	eliminar_usuarios_sin_confirmacion.service
Wed 2024-05-15 00:00:00 UTC	9h left	Tue 2024-05-14 00:01:05 UTC	14h ago	logrotate.timer	logrotate.service
Wed 2024-05-15 00:00:00 UTC	9h left	Tue 2024-05-14 00:01:05 UTC	14h ago	man-db.timer	man-db.service
Wed 2024-05-15 00:18:59 UTC	9h left	Tue 2024-05-14 06:58:42 UTC	7h ago	apt-daily.timer	apt-daily.service
Wed 2024-05-15 06:24:58 UTC	15h left	Tue 2024-05-14 06:04:47 UTC	8h ago	apt-daily-upgrade.timer	apt-daily-upgrade.service

**Figura 24** Lista de los temporizadores del sistema

Otro de los problemas que encontramos en el sistema, es que, no creemos conveniente que cualquier usuario se puede registrar con el rol que tenga, puesto que puede haber alumnos, que para tener privilegios de profesor se registren con el rol de profesor. Es por ello, que la solución que proponemos a este problema es que los usuarios por defecto queden registrados como alumnos y que por medio del envío de un formulario dentro del login de los clientes, podamos en el correo de los administrados recibir las solicitudes de cambio de rol y tratarlas. Creemos que esta forma es más segura, porque serán los administradores del sistema los que comprueben si el usuario que solicita el cambio de rol es realmente un profesor. Para solucionar de la forma más automática posible esta tarea, vamos a crear un script que por medio del nombre del usuario, se cambie el rol del mismo en la base de datos, aunque aún deberemos cambiar el rol tanto en moodle como en nextcloud. El archivo perl se llamará `cambiar_rol_usuario.pl` y por supuesto es importante que solo lo pueda ejecutar root, puesto que el único usuario que tiene el valor de root es el administrador (carldres).

```
carldres@carldres:/var/www/html$ sudo perl /usr/lib/cgi-bin/cambiar_rol_usuario.pl juanito
No se encontró el usuario 'juanito' en la base de datos.
carldres@carldres:/var/www/html$
```

**Figura 25** Prueba del script para cambiar rol

Otro de los problemas a los que nos enfrentamos es el de utilizar https en lugar de http. Nuestro principal problema aquí, se encuentra en que al no tener un dominio público y utilizar https por medio de un certificado autofirmado (que lo hemos generado nosotros mismos), es decir, que no lo ha autorizado ninguna institución designada para ello, en el navegador nuestras ventanas aparecerán con advertencias de seguridad, pero al menos sabremos que es posible utilizar nuestra página con https. Para corregir esto vamos a generar un certificado ssl con el software openssl, que aunque no es un certificado real, pero para hacer nuestras pruebas es suficiente.

```
carldres@carldres:/var/www/html/ssl$ sudo openssl req -x509 -nodes -newkey rsa:2048 -keyout server.key -out server.crt -days 365 -subj "/CN=tu.ip publica"
Generating a RSA private key
.....+++++
writing new private key to 'server.key'
-----
carldres@carldres:/var/www/html/ssl$ ls
server.crt server.key
```

**Figura 26** Creando la clave ssl mediante openssl

## Segunda fase

En esta segunda fase vamos a tratar todas las partes que tienen que aparecer una vez el usuario ha hecho login. En primer lugar, vamos a mostrar una página inicial donde el usuario disponga de una breve descripción que le advierta sobre el uso responsable del servidor y que tenga un menú donde aparecen las principales opciones disponibles. Esta ventana principal quedará de esta manera.



**Figura 27** Página inicial del login de usuario

Por supuesto si no has iniciado sesión en el sistema no se puede pasar a esta página, aunque se busque directamente en el buscador web. Para conseguir esto hemos utilizado una sentencia condicional que comprueba si el nombre de usuario se encuentra en la matriz de sesiones de php. Si el usuario que ha iniciado la sesión es el mismo que está navegando por la web, nos permite el acceso a la página de usuario. Esto lo realizaremos para todas las páginas dentro del menú de usuario.

```
// Verificar si el usuario ha iniciado sesión
if (!isset($_SESSION["username"])) {
    // No hay sesión activa, redirigir a la página de inicio de sesión
    header("Location: login.php");
    exit();
}
```

**Figura 28** Código para comprobar que hay una sesión iniciada

## Ventana blog

Cuando pulsamos en el enlace blog, la web nos redirecciona a la página principal de wordpress donde encontraremos otra zona de login. Si hemos conseguido autenticarnos, una vez dentro podremos realizar las gestiones que queramos según los permisos de los que disponga el usuario con el que iniciamos sesión. Los usuarios del blog, se crean de forma automática, a través de la API de wordpress, cuando solicitamos el alta de usuario. La web comprueba si el usuario del que estamos solicitando el alta en el blog, es el mismo que el que ha iniciado la sesión, así nos aseguramos que no se puede suplantar la identidad. Por defecto, los usuarios en wordpress se crean con el rol de autor, esto es, que pueden crear sus propias entradas en el blog y pueden visualizar las de los demás, pero no pueden realizar cambios en publicaciones ajenas. Por supuesto en Wordpress disponemos de nuestra cuenta de administradores con la que disponemos de todos los privilegios a nuestro alcance. En la figura 28 podemos ver la página principal del blog para el caso del usuario administrador.

