

Memoria

Práctica Final Obligatoria

Realizado por

Julia García Vega y Jorge Pozo Zapatero

Administración de Sistemas 2022-2023

1. Índice

| | |
|--|-----------|
| 1. Índice | 2 |
| 2. Introducción | 4 |
| 3. Descripción básica del sistema | 4 |
| 3.1. Interfaces Web | 5 |
| 3.1.1. Pantalla principal | 5 |
| 3.1.2. Pantalla de registro | 6 |
| 3.1.3. Pantalla de registro | 7 |
| 3.1.4. Correo con clave de confirmación | 7 |
| 3.1.5. Página para confirmar cuenta | 8 |
| 3.1.6. Página de registro confirmado | 9 |
| 3.1.7. Página de inicio de sesión | 9 |
| 3.1.8. Página de servicios | 10 |
| 3.1.9. Página de estado del sistema | 10 |
| 3.1.10. Página de perfil | 11 |
| 3.1.11. Página de contacto | 12 |
| 3.1.12. Página de ayuda | 12 |
| 4. Descripción básica de los servidores fundamentales | 13 |
| 4.1. Configuración de VirtualBox | 13 |
| 4.2. Servidores y herramientas instalados | 14 |
| 4.2.1 Apache2 | 14 |
| 4.2.2 MariaDB | 16 |
| 4.2.3 Postfix | 18 |
| 4.2.4 Dovecot | 21 |
| 4.2.5 SSH (SFTP) - albergue de páginas web | 24 |
| 4.2.6 Thunderbird | 25 |
| 4.2.7 Nagios | 28 |
| 4.2.8 Wordpress | 34 |
| 4.2.9 Tripwire | 38 |
| 4.2.13 Crontab | 40 |
| 4.2.14 Certificado https | 40 |
| 5. Ficheros fuente PERL | 42 |
| Acceder a la base de datos | 42 |
| Obtener los parámetros de un formulario en html | 43 |
| Sesiones en perl con CGI::Session(): | 43 |
| Uso del módulo Template | 44 |

| | |
|---|-----------|
| 5.1. registro.pl | 44 |
| 5.2. comprobarSolicitudesAlta.pl | 44 |
| 5.3. confirmarRegistro.pl | 45 |
| 5.4. inicioSesion.pl | 46 |
| 5.5. serviciosUsuario.pl | 46 |
| 5.6. contratarCorreo.pl | 46 |
| 5.7. finalizarCorreo.pl | 46 |
| 5.8. contratarBlog.pl | 47 |
| 5.9. finalizarBlog.pl | 47 |
| 5.10. contratarWeb.pl | 47 |
| 5.11. finalizarWeb.pl | 48 |
| 5.12. ejecutarObtenerEstadoServicios.pl | 48 |
| 5.13. obtenerEstadoServicios.sh | 48 |
| 5.14. estadoSistema.pl | 48 |
| 5.15. verPerfil.pl | 48 |
| 5.15. modificarPerfil.pl | 49 |
| 5.16. guardarModificarPerfil.pl | 49 |
| 5.17. eliminarPerfil.pl | 49 |
| 5.18. contacto.pl | 49 |
| 5.19. ayuda.pl | 49 |
| 5.20. cerrarSesion.pl | 49 |
| 6. Conclusión | 50 |
| 7. Referencias | 50 |

2. Introducción

El objetivo de este informe es documentar de forma exhaustiva el proceso de creación y los resultados obtenidos de la creación y puesta en marcha de una solución informática y de gestión enfocada a una empresa emergente de servicio informático (EESI) ficticia.

Esta solución buscará diseñar, instalar y configurar un servidor LINUX que brinde soporte telecinco integral a todos los clientes, la provisión de servicios de Internet que serán un servidor de correo que permita a los usuarios el envío y recepción de correos electrónicos, un albergue de páginas web, donde los usuarios podrán subir y ver sus propias páginas web y un servicio de blog donde los usuarios podrán acceder para subir sus propias entradas o para interactuar con las entradas en el blog del resto de usuarios.

El informe presentará paso a paso como se ha desarrollado esta solución de manera que se pudiese repetir el proceso mediante el correcto seguimiento del informe planteado.

Se comenzará realizando una descripción de las interfaces web y su interacción entre ellas mediante el uso de html, css y PERL, para continuar viendo la instalación de los servidores y herramientas utilizados para brindar los servicios estipulados.

El nombre de la empresa elegido será GarciaPozoIT en relación con los autores del proyecto y IT porque el tema de la empresa serán las tecnologías.

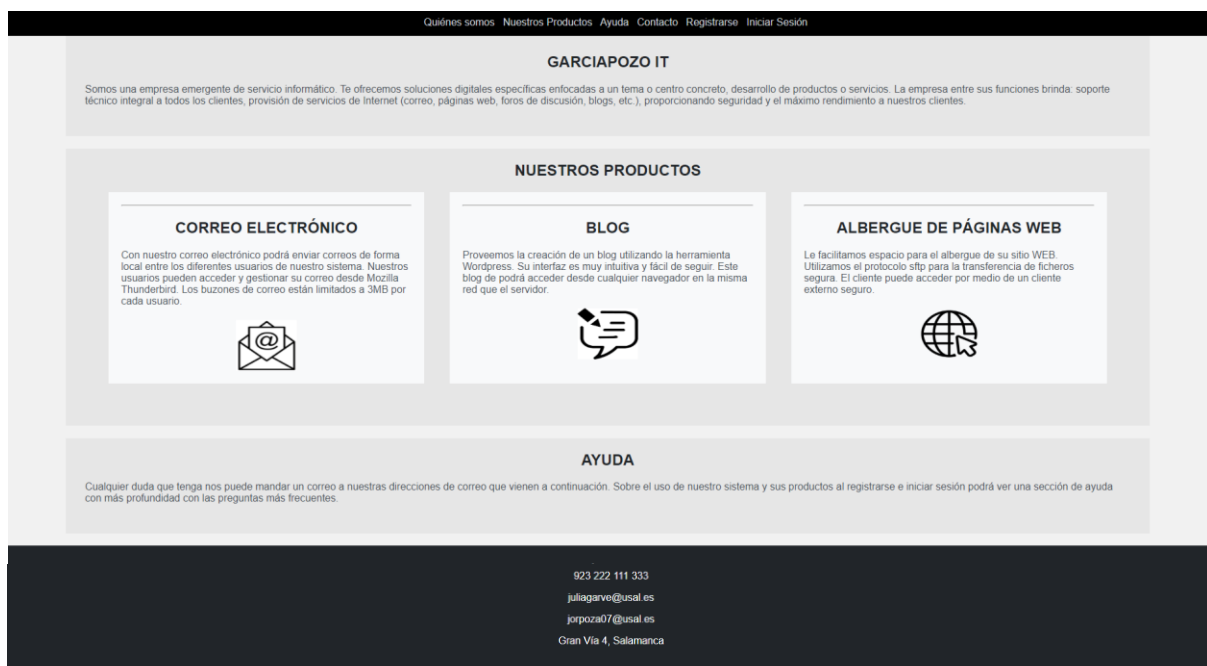
3. Descripción básica del sistema

Este apartado explicará detalladamente y de forma gráfica el contenido de las distintas interfaces web utilizadas, además de las interacciones que se forman entre ellas. Mediante el uso de los lenguajes html, utilizado para la visualización de las páginas web en una aplicación de navegador web como puede ser Mozilla Firefox, css, para la creación de las hojas de estilo que se utilizarán conjuntamente junto a html para mostrar los contenidos de manera correcta y PERL, que será utilizado para las propias labores de desarrollo y programación web que será el encargado de manejar la información y la interacción entre los usuarios y las páginas web utilizadas.

3.1. Interfaces Web

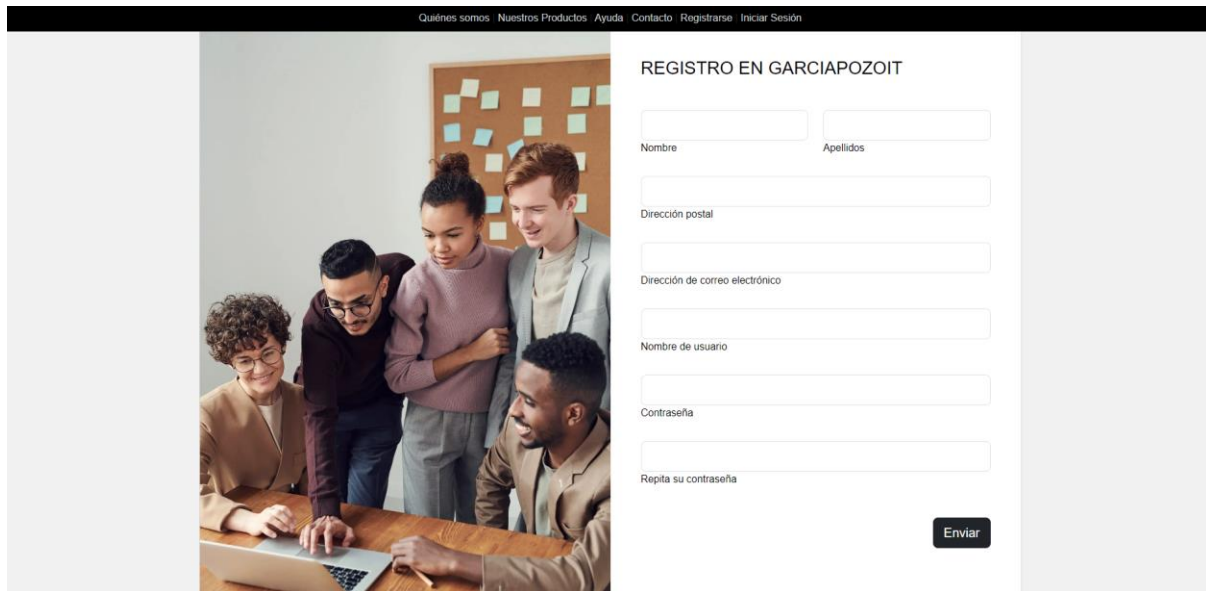
Para el desarrollo de las siguientes interfaces Web se ha utilizado la biblioteca de W3.CSS (W3.CSS Home, n.d.) y Bootstrap (Equipo Bootstrap, n.d.) para mejorar la visualización y ahorrar en el tiempo de desarrollo. También para los iconos se ha utilizado la biblioteca de Font Awesome 5 (*Font Awesome 5 Introduction*, n.d.). Además, para algunas interfaces también nos hemos basado en otras plantillas a las que daremos crédito en su respectiva interfaz.

3.1.1. Pantalla principal



Esta pantalla es la primera pantalla a la que se accede al entrar en nuestro servidor, se trata del "index.html". Consta de un menú de navegación superior estático para las distintas secciones. Tanto las secciones de "Quiénes somos", "Nuestros productos", "Ayuda" y "Contacto" se encuentran en esta misma página pero las secciones de "Registrarse" e "Iniciar sesión" se encuentran en sus propias páginas. De esta manera, presentamos en esta interfaz nuestra empresa, sus productos, nuestra información de contacto y una pequeña sección de ayuda.