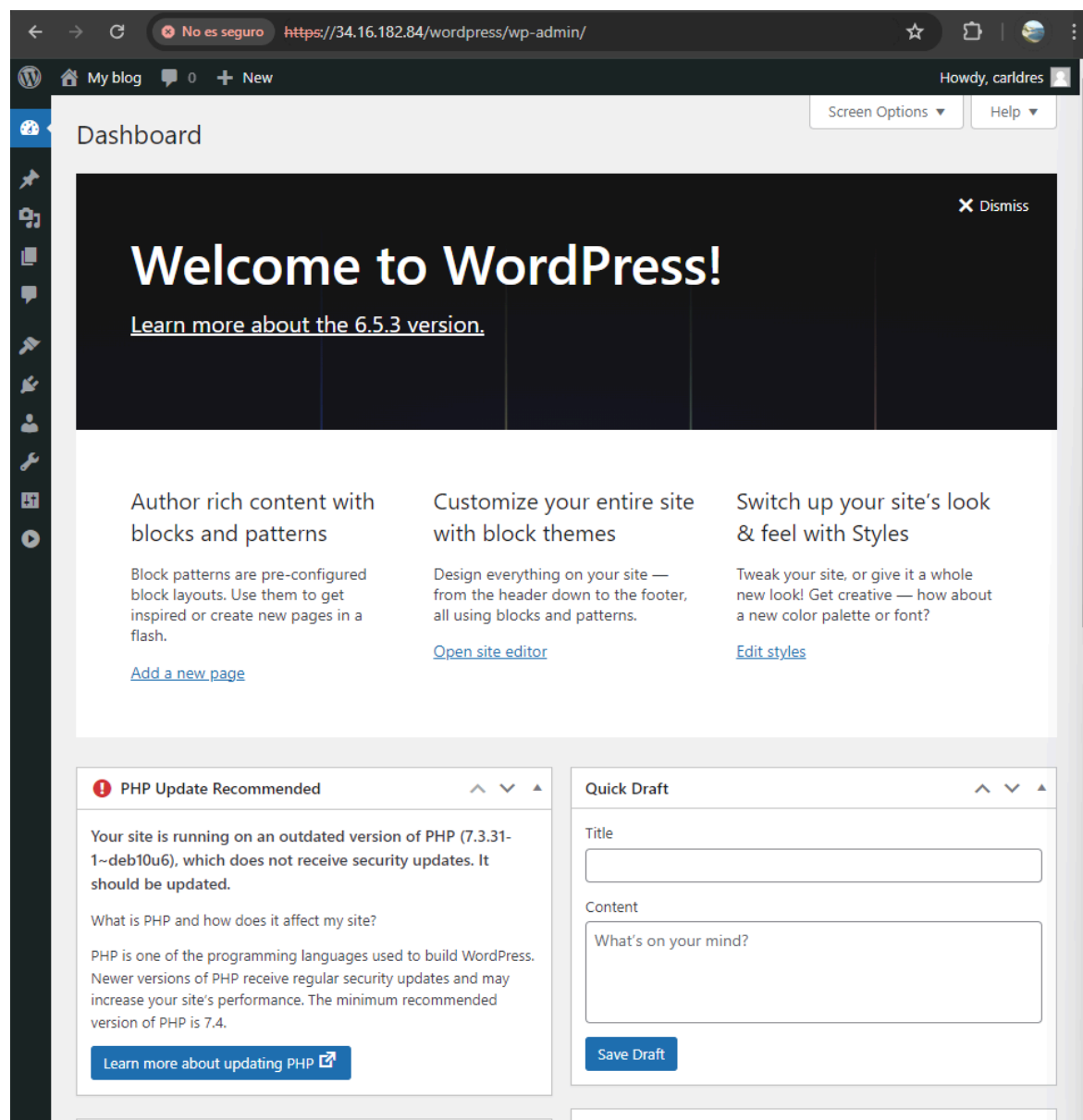


Figura 29 Página principal del blog

Para dar de baja la cuenta del blog, el usuario tiene que rellenar el formulario correspondiente en la ventana de formularios.

## Configuración de servidor DNS

Para poder establecer el servidor de correo electrónico tendremos que configurar nuestro servidor como un servidor dns. Para poder configurar el dns en una máquina cloud de google, es necesario hacerlo a través de una api dns.

Para crear un servidor DNS es necesario instalar el software bind9 y modificar una serie de archivos de configuración. Lo más destacable de este proceso es la creación de nuestra zona para su posterior conexión con el correo electrónico. Esta zona se llamará cardres.local y habrá que configurar tanto el archivo de resolución directa, que será el /etc/bind/zonas/db.cardres.local (Figura 30), como el archivo de resolución inversa, que será el /etc/bind/zonas/db.0.182.10 (Figura 31).

```
carldres@carldres:~$ sudo cat /etc/bind/zonas/db.carldres
[sudo] password for carldres:
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      carldres.local. root.carldres.local. (
                        2      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
carldres.local IN NS      servidor.carldres.local.
carldres.local IN MX      10 correo.carldres.local.
$ORIGIN carldres.local.
$TTL 300 ;5 minutes
servidor IN A 10.182.0.3
pop3 IN CNAME servidor
smtp IN CNAME servidor
correo IN CNAME servidor
```

**Figura 30** Contenido del archivo de configuración db.carldres.local

En este archivo básicamente le estamos indicando que cuando soliciten el nombre carldres.local que lo traduzca en la ip 10.182.0.3, que es la ip de la red local que corresponde a nuestro servidor. Al trabajar con una máquina de google cloud la configuración de nuestro servidor como dns se nos complicó bastante, aunque finalmente pudimos solucionarlo.

```

GNU nano 3.2                                     zonas/db.0.182.10
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      carldres. root.carldres. (
                        2      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
3         IN      NS       servidor.carldres.
3         IN      PTR      servidor.carldres

```

**Figura 31** Contenido del archivo de configuración db.0.182.10

Esta otra figura 31 corresponde con el archivo de resolución inversa de nuestra ip local.

## Ventana correo electrónico

Una vez hemos configurado nuestro servidor como servidor dns, en la ventana de correo electrónico, vamos a redireccionar al usuario a la interfaz de roundcube, lo cual nos facilitará en gran medida el envío de correos electrónicos y sobre todo nos ahorrará mucho tiempo en lo que es la interfaz. Vamos a utilizar roundcube que, por supuesto, es de código abierto.

Vamos a necesitar postfix (que ya está instalado) como servidor de correo electrónico, Dovecot para el acceso a IMAP y POP3 y roundcube que es un software que nos va a servir de interfaz web. Nos aseguraremos de seguir las instrucciones de seguridad para cada componente y garantizar que nuestro sistema de correo funcione correctamente.

Es importante asegurarnos de que el nombre del equipo sea el mismo que nombre de dominio que resuelve el servidor dns, o en su defecto como es nuestro caso en el que no podemos cambiar el hostname de la máquina, hay que incluir el nombre de dominio del correo electrónico en el archivo /etc/hosts con la ip local de nuestra máquina.

```

carldres@carldres:/etc/bind$ hostname
carldres

```

**Figura 32** Comprobando el nombre del equipo

Vamos a probar también a realizar ping con diferentes nombres para ver si el equipo sabe la dirección del nombre de dominio. En la Figura 35 podemos ver el resultado.



```

carldres@carldres:/etc/bind$ ping carldres
PING carldres.c.totemic-fact-421117.internal (10.182.0.3) 56(84) bytes of data:
64 bytes from carldres.c.totemic-fact-421117.internal (10.182.0.3): icmp_seq=
1 ttl=64 time=0.028 ms
64 bytes from carldres.c.totemic-fact-421117.internal (10.182.0.3): icmp_seq=
2 ttl=64 time=0.046 ms
^C
--- carldres.c.totemic-fact-421117.internal ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 30ms
rtt min/avg/max/mdev = 0.028/0.037/0.046/0.009 ms
carldres@carldres:/etc/bind$

```

**Figura 33** Probando la resolución del nombre de dominio

Una vez realizados las pruebas con postfix vamos a dovecotpop3, que es un servidor de código abierto y que se utiliza para recuperar correos electrónicos de un buzón de correos, y descargarlo en el cliente de correo del usuario. En este caso nos será muy útil si queremos gestionar todos los correos desde un único dispositivo, como en este caso es el servidor.

Primero tenemos que configurar el servidor pop para nuestra red local. En el archivo de configuración main.cf de postfix, vamos a añadir la red local en la que se encuentra el servidor, y una líneas (las últimas de la Figura 36) para que cree un directorio para cada usuario en su directorio principal, y que este directorio (Maildir/) albergue todos los correos electrónicos como ficheros independientes.

```

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_$
myhostname = carldres
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = $myhostname, carldres, carldres.c.totemic-fact-421117.interna$
relayhost = [smtp.gmail.com]:587
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 10.182.0.0/20
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all

home_mailbox = Maildir/
mailbox_command =

```

**Figura 34** Archivo de configuración main.cf de postfix

Importante que no deshabilitamos las autenticación por texto plano para los usuarios, porque esto nos permitirá que puedan iniciar sesión desde el servidor web. Para ello vamos al archivo 10-auth.conf del directorio /etc/dovecot/conf.d.

```
GNU nano 3.2 /etc/dovecot/conf.d/10-auth.conf Modified
##
## Authentication processes
##

# Disable LOGIN command and all other plaintext authentications unless
# SSL/TLS is used (LOGINDISABLED capability). Note that if the remote IP
# matches the local IP (ie. you're connecting from the same computer), the
# connection is considered secure and plaintext authentication is allowed.
# See also ssl=required setting.
disable_plaintext_auth = no
```

**Figura 35** Deshabilitamos la prohibición de autenticación por medio de texto plano

Lo siguiente es instalar la herramienta IMAP. Es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor. Nuestra idea es que con los equipos cliente se puedan acceder a estos correos electrónicos almacenados en el servidor. Para instalarlo vamos a utilizar `sudo apt install dovecot-imapd`. Es importante saber que imap utiliza un gestor de base de datos, que en nuestro caso va a ser mysql que ya teníamos instalado, pero si no estaba instalado, será necesario instalarlo.

Ahora vamos a instalar roundcube que es una plataforma de cliente de correo electrónico que nos proporciona una interfaz web que permite a los usuarios gestionar los correos electrónicos, y realizar acciones como leer, enviar o recibir correos. Lo primero es instalar roundcube con la orden de siempre. Luego vamos a crear un fichero en el directorio de apache `/etc/apache2/sites-available` que se llamará `round.conf`. En este fichero tenemos que cambiar las líneas que aparecen en la figura 38. Estas líneas permiten redirigir las peticiones de correo electrónico, a ese directorio (`/var/lib/roundcube`) que es donde se encuentra instalado roundcube.

```
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port$
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) t$
    # value is not decisive as it is used as a last resort host regardles$
    # However, you must set it for any further virtual host explicitly.
    ServerName correo.carldres

    ServerAdmin webmaster@localhost
    DocumentRoot /var/lib/roundcube
```

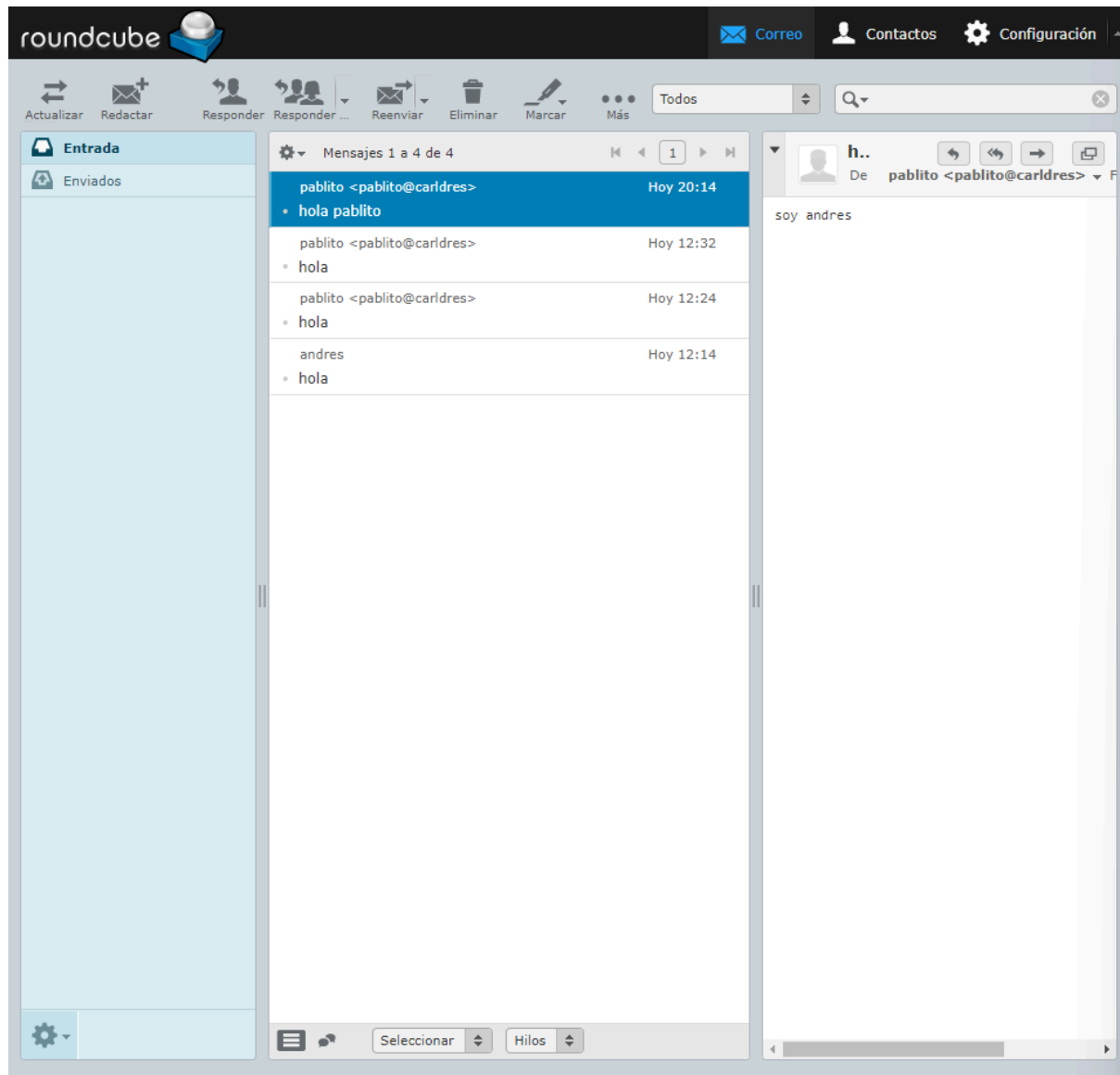
**Figura 36** Líneas a modificar en `round.conf`