3.1.2. Pantalla de registro



Quénes somos Nuestros Productos Ayuda Contacto Regístrate Iniciar Sesión

REGISTRO EN GARCIAPOZOIT

Nombre Apellidos

Dirección postal

Dirección de correo electrónico

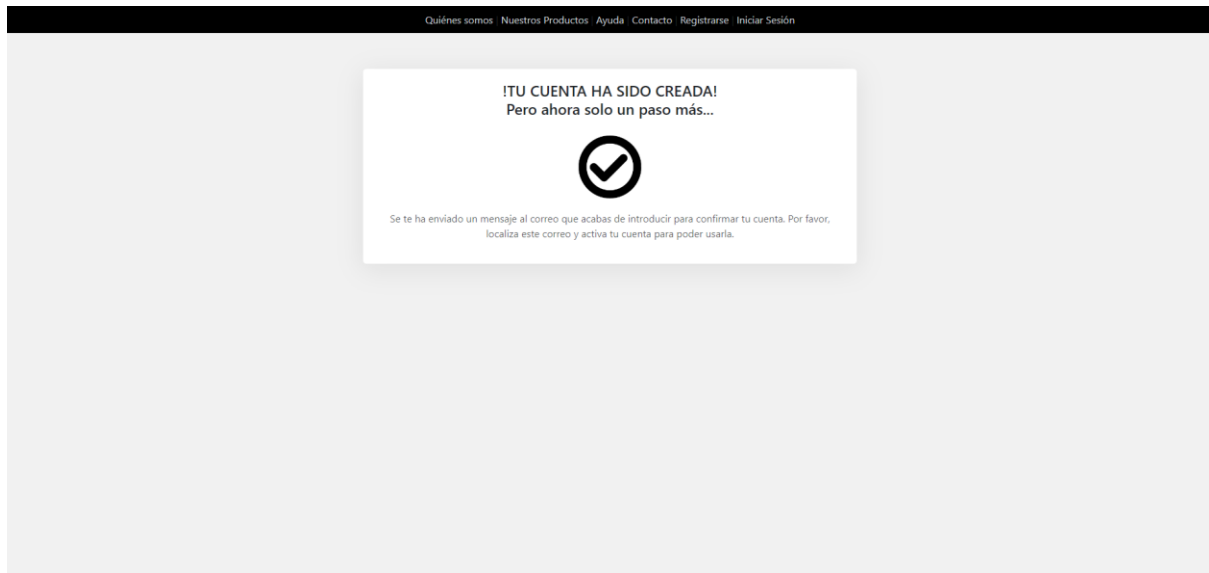
Nombre de usuario

Contraseña

Repita su contraseña

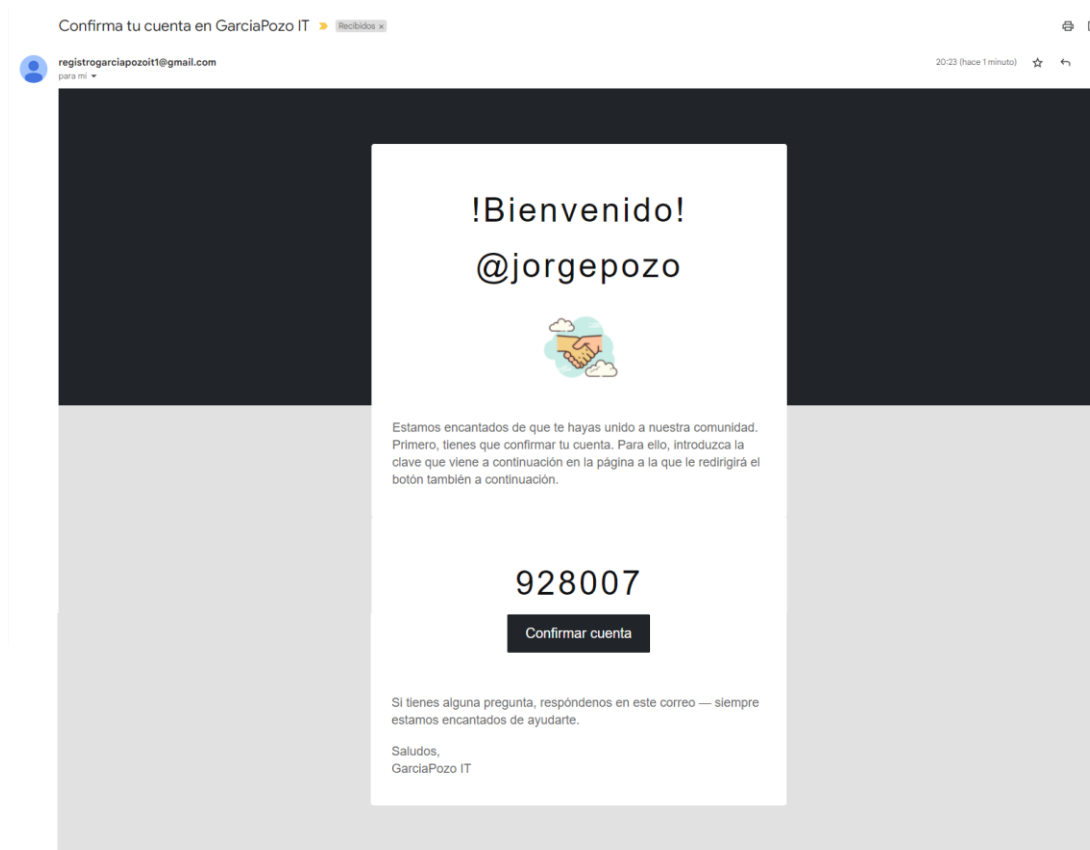
La pantalla de registro consiste en un formulario en html que al seleccionar el botón enviar se ejecuta el script de perl de registro de un nuevo usuario. Para enviar correctamente el cuestionario todos los campos deben estar rellenos, lo que se ha hecho con la propiedad `required` en los inputs. Además las contraseñas al introducirlas se comprueba que coincidan mediante una función de javascript en el html y si no no deja enviar el formulario (*Can You Require Two Form Fields to Match With HTML?*, n.d.). También es un requerimiento para enviar el formulario que la contraseña tenga mínimo 8 caracteres, 1 número y 1 mayúscula y que el formato de correo sea el correcto (*HTML Input Pattern Attribute*, n.d.). Si al enviar el formulario, el script de perl detectara que ya existe un usuario con ese nombre de usuario registrado advertirá al cliente para que introduzca un nombre de usuario distinto. Basado en: (*Bootstrap Registration Form - Free Examples & Tutorial*, n.d.)

3.1.3. Pantalla de registro



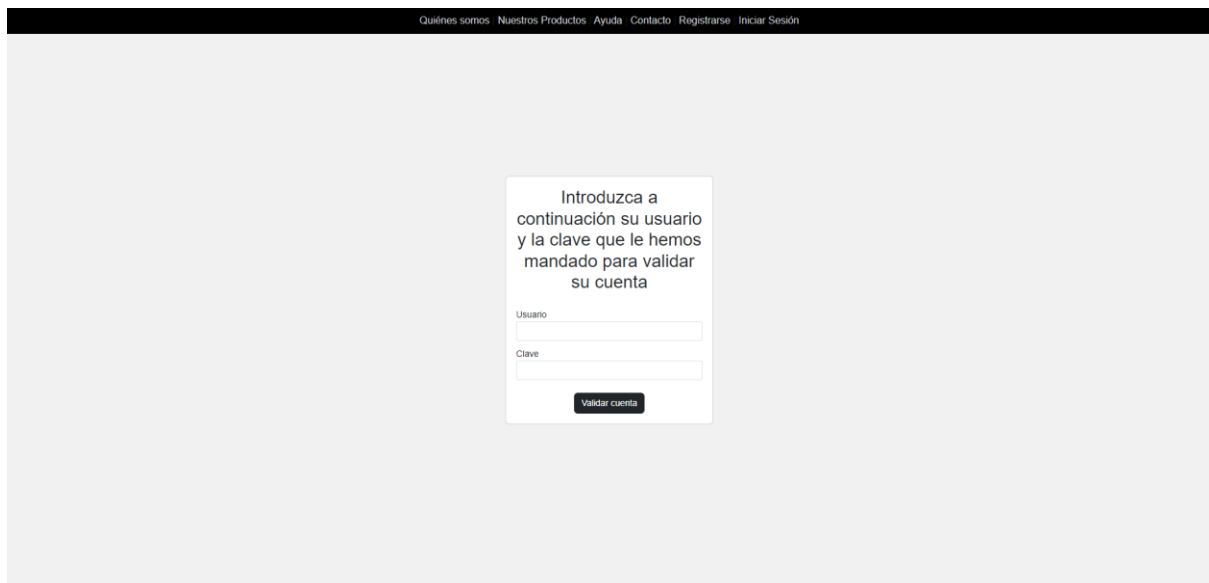
Tras enviar correctamente el formulario, se dirige al usuario a esta pantalla para informarle que tiene que ir al correo electrónico que ha introducido en el formulario para confirmar la cuenta. Basado en: (*Bootstrap Snippet. Confirmation Email*, 2016)

3.1.4. Correo con clave de confirmación



En este correo se da la bienvenida al usuario a nuestro sistema y se le informa que tiene que confirmar su cuenta. Para ello se le proporciona una clave aleatoria de 6 números que debe introducir en la página a la que le redirige el botón de confirmar cuenta. Basado en: (*Bootstrap 4 Confirm Account Welcome Email Template Example*, n.d.).

3.1.5. Página para confirmar cuenta



Quiénes somos · Nuestros Productos · Ayuda · Contacto · Regístrate · Iniciar Sesión

Introduzca a continuación su usuario y la clave que le hemos mandado para validar su cuenta

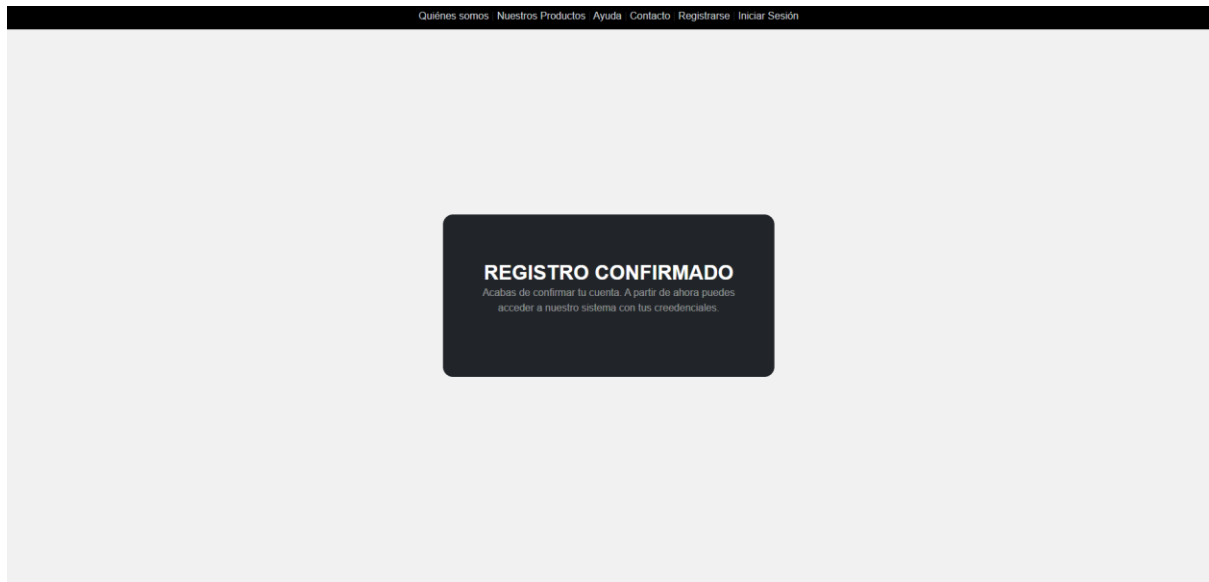
Usuario

Clave

Validar cuenta

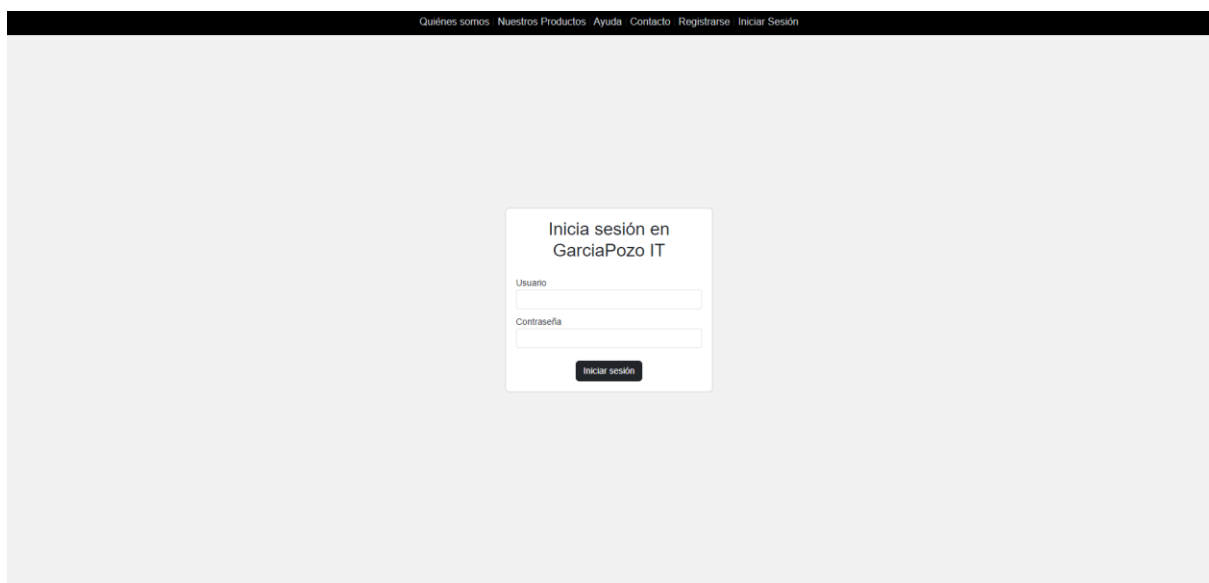
En esta página, a la que le redirige el correo con la clave, se solicita el nombre de usuario y la clave que se le ha mandado al cliente a su correo para confirmar su cuenta. Se trata otra vez de un formulario en html que al presionar el botón de validar cuenta se ejecuta un script de perl para actualizar la base de datos. Si la información introducida fuese incorrecta se le informaría al usuario y se le permitiría volver a introducir los datos.