Lo siguiente es activar el sitio web con el comando `a2ensite`.

```
carldres@carldres:/etc/apache2/sites-enabled$ ls
000-default.conf  mi-sitio.conf
carldres@carldres:/etc/apache2/sites-enabled$ sudo a2ensite round.conf
```

**Figura 37** Activamos el sitio web de roundcube

Ahora vamos a mostrar la interfaz de roundcube. Todo lo que tiene que ver con el envío de correos, ya se explicó en la parte donde instalamos postfix, por lo tanto de lo único que advertiremos aquí, es que para enviar un correo electrónico, la sintaxis del destinatario es “[usuario@carldres.local](#)”, y esto repetimos que se produce porque no es posible enviar correos con un solo nombre en el dominio, es decir, no se puede enviar “usuario@carldres”.



**Figura 38** Ventana principal roundcube

Las ventajas que encontramos utilizando roundcube, es que el envío de mensajes se realiza de forma segura y controlada. Además los usuarios dispondrán de una interfaz parecida a las que se utilizan en las plataformas más utilizadas de hoy en día. Gracias al archivo de configuración de postfix, somos capaces de limitar el buzón del usuario a 3MB. Es posible realizar otras muchas configuraciones, pero que ahora por simplicidad no vamos a utilizar.

Es importante mencionar, que los usuarios de nuestro sistema, son capaces de enviar correos a gmail. El problema de estos correos, es que el origen siempre corresponde con la

cuenta de gmail con la que tenemos sincronizada el servidor. Esto se debe, a que para poder comunicar con el correo exterior, es decir, con otros servidores de correos públicos, es necesaria una cuenta con un dominio público, y al no disponer de un dominio público en nuestro servidor, lo que hace es utilizar la cuenta del administrador, aunque le incluye delante el nombre de usuario.