3.1.6. Página de registro confirmado



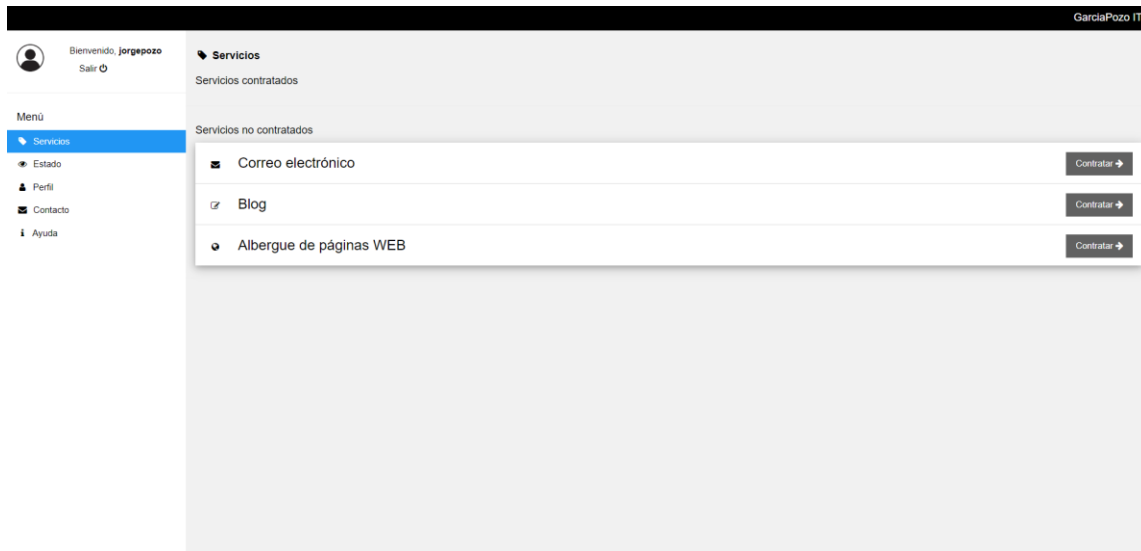
Tras introducir correctamente los datos en la pantalla anterior, el usuario ve esta pantalla en la que se le informa de que su registro ya se ha finalizado correctamente y de que ya puede acceder al sistema con sus credenciales.

3.1.7. Página de inicio de sesión



Esta es la pantalla de inicio de sesión en la aplicación web de nuestro sistema. Se trata de un formulario html que solicita obligatoriamente los campos de usuario y contraseña. Al enviar el formulario seleccionando el botón se ejecuta un script de perl que compara los datos introducidos con la base de datos y si son credenciales válidos le redirige a la interfaz principal del usuario y si no le informa y le permite volver a introducirlos.

3.1.8. Página de servicios



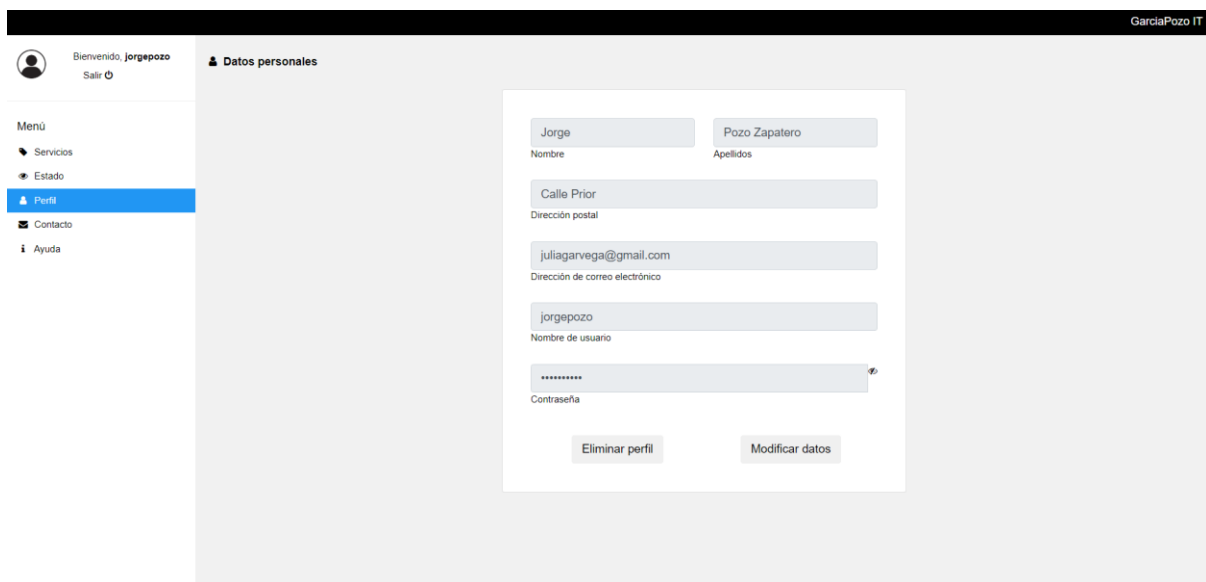
Al iniciar sesión, se redirige al usuario a la pantalla de servicios. En esta pantalla se informa al usuario de que servicios tiene contratados y cuáles no. Aquellos que no tenga contratados se le permite contratarlos seleccionando el botón de contratar y aquellos que ya tuviese contratos se le permite darlos de baja seleccionando el botón de finalizar. Al seleccionar cualquiera de los dos botones aparecerá en pantalla un pop-up solicitando la confirmación de la acción. Tras dar de alta un servicio, este ya le será disponible al usuario y pasará a estar en un lista de servicios contratados y viceversa. Basada en: (W3.CSS Templates, n.d.) Desde el menú a la izquierda se puede desplazar a las otras ventanas de la interfaz principal.

3.1.9. Página de estado del sistema



En esta página se informa al usuario sobre el estado de los distintos servicios del sistema que le afectan como son: postfix y dovecot para el correo, ssh para el protocolo sftp para el albergue de páginas web, rsyslog para mantener los logs, mariadb para la base de datos y apache2 para mantener el servidor web. Los servicios que se encuentren activos estarán coloreados de verde y se indicará su estado activo con un *checkmark* a la derecha. Aquellos que no estén activos se colorearán de rojo, indicando su estado también a la derecha y con un icono de una cruz.

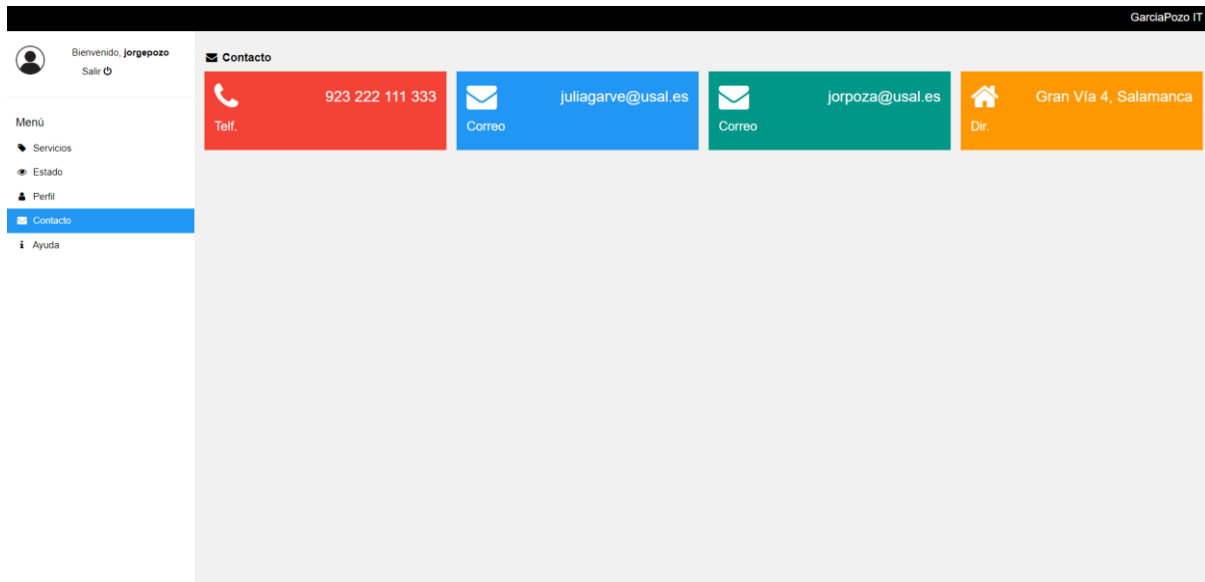
3.1.10. Página de perfil



The screenshot shows a web application interface for a user profile. At the top right, it says "GarcíaPozo IT". On the left, there is a sidebar with a "Menú" section containing links for "Servicios", "Estado", "Perfil" (which is highlighted in blue), "Contacto", and "Ayuda". The main content area is titled "Datos personales" and contains a form with the following fields: "Nombre" (Jorge), "Apellidos" (Pozo Zapatero), "Dirección postal" (Calle Prior), "Dirección de correo electrónico" (juliagarvega@gmail.com), "Nombre de usuario" (jorgepozo), and "Contraseña" (masked with asterisks). At the bottom of the form are two buttons: "Eliminar perfil" and "Modificar datos".