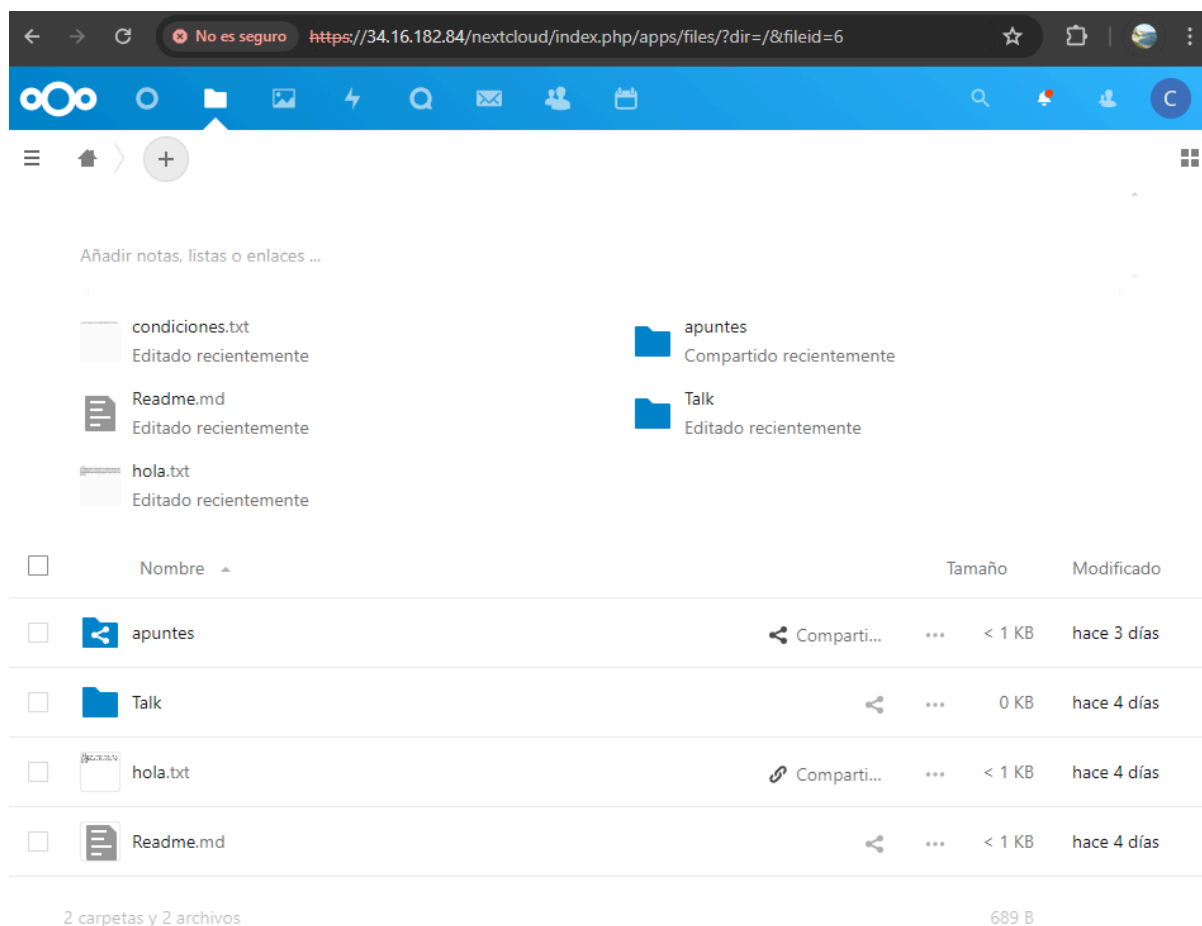
## Ventana archivos

La ventana de archivos comenzó siendo dos ventanas diferentes donde encontrábamos los archivos del directorio principal del usuario y la carpeta compartida de todos los usuarios (apuntes). El problema de esta manera era que la transferencia de archivos por mucho que usáramos el protocolo https no era muy segura, puesto que los archivos viajaban por texto plano a través de peticiones web. Esto nos llevó a la conclusión de que teníamos que idear otro sistema para poder realizar todo esto de forma segura, y que además nos facilitara el trabajo lo máximo posible. La conclusión final fue la de utilizar nextcloud.

Nextcloud, es un software de código abierto que se utiliza entre otras cosas para la gestión de archivos. Además, nextcloud permite otras muchas funcionalidades como son las videollamadas entre usuarios, el disponer de un calendario donde ir planificando los días y sobre todo una función muy importante para nosotros, que es la de crear grupos de usuarios que dispongan de permisos para manejar carpetas y archivos. Aquí es donde entra la parte más ventajosa de nextcloud para nosotros, y es que para controlar la carpeta apuntes, que era aquella donde solo tiene permisos de modificación los profesores y los alumnos solo pueden leer o descargar el contenido, podemos construir dos grupos (alumnos y profesores) y gestionar los permisos de cada grupo para acceder a esa carpeta de forma muy sencilla. Otra de las ventajas que nos ofrece nextcloud, es que nos permite ajustar las cuotas de usuario de forma automática en su creación.

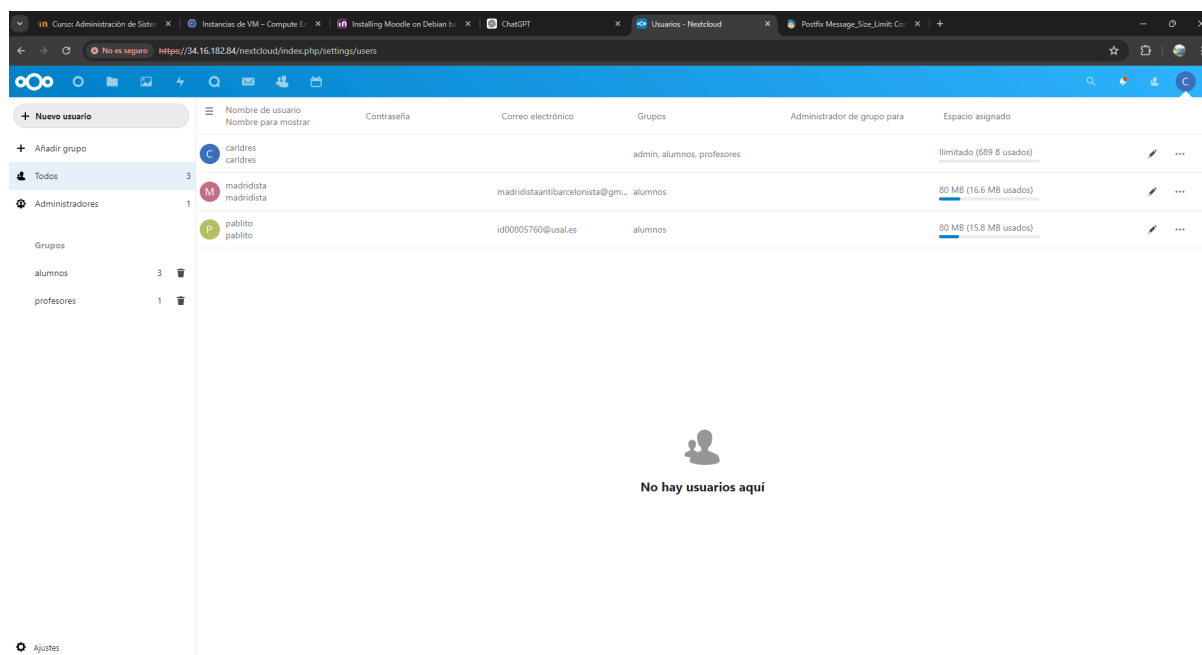
En la siguiente figura podemos ver la pestaña inicial de nextcloud (Figura 39). Cuando un usuario se crea con la tarea que ejecuta el servidor automáticamente cada hora, se crea su correspondiente usuario en nextcloud con su nombre de usuario, contraseña, rol de alumno (que posteriormente se puede cambiar mediante el formulario) y una cuota de 80Mb. Esto lo hemos conseguido a través de la API de nextcloud utilizando peticiones web al software de nextcloud. Esto mismo es lo que se realiza cuando se solicita la creación de un usuario en el blog de wordpress.

Aunque podríamos haber utilizado el correo de nextcloud, hemos decidido separar los servicios ofrecidos por el servidor en diferentes aplicaciones software para tolerar fallos, pero nextcloud nos parece una herramienta increíble para integrar todo lo que tiene que ver con la planificación y control del trabajo (archivos, calendario, contactos, videollamadas, etc).



**Figura 39** Ventana principal nextcloud

En la figura 40 podemos ver la página para gestionar usuarios que utiliza el administrador.