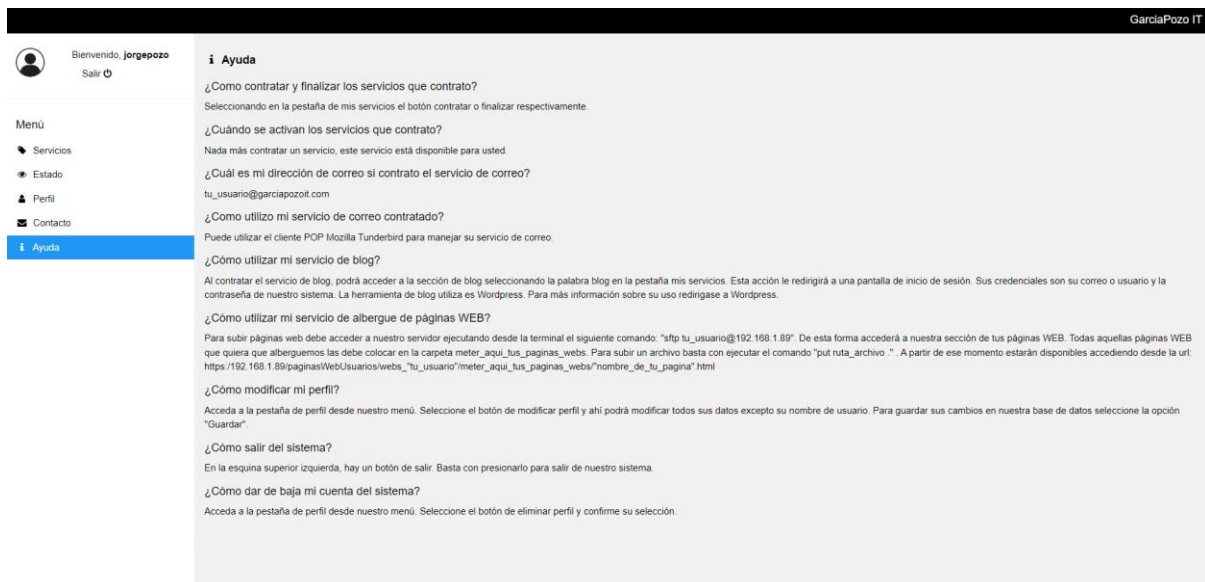
Se muestra en esta sección un formulario inicialmente no editable con los datos del usuario logueado. Desde esta sección puede tanto modificar como eliminar su perfil. Al seleccionar el botón de modificar datos, el cuestionario se volverá editable salvo el campo de nombre de usuario que no permitimos modificarlo. El usuario puede modificar los campos que desee y cuando haya finalizado puede seleccionar guardar o cancelar para guardar o no sus modificaciones. Al seleccionar el botón eliminar perfil, se le mostrará un pop-up para confirmar la eliminación de perfil informándole que se eliminarán todos sus datos y no podrá recuperarlos. Si acepta, se ejecutará un script que limpiará su rastro del sistema.

3.1.11. Página de contacto



En esta pantalla se muestra la información de contacto de nuestra empresa, incluyendo un supuesto número de teléfono fijo, el correo de los dos administradores y la supuesta dirección del local de la empresa.

3.1.12. Página de ayuda



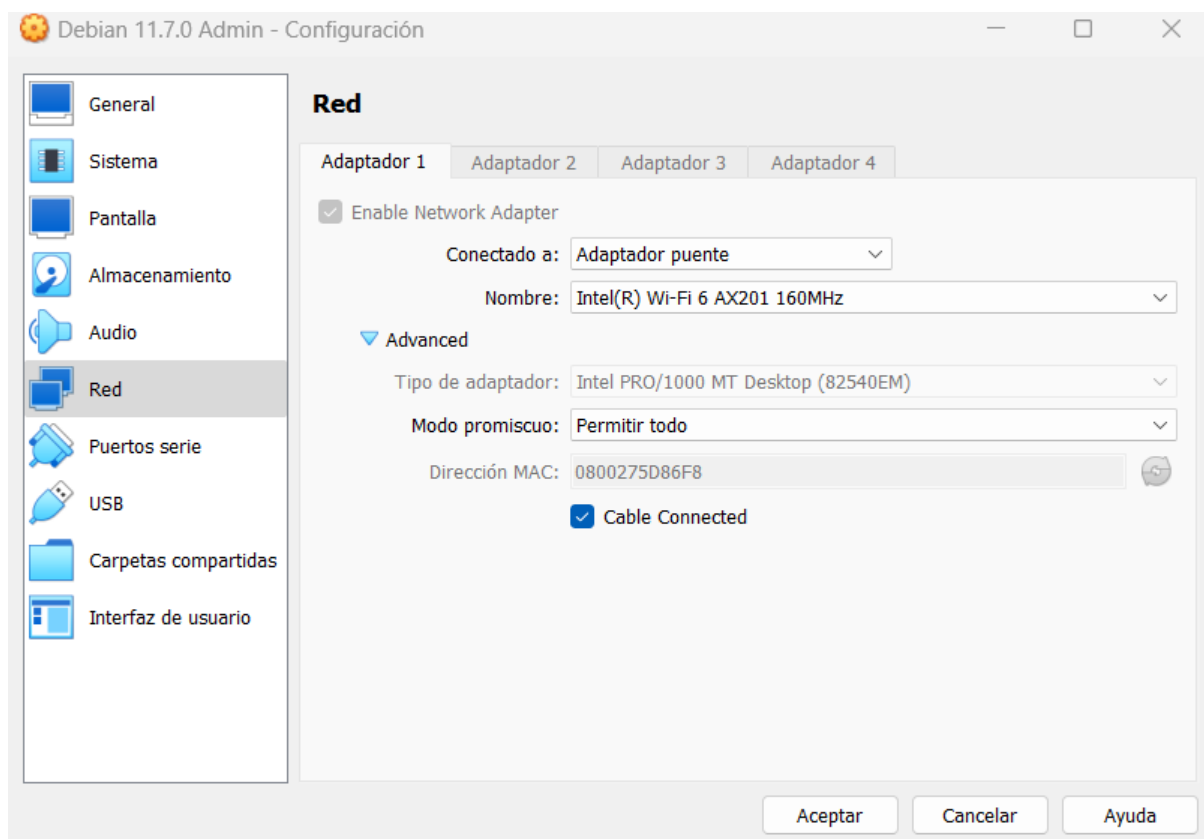
En esta ventana se muestra un listado de preguntas frecuentes que explican al usuario como realizar las distintas tareas que ofrece el sistema. Por último desde todas estas pantallas del usuario, debajo de usuario, en la esquina superior izquierda hay un botón de salir del sistema para cerrar sesión y volver a ir a la pantalla de inicio de sesión.

4. Descripción básica de los servidores fundamentales

En este apartado detallaremos los diferentes servidores que hemos implementado para ofrecer los servicios a los usuarios que queremos ofrecer. Hablaremos en primer lugar sobre la configuración de la máquina virtual en la que se ha realizado el despliegue de los servidores para continuar hablando sobre los servidores y herramientas implementados para cumplir los fines establecidos. Describiremos el uso que le damos a estas herramientas y servidores, para qué sirven y la instalación que hemos realizado con las opciones de configuración elegidas.

4.1. Configuración de VirtualBox

Para realizar el servidor de la práctica hemos utilizado una instalación limpia de Debian 11 en una máquina virtual de VirtualBox. Para poder conectar esta máquina con el resto de ordenadores de la red hay que modificar su configuración. En el apartado de red de la máquina virtual en VirtualBox, en la caja “conectado a:” debe estar elegido el adaptador puente y también estar en modo promiscuo permitiendo todo.



Además, para poder acceder a nuestro servidor, nos referimos a él por la ip para no tener que configurar el fichero hosts en todos los equipos de la red para que todos puedan acceder. Para que esta ip se mantenga siempre igual y así no tengamos que ir modificando los ficheros, en los ajustes de red del Debian hemos establecido la ip como estática en vez de por DHCP.

Por este motivo, cuando tengamos que defender la práctica, al conectarnos a una red distinta de con la que hemos realizado la práctica tendremos que modificar en los ficheros que esté la ip del servidor con la ip del servidor en esa red que nos conectemos en la defensa.

4.2. Servidores y herramientas instalados

4.2.1 Apache2

Comenzaremos hablando sobre el servidor de Apache2, puesto que este es la base de nuestro sistema. Apache será el encargado de recibir las peticiones de los clientes a través de protocolo HTTP y mediante este protocolo brindarles los servicios principales que están relacionados con una interfaz web.

Este servidor de código abierto y gratuito nos permitirá lograr el objetivo de recibir clientes provenientes tanto de diferentes sistemas operativos como son Linux y Windows.

Para realizar la instalación y configuración de Apache2 seguiremos los siguientes pasos:

En primer lugar veremos si existen actualizaciones disponibles:

```
root@garciapozoit:/# apt-get update
Hit:1 http://deb.debian.org/debian bullseye InRelease
Hit:2 http://security.debian.org/debian-security bullseye-security InRelease
Hit:3 http://deb.debian.org/debian bullseye-updates InRelease
Reading package lists... Done
```

En caso de que existan actualizaciones disponibles deberemos instalarlas mediante el siguiente comando:

```
root@garciapozoit:/# apt-get upgrade
```

En caso de que no existan no sería necesario y podríamos proseguir con los siguientes pasos directamente. Este paso será obviamente descargar Apache2:

```
root@garciapozoit:/# apt-get install apache2
```

Podemos comprobar la versión de apache2 que tenemos además del estado del servicio:

```
root@garciapozoit:/# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-05-20 23:39:49 CEST; 19h ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 817 (apache2)
    Tasks: 12 (limit: 9473)
   Memory: 38.0M
      CPU: 11.101s
   CGroup: /system.slice/apache2.service
           └─ 817 /usr/sbin/apache2 -k start
           └─ 4273 /usr/sbin/apache2 -k start
           └─ 4276 /usr/sbin/apache2 -k start
           └─ 4278 /usr/sbin/apache2 -k start
           └─ 4279 /usr/sbin/apache2 -k start
           └─ 4326 /usr/sbin/apache2 -k start
           └─ 5370 /usr/sbin/apache2 -k start
           └─ 5371 /usr/sbin/apache2 -k start
           └─ 5372 /usr/sbin/apache2 -k start
           └─ 5373 /usr/sbin/apache2 -k start
           └─ 11219 /usr/sbin/apache2 -k start
           └─ 15657 /usr/sbin/apache2 -k start

May 20 23:39:44 garciapozoit.garciapozoit.com systemd[1]: Starting The Apache HTTP Server...
May 20 23:39:49 garciapozoit.garciapozoit.com systemd[1]: Started The Apache HTTP Server.
May 21 00:00:18 garciapozoit.garciapozoit.com systemd[1]: Reloading The Apache HTTP Server.
May 21 00:00:18 garciapozoit.garciapozoit.com systemd[1]: Reloaded The Apache HTTP Server.
root@garciapozoit:/# apache2 -v
Server version: Apache/2.4.56 (Debian)
Server built: 2023-04-02T03:06:01
```

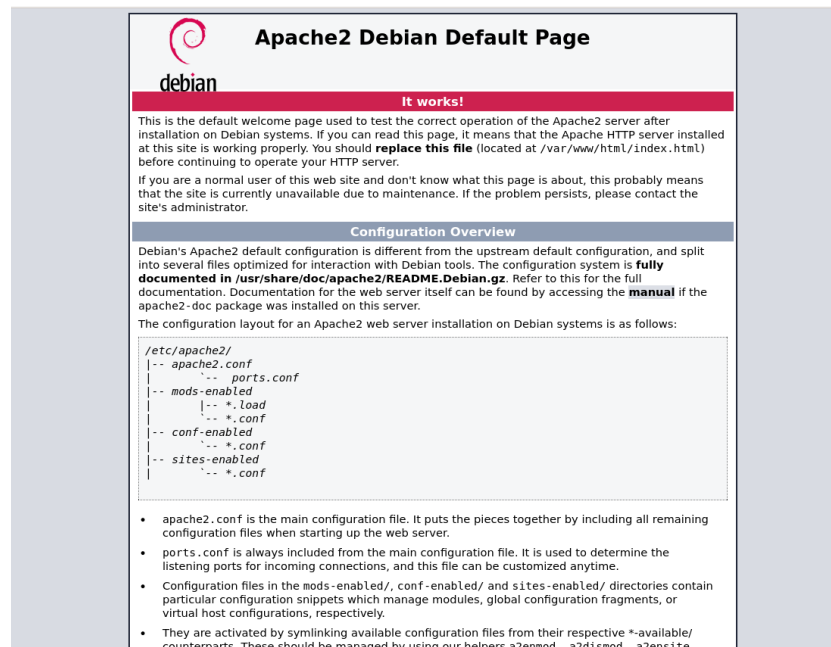
Vemos que está activo correctamente mediante el comando `systemctl status` y la versión con la opción propia `-v` de `apache2`.

Una vez hemos realizado la instalación de `apache2` podemos comprobar si ha funcionado correctamente la instalación mediante el acceso a través de un navegador web a una de las siguientes posibilidades:



Que es en el primer caso la ip del servidor (en el caso de que se esté realizando la prueba en la propia máquina podría utilizarse la ip de loopback 127.0.0.1) o bien poniendo el nombre de este en el caso de la imagen localhost puesto que se está realizando en la propia máquina.

Cuando accedemos a estas direcciones tendremos que obtener la siguiente salida:



Si la hemos obtenido significa que los pasos se han seguido correctamente y el servidor de apache2 ya está en funcionamiento.

Crearemos nuestra propia carpeta dentro de `/var/www` para guardar nuestras propias páginas que se van a mostrar en el navegador web y a través de las cuales los usuarios podrán interactuar. En nuestro caso se llamará como el propio nombre de la empresa garciapozoit.

4.2.2 MariaDB

Utilizaremos MariaDB como base de datos principal y única para guardar la información necesaria para nuestros servidores y sistema. Esta es una opción más segura que MySQL y ofrece mucha compatibilidad con la mayoría de herramientas y servidores que se van a utilizar.

Para instalar esta herramienta:


```
root@garciapozoit:/# apt-get install mariadb-server mariadb-client
```

Mediante el fichero `mysql_secure_installation` podremos configurar la herramienta para que sea más segura con opciones como deshabilitar los accesos de root remotos o eliminar las cuentas anónimas.

Para entrar a utilizar la herramienta utilizaremos el comando `mariadb -u root -p`, que nos solicitará crear una contraseña para la cuenta de root en caso de que no la tengamos.

Aquí crearemos el usuario admin con el que accederemos a partir de ahora y le asignaremos una contraseña, por lo que a partir de ahora accederemos mediante el comando `mariadb -u admin -p`.

Desde aquí ya podremos crear nuestra base de datos:

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

Que llamaremos DBgarciapozoit y podremos acceder a esta de la siguiente manera:

```
MariaDB [(none)]> use DBgarciapozoit;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

Database changed

```
MariaDB [DBgarciapozoit]>
```

Donde estarán las tablas que creemos para la distinta información que vamos a almacenar. Crearemos una primera tabla de usuarios donde almacenaremos la información de los usuarios del sistema:

```
MariaDB [DBgarciapozoit]> create table usuarios (nombreUsuario varchar(20) not null primary key, nombre varchar(20) not null , apellido varchar(50) not null , contrasena varchar(30) not null, direccionP varchar(50) not null , direccionC varchar(50) not null, correo integer, blog integer, web integer, webCreado integer);
Query OK, 0 rows affected (0.096 sec)
```

Donde se introducirán la información de los usuarios de manera automática cuando se registren en el sistema:

```
MariaDB [DBgarciapozoit]> select * from usuario;
```

| nombreUsuario | nombre | apellido | contrasena | direccionP | direccionC | estado | correo | blog | web | clave | creadoWeb |
|---------------|--------|---------------|------------|----------------------|------------------------|--------|--------|------|-----|--------|-----------|
| jorgepozo | Jorge | Pozo Zapatero | Benavente2 | Calle Prior | juliagarvega@gmail.com | 3 | 0 | 0 | 0 | 928007 | 0 |
| juliagvl | Julia | Garcia Vega | Salamanca4 | Calle Conde Don Vela | juliagarve@gmail.com | 3 | 1 | 0 | 0 | 497426 | 0 |

Los accesos a la base de datos a partir de ahora serán todos automáticos, mediante ficheros de perl, por tanto no será necesario a partir de ahora modificar nada ni acceder de manera manual.

Y por último en la tabla virtual_aliases se guardarán los alias de las cuentas que hemos creado en caso de que tengan:

```
MariaDB [DBgarciapozoit]> select * from virtual_aliases;
```

| id | domain_id | source | destination |
|----|-----------|-----------------------------|-------------------------|
| 1 | 1 | soporte@garciapozoit.com | email1@garciapozoit.com |
| 2 | 1 | postmaster@garciapozoit.com | email2@garciapozoit.com |

En estas tablas se introducirá o sacará la información correspondiente a los usuarios que hayan dado de alta o de baja el servicio de correo electrónico. Además Postfix obliga a tener la contraseña encriptada de manera que aunque alguien consiguiese el acceso a la base de datos no tendría posibilidad de extraer las contraseñas.

Lo siguiente que tendríamos que hacer respecto a la configuración de postfix sería editar la configuración de este. Existen dos modos de realizarlo: mediante el comando `dpkg -reconfigure postfix` o directamente editando los ficheros de su configuración. En nuestro caso hemos realizado la edición directamente de estos ficheros.

De los cambios en los ficheros de configuración de postfix que son principalmente los de los archivos `main.cf` y `master.cf` que se encuentran en la ruta `/etc/postfix/`.

Dentro de `main.cf` destacaremos las siguientes líneas:

```
mydomain = garciapozoit.com
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
#mydestination = garciapozoit.org, garciapozoit.myquest.virt
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 3145728
message_size_limit = 50000
```

```
virtual_mailbox_domains= mysql:/etc/postfix/mysql-virtual-mailbox-domains.cf
virtual_mailbox_maps= mysql:/etc/postfix/mysql-virtual-mailbox-maps.cf
virtual_alias_maps = mysql:/etc/postfix/mysql-virtual-alias-maps.cf
```

#mas lineas añadidas

```
home_mailbox= Maildir/
virtual_transport= lmtp:unix:private/dovecot-lmtp
smtpd_use_tls = yes
smtpd_tls_auth_only = yes
```

En estas líneas podemos destacar el dominio que como ya hemos dicho será garciapozoit.com, además del tamaño del buzón que es un requisito el que estén limitados a 3 MB como podemos ver en la línea mailbox_size_limit = 3145728 que es el equivalente de 3 MB a bytes. También destacar el tamaño del mensaje que ha sido reducido a 50000 bytes debido a que por defecto el tamaño de este podía ser superior al tamaño del buzón.

Por último comentar que en estos ficheros ya se tiene en cuenta a dovecot y que podemos ver los ficheros que van a ser usados para el acceso a la base de datos. Estos son mysql-virtual-mailbox-maps.cf, mysql-virtual-mailbox-maps.cf y mysql-virtual-alias-maps.cf haciendo referencia a las tres tablas que hemos creado ya en MariaDB.

Estos ficheros tienen la siguiente forma:



```
Terminal x jorge@garciapozoit: ~ x jorge@garciapozoit: ~
GNU nano 5.4 mysql-virtual-mailbox-maps.cf
user = admin
password = labii
hosts = 127.0.0.1
dbname = DBgarciapozoit
query = select 1 from virtual_users where email= '%s'
```

Y accederán a la base de datos como el usuario administrador para obtener los datos almacenados.

Ahora que ya tenemos Postfix configurada correctamente podremos reiniciarla y comprobar que el servicio está activo correctamente:

```

root@garciapozoit:/etc/postfix# systemctl restart postfix
root@garciapozoit:/etc/postfix# systemctl status postfix
● postfix.service - Postfix Mail Transport Agent
   Loaded: loaded (/lib/systemd/system/postfix.service; enabled; vendor preset: enabled)
   Active: active (exited) since Sun 2023-05-21 23:52:59 CEST; 4s ago
   Process: 27976 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 27976 (code=exited, status=0/SUCCESS)
   CPU: 17ms

May 21 23:52:59 garciapozoit.garciapozoit.com systemd[1]: Starting Postfix Mail Transport Agent...
May 21 23:52:59 garciapozoit.garciapozoit.com systemd[1]: Finished Postfix Mail Transport Agent.

```

4.2.4 Dovecot

Dovecot es el servidor de correo entrante que vamos a utilizar para poner en funcionamiento el servidor de correo junto a Postfix. Este es un servidor de código abierto de IMAP y POP3 que permitirá a los usuarios recibir el correo electrónico, almacenarlo y además acceder desde una aplicación de correo electrónico como puede ser Mozilla Thunderbird para utilizarlo de manera más sencilla.

La instalación de los paquetes necesarios ya está incluida en los paquetes instalados de Postfix. Además Dovecot también utilizará la base de datos de MariaDB.

Comenzaremos por tanto con la configuración que se ha realizado, partiendo del directorio `/etc/dovecot`:

En primer lugar modificaremos el fichero `dovecot.conf`, que es el fichero principal de configuración de esta herramienta:

```

#protocolos
protocols = imap pop3 lmtp

```

Con esta línea estableceremos los protocolos que utiliza dovecot para su correcto funcionamiento, que son imap, pop3 y lmtp. Continuaremos editando el archivo `/conf.d/10-mail.conf`:

```

#mail_location = mbox:~/mail:INBOX=/var/mail/%u
mail_location = maildir:/home/vmail/%d/%n/

```

Esta orden nos permitira encontrar donde realizar la búsqueda de los correos que en este caso estarán en `/home/vmail/garciapozoit.com/usuario`. Para que encuentre esta ruta, obviamente tendremos que crearla:

```

root@garciapozoit:/etc/dovecot/conf.d# groupadd -g 5000 vmail
root@garciapozoit:/etc/dovecot/conf.d# useradd -g vmail -u 5000 -d /home/vmail
root@garciapozoit:/etc/dovecot/conf.d# mkdir -p /home/vmail/garciapozoit.com

```

Continuaremos con el archivo de autenticación de usuario para edición, que prohibirá a usuarios que nos estén autenticados de leer correos en el servidor. Esto se realizará en el archivo /conf.d/10-auth.conf:

```

#incluimos
disable_plaintext_auth = yes

!include auth-sql.conf.ext

```

Mediante la segunda orden habilitamos la autenticación por MariaDB de los usuarios.

Continuaremos editando el fichero /conf.d/auth-sql.conf.ext que dirá a Dovecot como encontrar los usuarios autenticados, mediante MariaDB y como encontrar los correos de los usuarios en el servidor.

El siguiente fichero es el que decide el encriptamiento de la contraseña, que es de tipo sha512. Este fichero es el dovecot-sql.conf.ext.

```

#incluimos
password_query= SELECT email as user, password FROM virtual_users WHERE email='%u';

```

Y con esta consulta a la base de datos podremos confirmar si la contraseña es correcta.

En el fichero conf.d/10-master.conf tendremos que realizar varios cambios:

En primer lugar configurar los puertos de IMAP y POP3 siendo respectivamente el 143 y el 110:

```

service imap-login {
    inet_listener imap {
        port = 143
    }

    service pop3-login {
        inet_listener pop3 {
            port = 110
        }
    }
}

```

A continuación hacemos el socket LMTP para evitar que postfix tenga que buscarlo:

```
service lmtp {
    unix_listener /var/spool/postfix/private/dovecot-lmtp {
        mode= 0600
        user= postfix
        group= postfix
    }

    #Create inet listener only if you can't use the above UNIX socket
    #inet_listener lmtp{
    #Avoid making LMTP visible for the entire internet
    #address=
    #port= }
}
```

Por último se creará un bloque para que unix escuche y fijaremos al usuario (vmail).

Por último tendremos que editar el fichero /conf.d/10-ssl.conf para que Dovecot encuentre el certificado SSL.

Y mediante el reinicio de este servicio ya estará el servidor de correo completo en funcionamiento:

```
root@garciapozoit:/etc/dovecot/conf.d# systemctl restart dovecot
root@garciapozoit:/etc/dovecot/conf.d# systemctl status dovecot
● dovecot.service - Dovecot IMAP/POP3 email server
   Loaded: loaded (/lib/systemd/system/dovecot.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-05-22 01:22:30 CEST; 5s ago
     Docs: man:dovecot(1)
           http://wiki2.dovecot.org/
  Main PID: 29760 (dovecot)
    Tasks: 4 (limit: 9473)
   Memory: 3.1M
      CPU: 24ms
   CGroup: /system.slice/dovecot.service
           └─29760 /usr/sbin/dovecot -F
             └─29762 dovecot/anvil
               └─29763 dovecot/log
                 └─29764 dovecot/config

May 22 01:22:30 garciapozoit.garciapozoit.com systemd[1]: Starting Dovecot IMAP/POP3 email server...
May 22 01:22:30 garciapozoit.garciapozoit.com dovecot[29760]: master: Dovecot v2.3.13 (89f716dc2) starting up
```

4.2.5 SSH (SFTP) - albergue de páginas web

SFTP es un protocolo que permite la transferencia de archivos mediante conexión remota a través de la herramienta/servidor de ssh.

Configuraremos esta herramienta para ser utilizada para el servicio ofrecido de albergue de páginas web, donde los usuarios podrán acceder a una carpeta donde podrán subir sus

ficheros para crear su sitio web dentro de nuestro servidor. De manera que querremos aislar las conexiones de los usuarios del resto de archivos para que solo puedan acceder a esta carpeta.

Este proceso se ha desarrollado de la siguiente manera. En primer lugar descargamos la herramienta de openssh-server que incluye a sftp:

```
root@garciapozoit:/# apt-get install openssh-server
```

Y editaremos el fichero de configuración que se encuentra en la ruta por defecto: /etc/ssh/sshd_config.