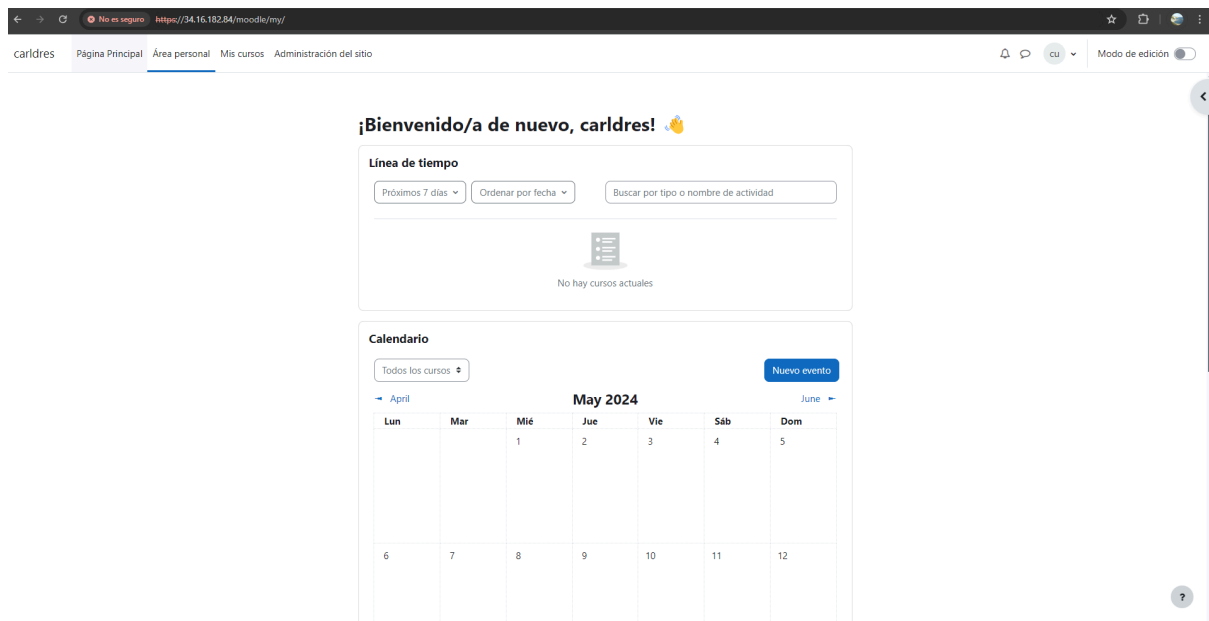


**Figura 40** Ventana de gestión de usuarios del administrador

## Ventana moodle

Para moodle hemos tenido que utilizar una versión bastante antigua (la 4.0) teniendo en cuenta que actualmente están utilizando la 4.4. Esto se debe a que Debian 10 está hecho para funcionar con la versión 7.3 de PHP y las actuales versiones de moodle solo funcionan con PHP 8.0 y superiores, además de incompatibilidad con versiones de mariaDb y otros muchos problemas.

En cuanto a moodle, no hemos probado mucho el software, pero lo que sí podemos comunicar, es que para darse de alta en moodle, tenemos un apartado en el formulario. Cuando se rellena el formulario, la petición se escribe en un archivo encolado, y todas las mañanas se envía este archivo al administrador del servidor para que realice las altas correspondientes. Podríamos haber utilizado la API de moodle pero el problema de las APIs es que cada una tiene sus formatos, y lleva mucho tiempo entender cada una de ellas, puesto que hay una distinta para cada aplicación (nextcloud, wordpress, etc). Es por ello, que por falta de tiempo no hemos decidido crear usuarios de forma automática al igual que se hace con wordpress y nextcloud, pero al haberlo conseguido con las otras aplicaciones somos conscientes de que es posible conseguirlo con esta también.



**Figura 41** Ventana principal de moodle

## Ventana de formularios

La ventana de formularios se divide en 6 formularios principales. El primero consiste en el formulario de alta en wordpress (Figura 42). En él encontramos una serie de parámetros a introducir. Este formulario ejecuta un script que automáticamente por medio de la api de wordpress, crea un usuario con los datos introducidos. Es importante tener en cuenta que en todos los formularios que se utiliza un nombre de usuario, el sistema comprueba que el usuario que se introduce en el campo, es el mismo que el que ha iniciado la sesión. Con esto conseguimos bloquear la posibilidad de que un usuario cualquiera pudiera crear cuentas en los servicios de la aplicación suplantando la identidad de otras personas, sobre todo en los formularios que producen una respuesta inmediata sin pasar por los administradores.

The screenshot shows the "Formularios" (Forms) page. At the top, there is a navigation bar with links: "Inicio", "Blog", "Correo", "Archivos", "Moodle", and "Formularios". A "Cerrar sesión" (Log out) button is located below the navigation bar. Below the navigation bar is a horizontal bar with links: "Formulario de Alta", "Formulario de Baja", "Cambiar Contraseña", "Baja del Sistema", and "Cambiar Rol". The main content area displays the "Formulario de Alta en el Blog" (Blog Registration Form). The form includes fields for "Nombre de usuario", "Correo electrónico", "Nombre", "Apellido", and "Contraseña". A "Darse de Alta" (Register) button is located at the bottom of the form. The footer of the page reads "© 2024 Carldres - Proveedor de Servicios".

**Figura 42** Formulario de alta en wordpress

En segundo lugar encontramos el formulario de alta en moodle. Es sencillo y solicita los datos básicos. En este caso, la solicitud se envía a un fichero de texto en el sistema donde se encuentran todas las solicitudes relacionadas, y que todas las mañanas se envía al correo del administrador.



**Figura 43** Formulario de alta en moodle

El siguiente formulario es el de solicitud de baja en wordpress, que no es algo automático y que de nuevo se envía a un blog de solicitudes.

A la derecha de este encontramos un formulario que indica cambiar la contraseña pero que al desplegarlo son tres formularios distintos, uno para cambiar la contraseña del sistema, otro para cambiar la contraseña del blog, y otro para cambiar la contraseña de nextcloud. Estos tres formularios también escribirán las solicitudes en un log para enviarlo al correo del administrador.

El siguiente formulario es el de baja total del sistema, donde encontramos una serie de frases que indican las consecuencias que puede llevar la solicitud de esa acción, y evidentemente encontramos un nombre de usuario y contraseña, que se comprobará si coinciden los datos del usuario del sistema. También hemos incluido un campo en el que hay que introducir la palabra clave “CONFIRMAR” para evitar errores en esta petición. Dado a la importancia de esta acción, hemos creído conveniente que las bajas del sistema de momento, las hagan los administradores, de hecho, en el directorio personal del administrador encontramos una serie de archivos de texto que nos dan los pasos para realizar diferentes acciones como dar de baja usuarios o cambiar las contraseñas.



**Formularios**

Inicio Blog Correo Archivos Moodle Formularios

Cerrar sesión

Formulario de Alta Formulario de Baja Cambiar Contraseña Baja del Sistema Cambiar Rol

**Baja del Sistema**

Para confirmar la baja del sistema, por favor escriba "CONFIRMAR" en el siguiente campo de texto. La baja del sistema es irreversible y se eliminarán todos sus datos.

Escriba CONFIRMAR

Nombre de usuario

Contraseña

**Dar de Baja**

© 2024 Cardres - Proveedor de Servicios

**Figura 44** Formulario de baja del sistema

Por último encontramos el formulario para cambiar el rol del usuario de alumno a profesor. Aunque el enunciado indicaba que el usuario introduciría su rol en el formulario, creímos conveniente que esta decisión pasará a través de los administradores. Esta forma de actuar nos resolverá problemas de suplantación de identidad, en los que por ejemplo alumnos se hagan pasar por profesores para tener privilegios.

## Configuraciones adicionales

### Copia de seguridad

En nuestro servidor será muy importante almacenar nuestros archivos varias veces para prevenir su pérdida en catástrofes naturales u otros fallos. Para nuestro caso, hemos decidido realizar copias de seguridad periódicas a un servidor remoto. En este caso vamos a necesitar un servidor con una ip pública aparte del que ya tenemos. Necesitamos que este nuevo servidor tenga instalado apache y el servicio ssh. La ip pública de nuestro servidor remoto será la 34.125.136.211 y la de nuestro servidor local es la 34.16.182.84.

tado	Nombre ↑	Zona	Recomendaciones	En uso por	IP interna	IP externa	Conectar
	<a href="#">cardres</a>	us-west4-b	⚠ Ahorrar \$27/mes		10.182.0.3 (nic0)	34.16.182.84 (nic0)	SSH ▾ ⋮
	<a href="#">cardres-auxiliar</a>	us-west4-b			10.182.0.4 (nic0)	34.125.136.211 (nic0)	SSH ▾ ⋮

Lo siguiente es instalar rsync en nuestro servidor local, que es el que utiliza el nombre cardres. También tenemos que instalar el servicio rsync en el servidor remoto. Es importante comprobar que en ambos servidores, el servicio rsync está activo, por medio del estado del servicio.

Para poder realizar las copias de seguridad de forma automática, es necesario que creemos una clave ssl en nuestro servidor local, y que la copiemos al servidor remoto, para que a la hora de realizar la transferencia de archivos no se solicite la contraseña del usuario en el servidor remoto. El usuario que hemos utilizado en el servidor remoto es otro con nombre carldres, al que nos conectamos con el protocolo ssh. Esquemmatizando un poco el proceso, lo que va a ocurrir, es que a la hora de realizar la copia de seguridad, la orden rsync va a utilizar la clave privada generada por el usuario carldres, y gracias a ella al conectarse al servidor remoto, no se le solicitará ninguna contraseña adicional.

En los archivos incluidos en la copia de seguridad, hemos creído conveniente conservar todos lo que contiene el directorio /var/www, ya que contiene la información de los usuarios en el software utilizado, y sobre todo lo más importante a conservar es el directorio data dentro del directorio nextcloud debido a que es donde se encuentran todos los archivos utilizados en nextcloud separados por los directorios de cada usuario. Además también hemos realizado la copia de seguridad de la base de datos usuarios, que es donde tenemos la información correspondiente a los usuarios del sistema.

El directorio remoto donde hemos realizado esa copia de seguridad es /home/carldres, es decir, el directorio personal del usuario carldres del servidor remoto, el cual es también administrador de ese servidor por lo tanto los archivos estarán seguros gracias a los permisos del directorio.

## Tareas del sistema

El sistema se encarga de realizar algunas tareas de apoyo en el servidor. Una de las tareas del sistema, es que todas las noches realiza los cambios de contraseña solicitados durante el día. Esto lo consigue por medio de la lectura de un log, en el que se han almacenado las peticiones de cambio de contraseña. Es importante mencionar, que antes de almacenar una solicitud de cambio de contraseña, se comprueba si la contraseña actual introducida es correcta, y por lo tanto si lo es, se escribe la petición en el log. Esta tarea se ejecuta por parte del servicio temporizador cambiar\_contrasena\_sistema.timer.

Otra de las tareas principales del sistema, es la creación de usuarios. Cada hora en punto, se ejecuta un servicio que se encarga de comprobar si existen usuarios en la base de datos que están confirmados pero que todavía no tienen un usuario en el sistema LINUX. A estos usuarios se les crea el usuario en el sistema, los directorios personales de usuarios (que no son utilizados, pero los creamos por seguir con las reglas de los sistemas UNIX), y otra de las cosas más importantes es que les crea un usuario automáticamente en nextcloud con el rol de alumno. Esta tarea es ejecutada por el temporizador crea\_usuarios.timer.

A continuación, encontramos la tarea de stats, que consiste en un servicio de temporización (stats.timer) que se ejecuta una vez cada hora, y que recopila información del sistema la guarda en el archivo stats.txt del directorio personal del administrador.

La tarea para realizar las copias de seguridad la ejecuta el temporizador backup.timer.

Otra de las tareas de control es la de eliminar usuarios sin confirmación de la base de datos. Cada noche existe un servicio que ejecuta un script para poder eliminar los usuarios de la base de datos que no están confirmados. Esto nos permite ir limpiando la base de datos de usuarios y evitar que se sobrecargue de información que no es relevante para el sistema. Además nos protegemos ante posibles ataques donde la función principal sea la saturación del sistema por medio de usuarios no confirmados.

Por último encontramos dos servicios que se encargan de tareas muy parecidas, estos son: el servicio email\_stats.timer, que se encarga de enviar las estadísticas del sistema almacenadas en el documento stats.txt del directorio personal del administrador, y el servicio enviar\_correo\_cambios\_contraseña.timer, que aunque se llame cambios de contraseña lo que hace es enviar al correo electrónico del administrador todos los ficheros con solicitudes de cambios de contraseña, con altas de usuarios en aplicaciones,etc. También se envía un fichero log donde se almacenan todos los accesos al servidor, tanto los correctos como los incorrectos. Una vez enviados los correos, los archivos se borran y vuelven a escribirse de cero durante el día. Estos correos se envían todas las mañanas a las 10, cuando creemos que es seguro que el administrador esté en su puesto de trabajo.

```
carldres@carldres:~$ sudo systemctl list-timers --all
```

NEXT	LEFT	LAST	PASSED	UNIT
Mon 2024-05-20 19:21:25 UTC	1min 4s left	Mon 2024-05-20 19:11:25 UTC	8min ago	gce-workload-cert-refresh.timer
Mon 2024-05-20 19:39:00 UTC	18min left	Mon 2024-05-20 19:09:00 UTC	11min ago	phpsessionclean.timer
Mon 2024-05-20 20:00:00 UTC	39min left	Mon 2024-05-20 19:00:02 UTC	20min ago	crea_usuarios.timer
Mon 2024-05-20 20:00:00 UTC	39min left	Mon 2024-05-20 19:00:02 UTC	20min ago	stats.timer
Mon 2024-05-20 22:04:47 UTC	2h 44min left	Mon 2024-05-20 16:04:47 UTC	3h 15min ago	google-oslogin-cache.timer
Tue 2024-05-21 00:00:00 UTC	4h 39min left	Mon 2024-05-20 00:00:19 UTC	19h ago	backup.timer
Tue 2024-05-21 00:00:00 UTC	4h 39min left	Mon 2024-05-20 00:00:19 UTC	19h ago	cambiar_contraseña_sistema.timer
Tue 2024-05-21 00:00:00 UTC	4h 39min left	Mon 2024-05-20 00:00:19 UTC	19h ago	eliminar_usuarios_sin_confirmacio
Tue 2024-05-21 00:00:00 UTC	4h 39min left	Mon 2024-05-20 00:00:19 UTC	19h ago	email_stats.timer
Tue 2024-05-21 00:00:00 UTC	4h 39min left	Mon 2024-05-20 00:00:19 UTC	19h ago	logrotate.timer
Tue 2024-05-21 00:00:00 UTC	4h 39min left	Mon 2024-05-20 00:00:19 UTC	19h ago	man-db.timer
Tue 2024-05-21 05:12:40 UTC	9h left	Mon 2024-05-20 09:47:30 UTC	9h ago	apt-daily.timer
Tue 2024-05-21 06:25:34 UTC	11h left	Mon 2024-05-20 06:27:23 UTC	12h ago	apt-daily-upgrade.timer
Tue 2024-05-21 08:00:00 UTC	12h left	Mon 2024-05-20 08:00:19 UTC	11h ago	enviar_correos_cambios_contraseña
Tue 2024-05-21 10:19:39 UTC	14h left	Mon 2024-05-20 10:19:39 UTC	9h ago	systemd-tmpfiles-clean.timer

# Bibliografía

- Learn Linux TV. (2022, 22 agosto). *Build an Awesome Nextcloud Server (Updated for Ubuntu 22.04!)* [Video]. YouTube.  
<https://www.youtube.com/watch?v=5lUKE3oA7AY>
- Schroder, C. (2021). *Linux Cookbook*. «O'Reilly Media, Inc.»
- *How to Send Email on Ubuntu from Gmail (SMTP Postfix tutorial) – Tony Teaches Tech*. (2020, 9 diciembre).  
<https://tonyteaches.tech/postfix-gmail-smtp-on-ubuntu/>
- Wong, L. Z. (2014, 22 septiembre). *How To Configure WebDAV Access with Apache on Ubuntu 14.04*. DigitalOcean.  
<https://www.digitalocean.com/community/tutorials/how-to-configure-webdav-access-with-apache-on-ubuntu-14-04>
- Clockwork Computer. (2023, 16 enero). 🐧 *Servidor DNS con BIND9 en Ubuntu Server 22.04* [Video]. YouTube.  
[https://www.youtube.com/watch?v=jq5potgQ7\\_k](https://www.youtube.com/watch?v=jq5potgQ7_k)
- Clockwork Computer. (2023b, octubre 29). 🐧 *SERVIDOR DE CORREO LOCAL con POSTFIX - DOVECOT - ROUNDROBIN Y THUNDERBIRD* [Video]. YouTube. <https://www.youtube.com/watch?v=KqHiLGgaolQ>
- *NextCloud – Complete setup Guide – Learn Linux TV*. (s. f.).  
<https://www.learnlinux.tv/nextcloud-complete-setup-guide/>
- *curl:(51) : SSL certificate subject name does not match target host name*. (s. f.). Stack Overflow.  
<https://stackoverflow.com/questions/44445368/curl51-ssl-certificate-subject-name-does-not-match-target-host-name>

- *Servicios de cloud computing | Google Cloud.* (s. f.). Google Cloud.  
<https://cloud.google.com/?hl=es>
- Bunce, T. (s. f.). *DBI. Database Independent Interface For Perl* -  
metacpan.org. <https://metacpan.org/pod/DBI>
- *Instalando Moodle en distribuciones basadas en Debian - MoodleDocs.* (s. f.).  
[https://docs.moodle.org/all/es/Instalando\\_Moodle\\_en\\_distribuciones\\_basadas\\_en\\_Debian#Instalar\\_Moodle](https://docs.moodle.org/all/es/Instalando_Moodle_en_distribuciones_basadas_en_Debian#Instalar_Moodle)
- *How to Send Email on Ubuntu from Gmail (SMTP Postfix tutorial) – Tony Teaches Tech.* (2020b, diciembre 9).  
<https://tonyteaches.tech/postfix-gmail-smtp-on-ubuntu/>
- *Build software better, together.* (s. f.). GitHub.  
<https://gist.github.com/parrazam/9b8c86458196e3e910d07b5880d727c3/revisions>
- Learn Linux TV. (2023b, febrero 7). *Essential First Steps for Every New Linux Server Build* [Video]. YouTube.  
<https://www.youtube.com/watch?v=GUpZELktYKQ>
- *Installing Moodle on Debian based distributions - MoodleDocs.* (s. f.).  
[https://docs.moodle.org/404/en/Installing\\_Moodle\\_on\\_Debian\\_based\\_distributions#Installing\\_moodle\\_from\\_.tgz\(.tar.gz\)\\_or\\_.zip\\_file](https://docs.moodle.org/404/en/Installing_Moodle_on_Debian_based_distributions#Installing_moodle_from_.tgz(.tar.gz)_or_.zip_file)
- *Installing Moodle on Debian based distributions - MoodleDocs.* (s. f.).  
[https://docs.moodle.org/404/en/Installing\\_Moodle\\_on\\_Debian\\_based\\_distributions#Installing\\_moodle\\_from\\_.tgz\(.tar.gz\)\\_or\\_.zip\\_file](https://docs.moodle.org/404/en/Installing_Moodle_on_Debian_based_distributions#Installing_moodle_from_.tgz(.tar.gz)_or_.zip_file)