```
Subsystem      sftp    internal-sftp -l INFO
```

```
Match User jorgepozo
ChrootDirectory /var/www/GarciaPozoIT/paginasWebUsuarios/webs_jorgepozo
ForceCommand internal-sftp -l INFO
AllowTcpForwarding no
X11Forwarding no
```

Donde la primera línea será utilizada para crear el log de accesos y movimientos que se explicará más adelante. Por ahora podemos fijarnos en el segundo bloque. Necesitaremos un bloque de estos por cada usuario que querramos permitir que acceda a una sección separada, este bloque solo les permitirá acceder a la carpeta de la ruta establecido y no le permitirá acceder al resto de carpetas que no estén dentro de ese directorio. Esto se automatizará cuando un usuario quiera dar de alta este servicio mediante el fichero de perl contratarWeb.pl que será la encargada de modificar el fichero sshd_config para añadir el bloque correspondiente, así como de crear las carpetas del usuario.

La carpeta webs_usuario será la carpeta a la que tenga acceso el usuario mediante sftp, pero dentro de esta tendrá otra donde podrá subir los archivos que desee. Esto es debido a que toda la ruta de la carpeta a la que tenga acceso el usuario tiene que tener únicamente permisos de root, al igual que la propia carpeta a la que accede. Por tanto para que pueda modificar los contenidos el propio usuario se crea dentro de esta carpeta otra donde tenga los permisos necesarios.

Ahora sí, la primera línea del fichero de configuración de sshd_config nos lleva al fichero de la siguiente imagen, que escuchará esperando a que se realicen conexiones mediante sftp para guardarlo en el log:


```

module(load="imuxsock");
input(type="imuxsock" Socket="/var/www/GarciaPozoIT/paginasWebUsuarios/webs_jorgepozo/dev/log" CreatePath="on")

if $programname == 'internal-sftp' then /var/log/sftp.log
& stop

```

Por último respecto a SFTP, queremos llevar un registro de los accesos y los movimientos que se han realizado mediante este protocolo, por tanto vamos a configurar un fichero de log mediante el uso de Rsyslog. Este será sftp.log y almacenaremos aquí los accesos, movimientos y desconexiones de usuarios:

```

May 21 01:57:40 garciapozoit internal-sftp[10692]: session opened for local user julia from [192.168.1.68]
May 21 01:57:43 garciapozoit internal-sftp[10692]: session closed for local user julia from [192.168.1.68]
May 21 02:23:20 garciapozoit internal-sftp[11511]: session opened for local user jorpoza from [192.168.1.68]
May 21 02:23:22 garciapozoit internal-sftp[11511]: session closed for local user jorpoza from [192.168.1.68]
May 21 02:25:37 garciapozoit internal-sftp[11780]: session opened for local user jorpoza from [192.168.1.68]
May 21 02:25:40 garciapozoit internal-sftp[11780]: session closed for local user jorpoza from [192.168.1.68]
May 21 02:26:50 garciapozoit internal-sftp[11940]: session opened for local user julia2 from [192.168.1.68]
May 21 02:26:52 garciapozoit internal-sftp[11940]: session closed for local user julia2 from [192.168.1.68]
May 21 12:31:23 garciapozoit internal-sftp[18748]: session opened for local user juliagarciavega from [192.168.1.68]
May 21 12:31:26 garciapozoit internal-sftp[18748]: opendir "/"
May 21 12:31:26 garciapozoit internal-sftp[18748]: closedir "/"
May 21 12:31:37 garciapozoit internal-sftp[18748]: opendir "/meter_aqui_tus_paginas_webs"
May 21 12:31:37 garciapozoit internal-sftp[18748]: closedir "/meter_aqui_tus_paginas_webs"
May 21 12:32:56 garciapozoit internal-sftp[18748]: open "/meter_aqui_tus_paginas_webs/./holaMundo.html" flags WRITE,CREATE,TRUNCATE mode 0666
May 21 12:32:56 garciapozoit internal-sftp[18748]: close "/meter_aqui_tus_paginas_webs/./holaMundo.html" bytes read 0 written 306
May 21 12:32:59 garciapozoit internal-sftp[18748]: opendir "/meter_aqui_tus_paginas_webs"
May 21 12:32:59 garciapozoit internal-sftp[18748]: closedir "/meter_aqui_tus_paginas_webs"

```

4.2.6 Thunderbird

La herramienta de Thunderbird será utilizada para el acceso mediante la cuenta de correo electrónico del dominio garciapozoit.com y su contraseña, que se han generado automáticamente a partir de las credenciales propias del usuario en cuestión al solicitar el alta en el servicio de correo electrónico.

Esta herramienta es fácil de utilizar y de configurar. Simplemente la descargamos de la página web oficial y la descomprimiremos. Así ya el usuario podrá acceder mediante sus credenciales a su cuenta de correo electrónico.

Nombre completo
ejemplo4 ⓘ

Dirección de correo electrónico
email4@garciapozoit.com ⓘ

Contraseña
●●●●●●●● ⓘ

☒ Recordar contraseña

✓ Configuración encontrada probando nombres de servidor comunes.

Configuraciones disponibles

☒ **IMAP**
Mantener sus carpetas y correos electrónicos sincronizados en su servidor

✉ **Entrante** **IMAP** **STARTTLS**
imap.garciapozoit.com

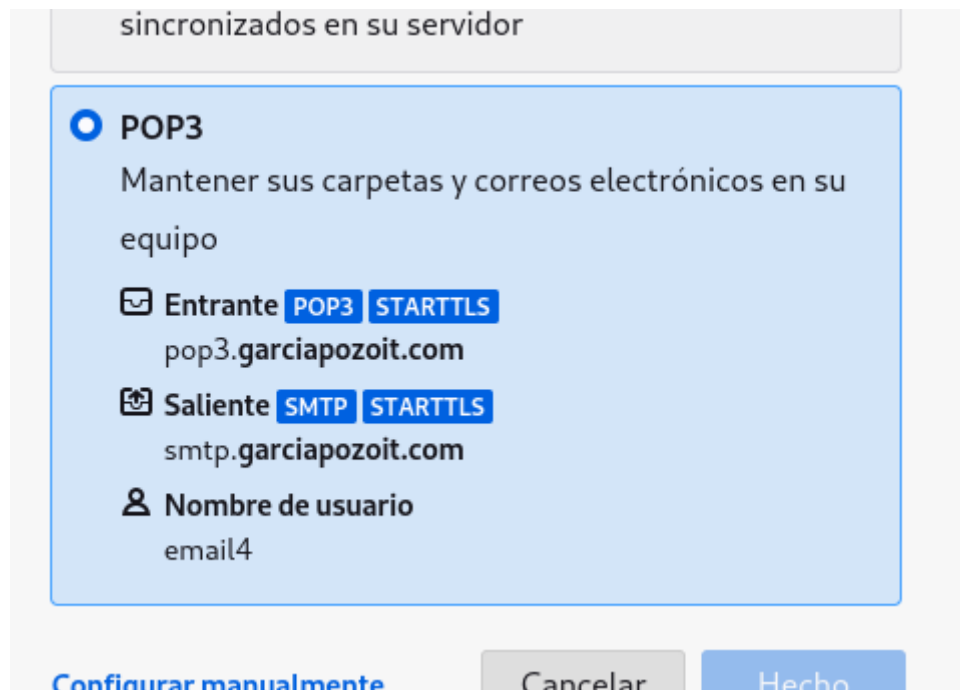
✉ **Saliente** **SMTP** **STARTTLS**
smtp.garciapozoit.com

👤 **Nombre de usuario**
email4

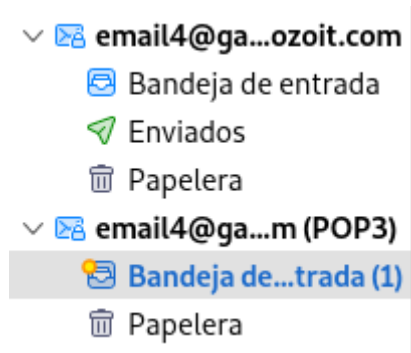
☐ **POP3**
Mantener sus carpetas y correos electrónicos en su equipo

[Configurar manualmente](#)

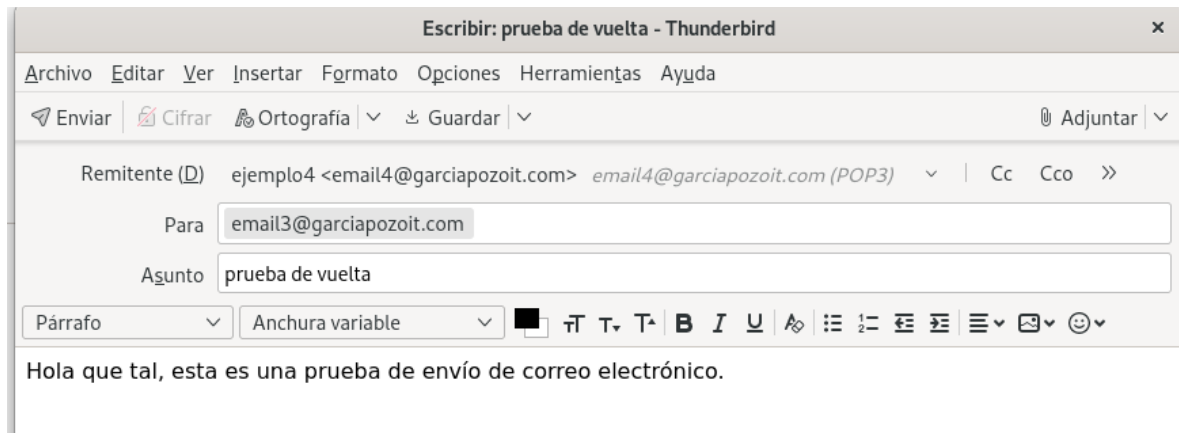
Al introducir las credenciales veremos que detecta automáticamente el servidor de correo electrónico. Aquí seleccionamos POP3 para acceder como un cliente POP (aunque no existen problemas para acceder de igual manera con IMAP) y se comprobará la contraseña introducida.



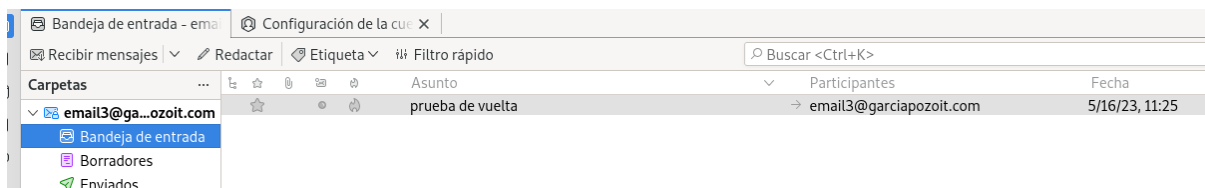
Una vez estemos dentro podemos ver la siguiente interfaz:



Donde podremos ver nuestra bandeja de entrada, ... Y también podremos enviar correos de manera local a los usuarios de este servidor de correo:



Y podemos observar que en el destinatario recibiremos el correo:



4.2.7 Nagios

Utilizaremos la herramienta de Nagios para permitir a los administradores de los servidores configurados poder llevar a cabo la monitorización. Esto incluirá un servicio de recolección de la información de los diferentes componentes de nuestro sistema, donde además de las información recolectada por defecto por Nagios recogeremos información sobre la carga de CPU, el uso de memoria y el uso del disco. De manera que con esta herramienta permitiremos un acceso a esta información y poder evaluar los diferentes componentes de manera más gráfica y rápida.

Comencemos sin más con la instalación de nagios:

Como ya hemos comentado tantas veces comprobaremos si los paquetes del sistema están actualizados con la orden `apt-get update` y en caso de que sea necesario los actualizaremos con la orden `apt-get upgrade`.

A continuación instalaremos las dependencias que requiere Nagios que son básicamente wget y zip (unzip) que utilizaremos para la descarga directamente desde la página web oficial y poder descomprimirlo y otras varias librerías que requiere Nagios:

```
root@garciapozoit:/# apt install wget unzip zip autoconf gcc libc6 make libgd-dev
```

Ahora ya podremos descargar la propia herramienta directamente desde la página web oficial mediante la herramienta que acabamos de descargar de wget:

```
root@garciapozoit:/# wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz
```

Esta se descargará en el directorio en el que estamos trabajando cuando introducimos el comando en un formato zip. El contenido descargado tendremos que descomprimirlo mediante:

```
root@garciapozoit:/# tar -xvzf nagios-4.4.6.tar.gz
```

Una vez lo tengamos descomprimido tendremos que acceder a la propia carpeta descomprimida (nagios-4.4.6). Y dentro de aquí tendremos que configurar el uso de este junto con apache mediante el uso del siguiente comando, de manera que Nagios sepa donde se encuentran los ficheros de configuración de apache2:

```
root@garciapozoit:/# ./configure --with-httpd-conf=/etc/apache2/sites-enabled
```

A continuación tendremos que crear un usuario propio para nagios y meterlo en el grupo de usuarios de www-data que es el grupo de usuarios que tienen permisos con los ficheros de apache2.

Y ahora construiremos los ficheros, ficheros binarios, scripts CGI y ficheros de HTML de Nagios de forma automática con los siguientes comandos:

```
root@garciapozoit:/# make all
```

```
root@garciapozoit:/# make install
```

Por último tendremos que instalar el proceso daemon y los ficheros de configuración de comandos externos mediante las siguientes órdenes:

```
root@garciapozoit:/# make install-init
```

```
root@garciapozoit:/# make install-commandmode
```

Y ya podremos activar el propio servicio con la orden systemctl:

```
root@garciapozoit:/# systemctl enable nagios
```

Y comprobamos que está activo:

```
root@garciapozoit:/# systemctl status nagios
● nagios.service - Nagios Core 4.4.6
   Loaded: loaded (/lib/systemd/system/nagios.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2023-05-21 03:00:25 CEST; 17h ago
     Docs: https://www.nagios.org/documentation
   Main PID: 11605 (nagios)
    Tasks: 23 (limit: 9473)
   Memory: 113.4M
      CPU: 31.368s
   CGroup: /system.slice/nagios.service
           └─11605 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           └─11607 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─11608 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─11609 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─11610 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─11611 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─11612 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─11613 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
```

Vemos que efectivamente se está ejecutando correctamente.

Por último tenemos que crear la cuenta del administrador para esta herramienta, en nuestro caso lo llamamos juliagarve y tenemos que asignarle una contraseña mediante el siguiente comando:

```
root@garciapozoit:/# htpasswd -c /usr/local/nagios/etc/htpasswd.users juliagarve
```

A continuación nos pedirá la contraseña y ya podremos acceder a la página principal de nagios en nuestro servidor de apache. Aquí tendremos que acceder con el usuario y contraseña recientemente creados para Nagios del administrador.

← → ↻ No es seguro | 192.168.1.102/nagios/

Nagios®

General

- Home
- Documentation

Current Status

- Tactical Overview
- Map (Legacy)
- Hosts
- Services
- Host Groups
- Summary
- Grid
- Service Groups
- Summary
- Grid

Problems

- Services (Unhandled)
- Hosts (Unhandled)
- Network Outages

Quick Search:

Reports

- Availability
- Trends (Legacy)
- Alerts
- History
- Summary
- Histogram (Legacy)
- Notifications
- Event Log

System

- Comments
- Downtime
- Process Info
- Performance Info
- Scheduling Queue
- Configuration

Nagios® Core™

✓ Daemon running with PID 11605

Nagios® Core™
Version 4.4.6
April 28, 2020
[Check for updates](#)

Nagios XI
Easy Configuration
Advanced Reporting
[Download](#)

Nagios Log Server
Monitor and analyze
logs from anywhere
[Download](#)

Nagios Network Analyzer
Real-time netflow and
bandwidth analysis
[Download](#)

Get Started

- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of add-ons
- Get support
- Get training
- Get certified

Quick Links

- Nagios Library (tutorials and docs)
- Nagios Labs (development blog)
- Nagios Exchange (plugins and add-ons)
- Nagios Support (tech support)
- Nagios.com (company)
- Nagios.org (project)

Get sta...
NAGIOS CORE XI 5 AVAILABLE NOW!
ACT TOUR

Latest News

- Nagios Update: XI 5.6.6
- Nagios Update: XI 5.6.5
- Nagios Update: XI 5.6.4
- More news...

Don't Miss...

- Monitoring Log Data with Nagios - Nagios Log Server can handle all log data in one central location.
- Can Nagios monitor netflow? - Yes! Nagios Network Analyzer can take in a variety of flow data. [Learn More](#)
- Nagios XI 5 Available Now! - Easier configuration, Advanced Reporting. [Download Today!](#)

Copyright © 2010-2020 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Nagios Core is licensed under the GNU General Public License and is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Nagios, Nagios Core and the Nagios logo are trademarks, servicemarks, registered trademarks or registered servicemarks owned by Nagios Enterprises, LLC. Use of the Nagios marks is governed by the trademark use restrictions.

SOURCEFORGE.NET

Aquí vemos en el menú horizontal del lado izquierdo las diferentes posibilidades que tenemos:

Nagios®

General

Home
Documentation

Current Status

Tactical Overview
Map (Legacy)
Hosts
Services
Host Groups
Summary
Grid
Service Groups
Summary
Grid
Problems
Services (Unhandled)
Hosts (Unhandled)
Network Outages

Quick Search:

Reports

Availability
Trends (Legacy)
Alerts
History
Summary
Histogram (Legacy)
Notifications
Event Log

System

Comments
Downtime
Process Info
Performance Info
Scheduling Queue
Configuration

Donde podemos ver el estado de los servicios, los hosts, las notificaciones que ha realizado Nagios y otras muchas secciones de información.

Podemos ver la parte que más nos interesa que es la de servicios:

Current Network Status

Last Updated: Sun May 21 20:24:26 CEST 2023
Updated every 30 seconds
Nagios® Core™ 4.4.6 - www.nagios.org
Logged in as admin

[View History For all hosts](#)
[View Notifications For All Hosts](#)
[View Host Status Detail For All Hosts](#)

Host Status Totals

| | | | |
|--------------|------|-------------|---------|
| Up | Down | Unreachable | Pending |
| 11 | 0 | 0 | 0 |
| All Problems | | All Types | |
| 0 | | 1 | |

Service Status Totals

| | | | | |
|--------------|---------|-----------|----------|---------|
| Ok | Warning | Unknown | Critical | Pending |
| 11 | 0 | 0 | 0 | 0 |
| All Problems | | All Types | | |
| 0 | | 11 | | |

Service Status Details For All Hosts

Limit Results: 100

| Host | Service | Status | Last Check | Duration | Attempt | Status Information |
|-----------|-----------------|--------|---------------------|----------------|---------|--|
| localhost | Current Load | OK | 05-21-2023 20:19:58 | 0d 15h 27m 37s | 1/4 | OK - load average: 0.17, 0.08, 0.02 |
| | Current Users | OK | 05-21-2023 20:19:58 | 0d 15h 50m 4s | 1/4 | USERS OK - 1 users currently logged in |
| | HTTP | OK | 05-21-2023 20:19:58 | 0d 15h 49m 28s | 1/4 | HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time |
| | PING | OK | 05-21-2023 20:19:58 | 0d 5h 27m 48s | 1/4 | PING OK - Packet loss = 0%, RTA = 0.04 ms |
| | Root Partition | OK | 05-21-2023 20:19:58 | 0d 5h 41m 13s | 1/4 | DISK OK - free space: / 6704 MB (50% inode=77%): |
| | SSH | OK | 05-21-2023 20:24:23 | 0d 0h 0m 5s | 1/4 | SSH OK - OpenSSH_8.4p1 Debian-5+deb11u1 (protocol 2.0) |
| | Swap Usage | OK | 05-21-2023 20:19:58 | 0d 15h 51m 57s | 1/4 | SWAP OK - 100% free (974 MB out of 974 MB) |
| | Total Processes | OK | 05-21-2023 20:19:58 | 0d 15h 51m 20s | 1/4 | PROCS OK: 118 processes with STATE = RSZDT |

Donde podemos ver el estado de los servicios, que se comprobará cada cierto intervalo de tiempo y saliendo avisos en caso de que haya habido algún problema o que el servicio en cuestión se encuentre en estado crítico.

Además de todo lo que permite monitorear Nagios se va a añadir la recopilación de datos del uso de memoria, de cpu y de disco.

Para lograr esto tendremos que modificar los archivos de configuración de nagios y utilizar diferentes librerías/plugins de código abierto. Necesitamos una librería por cada nuevo componente de nuestro sistema que vamos a modificar. Las librerías utilizadas son: check_cpu_info, check_disk y check_mem que podemos descargarlas de páginas web creadas por los usuarios de esta herramienta y que se dejarán los enlaces en el apartado de referencias.

De manera que una vez tengamos las librerías especificadas tendremos que añadir los servicios y comandos que necesitará Nagios para usarlas. Se mostrará por ejemplo que hay que modificar para el uso de la primera de las librerías: check_cpu_info:

En primer lugar editaremos el fichero /usr/local/nagios/etc/objects/commands.cfg añadiendo las siguientes líneas:


```
define command {
    command_name check_cpu_info
    command_line $USER1$/check_cpu_info -w $ARG1$ -c $ARG2$
}
```

Estas líneas definen el comando que se usa para comprobar el estado de la cpu y será utilizado en el siguiente fichero a editar que es localhost.cfg:

```
define service {
    use      local-service
    host_name      localhost
    service_description      Uso de CPU
    check_command  check_cpu_info!80!90
    notifications_enabled    1
}
```

Donde definimos el servicio en sí y utilizamos el comando anteriormente mencionado con los parámetros que deseamos y que se explicarán a continuación. De la misma manera configuraríamos el resto de componentes que queremos monitorizar únicamente modificando los parámetros para la ejecución de sus librerías propias.

Parámetros para las implementaciones de monitorización:

- `check_cpu_info`: necesitará de dos parámetros. El primero será utilizado para saber a qué valor se pondrá el “warning” de uso de la cpu y el segundo contendrá el valor de “critical”. Esto significa que cuando el uso de cpu supere estos valores Nagios detectará que hay problemas y en caso de tener las notificaciones activadas notificará de la manera que haya sido especificada.
- `check_mem`: este necesitará de cuatro parámetros que son la memoria y el intercambio de memoria para dar “warning” y la memoria y el intercambio de memoria para dar “critical”.
- `check_disk`: que nos avisará del estado de los discos necesitando de 3 parámetros que indican cuando se da un estado crítico en este componente.

De esta manera reiniciando el servicio de Nagios ya empezaremos a monitorear todos estos componentes:

| | | | | | |
|----------------|----|---------------------|----------------|-----|--|
| Uso de CPU | OK | 05-21-2023 20:49:04 | 0d 0h 18m 36s | 1/4 | [CPU_INFO: 10%] User: 0.00% - System: 0.10% - Idle: 99.90% - IOWait: 0.00% |
| Uso de Disco | OK | 05-21-2023 20:49:58 | 0d 10h 6m 39s | 1/4 | DISK OK - free space: / 6700 MB (50% inode=77%); |
| Uso de Memoria | OK | 05-21-2023 20:49:58 | 0d 10h 33m 54s | 1/4 | OK: Memory Usage (W> 90, C> 95): 34% Swap Usage (W> 100, C> 100): 30% |

Por último respecto a Nagios comentar que permite la notificación en caso de caída o problema con los componentes monitorizados mediante correo electrónico tanto local como con servidores más amplios como puede ser Gmail. Esta funcionalidad no ha sido implementada debido a los errores que daban los certificados SMTP al ser autorregulados respecto al servidor propio implementado. Es por esto que como se explicará más adelante la notificación por correo a los administradores respecto a estas estadísticas e información se ha realizado por otro método.

4.2.8 Wordpress

Wordpress es una herramienta que nos permitirá ofrecer a los usuarios un sistema de blogs donde podrán escribir sus entradas, comentar la del resto de usuarios y mantener sus blogs actualizados todo esto en nuestro propio servidor de apache2 y mediante el uso de las tablas de MariaDB. Es de código abierto y es una de las herramientas más utilizadas para este fin.

Esta herramienta también utiliza la base de datos que hemos creado de MariaDB donde almacenará la diferente información de los usuarios así como otras informaciones relevantes. Además la configuración es muy sencilla, intuitiva y prácticamente automática ofreciendo unas funcionalidades de igual forma eficientes.

Se comenzará sin más dilación con la instalación y configuración de Wordpress:

Como para todas las herramientas vistas comenzaremos instalando las dependencias:

```
root@garciapozoit:/etc/postfix# apt-get install php7.4 php7.4-bcmath php7.4-curl php7.4-gd php7.4-mbstring php7.4-mysql php7.4-pgsql php7.4-xml php7.4-zip curl
```

Como vemos la mayoría hacen referencia a php puesto que Wordpress hace uso de este lenguaje para manejar sus páginas. También tendremos que descargar la propia herramienta de Wordpress que descargaremos directamente la última versión de la página oficial de esta (la dirección url se dejará en referencias):

Impagable y, al mismo tiempo, gratuito

Descarga WordPress y úsalo en tu sitio.

 Descargar WordPress 6.2.2

[Descargar .tar.gz](#)

Una vez descargado, lo descomprimos y lo movemos a la carpeta donde tenemos el resto de nuestras paginas web, en nuestro caso `/var/www/garciapozoit`, esto nos permitirá el acceso a las propias páginas web de wordpress desde nuestro sitio web.

Cuando comencemos tendremos que crear la cuenta del administrador, como ya hemos hecho en otras herramientas, además de especificar el nombre de la base de datos que va a utilizar.



A continuación debes introducir los detalles de conexión de tu base de datos. Si no estás seguro de esta información contacta con tu proveedor de alojamiento web.

| | | |
|------------------------------|--|---|
| Nombre de la base de datos | <input type="text" value="wordpress"/> | El nombre de la base de datos que quieres usar con WordPress. |
| Nombre de usuario | <input type="text" value="wp_user"/> | El nombre de usuario de tu base de datos. |
| Contraseña | <input type="text" value="supercontraseña"/> | La contraseña de tu base de datos. |
| Servidor de la base de datos | <input type="text" value="localhost"/> | Deberías recibir esta información de tu proveedor de alojamiento web, si localhost no funciona. |
| Prefijo de tabla | <input type="text" value="eb"/> | Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto. |

Una vez se haya realizado esto ya se crearán las tablas correspondientes a wordpress en nuestra base de datos de MariaDB.

```
| wp_commentmeta
| wp_comments
| wp_links
| wp_options
| wp_postmeta
| wp_posts
| wp_term_relationships
| wp_term_taxonomy
| wp_termmeta
| wp_terms
| wp_usermeta
| wp_users
```

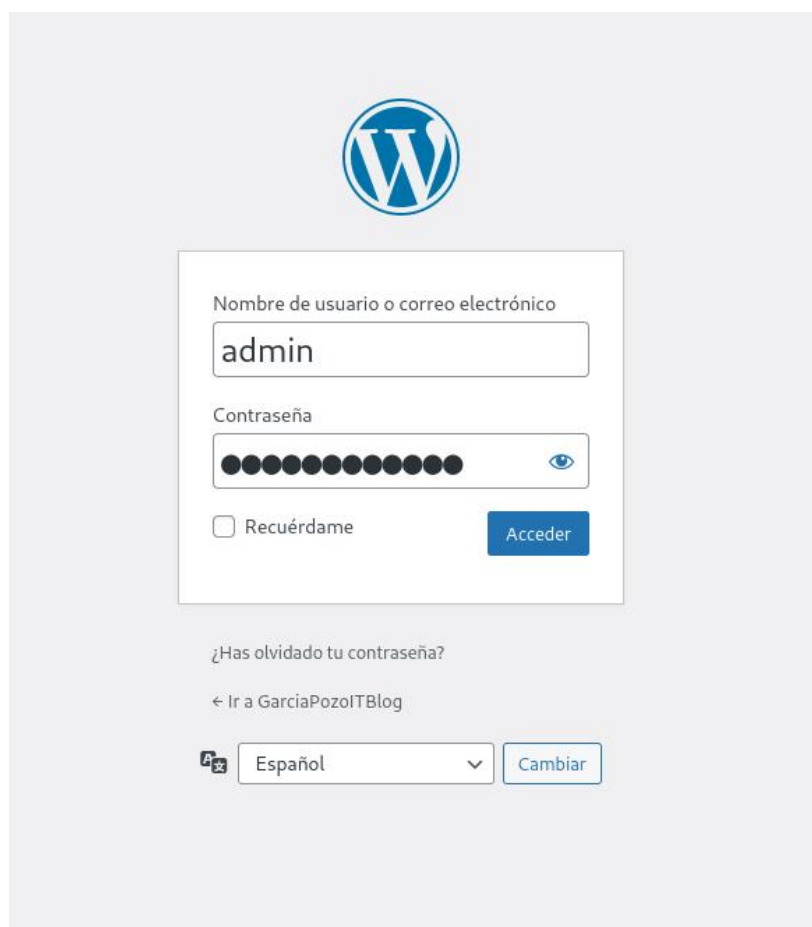
De estas tablas podemos destacar wp_users, que es donde se almacena la información de los usuarios:

```
MariaDB [DBgarciapozoit]> select * from wp_users;
+-----+-----+-----+-----+-----+-----+
| ID | user_login | user_pass | user_nicename | user_email | user_url |
| user_registered | user_activation_key | user_status | display_name |
+-----+-----+-----+-----+-----+-----+
| 1 | admin | $P$B59.dYY.vJZnsRyJLbUnK5QeL7L/d0 | admin | soporte@garciapozoit.com | http://www.garciapozoit.com/wordpress |
| 2023-05-16 10:44:12 | 0 | admin |
| 2 | ejemplo3 | $P$BpL70ypnI1SQYg8jUYEjzoLxkBaXL71 | ejemplo3 | ejemplo3@garciapozoit.com | http://ejemploBlog |
| 2023-05-16 11:23:17 | 1684236200:$P$BCvZtcYxkDCKhB.Ttn8UEEAeocYx3s/ | 0 | Ejemplo 3 |
| 3 | ejemplo4 | $P$BCZQb064pKoVjBSrLmDGoENGdE6LTv1 | ejemplo-4 | ejemplo4@garciapozoit.com |
| 0000-00-00 00:00:00 | 0 | Ejemplo 4 |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)
```

También podemos destacar la tabla de wp_usermeta, que es donde se almacena la información de los permisos de los distintos usuarios. Todos los usuarios que soliciten este servicio entrarán como autores y además solo tenemos el administrador que tenga un rol distinto:

```
MariaDB [DBgarciapozoit]> select * from wp_usermeta;
+-----+-----+-----+-----+
| umeta_id | user_id | meta_key | meta_value |
+-----+-----+-----+-----+
| 1 | 1 | nickname | admin |
| 2 | 1 | first_name | |
| 3 | 1 | last_name | |
| 4 | 1 | description | |
+-----+-----+-----+-----+
```

En nuestro caso el nombre de la base de datos será la que ya hemos creado y utilizado para otros usos, como guardar los usuarios y para la herramienta de postfix, que es DBgarciapozoit y simplemente introduciremos los datos del admin.

The image shows the WordPress login interface. At the top center is the WordPress logo, a blue 'W' inside a circle. Below it is a white rectangular box containing the login fields. The first field is labeled 'Nombre de usuario o correo electrónico' and contains the text 'admin'. The second field is labeled 'Contraseña' and contains a series of black dots, with an eye icon to its right for toggling visibility. Below these fields is a checkbox labeled 'Recuérdame' and a blue button labeled 'Acceder'. Under the login box, there is a link '¿Has olvidado tu contraseña?' and another link '← Ir a GarciaPozoITBlog'. At the bottom, there is a language selector showing 'Español' with a dropdown arrow and a 'Cambiar' button next to it.

Nombre de usuario o correo electrónico


admin

Contraseña

☐ Recuérdame [Acceder](#)

[¿Has olvidado tu contraseña?](#)

[← Ir a GarciaPozoITBlog](#)

 Español [Cambiar](#)

Una vez esto podremos logear como el admin de manera gráfica y veremos la siguiente interfaz:



Donde podríamos añadir entradas o eliminarlas, crear usuarios, etc. Esto es debido a que en wordpress como se verá más profundamente cuando se explique el perl que permite el dado de alta automático a los usuarios que lo soliciten, tiene diferentes roles, donde podemos destacar el de administrador, que como es común, lo puede hacer todo y el de autor, que puede subir entradas al blog y comentar en las entradas del resto de usuarios.

4.2.9 Tripwire

Para realizar un control de seguridad diario de los ficheros críticos para el buen funcionamiento del sistema, hemos utilizado la herramienta Tripwire.

Antes de instalar la herramienta, actualizamos el sistema con el comando apt update.

A continuación, instalamos Tripwire con el comando apt install Tripwire.

Debido a errores en la generación de la base de datos, añadimos en el fichero twcfg.txt en /etc/tripwire la siguiente línea para que no de problemas el siguiente paso: RESOLVE_IDS_TO_NAMES=false.

A continuación, se actualiza el fichero tw.cfg ya que hemos modificado el fichero twcfg.txt: twadmin --create-cfgfile --site-keyfile site.key twcfg.txt

Por último creamos la base de datos con el comando tripwire -init.

De esta forma ya estaría funcional la herramienta. En nuestro caso realizamos controles con el comando tripwire -check y almacenando su salida en un log de nombre la fecha del día.

4.2.10 Quota

Para gestionar las cuotas de los usuarios utilizamos el paquete quota. Se puede instalar con el comando apt-get install quota. Este sistema de cuotas lo montamos sobre un sistema de ficheros completo que hemos creado. Este sistema de ficheros lo hemos creado con las siguientes sentencias:

```
dd if=/dev/zero of=/home/SFich bs=1M count=10
mkfs.ext3 /home/SFich
mount -o loop /home/SFich /mnt/Pquota
```

Anteriormente hemos creado el directorio Pquota en /mnt.

Una vez hemos creado el sistema de ficheros con el que vamos a trabajar, editamos el fichero /etc/fstab incorporando la siguiente línea para habilitar el sistema de cuotas:

```
/home/SFich    /mnt/Pquota    ext3    loop,usrquota,grpquota    0    0
```

A continuación, reiniciamos el sistema para que se incorporen los cambios con la sentencia: mount -o remount,usrquota,grpquota /home/SFich.

De esta forma ya podríamos hacer un chequeo de cuotas con el comando: quotacheck -auvg.

Por último ejecutamos el comando quotaon -uv /mnt/Pquota para habilitar las cuotas.

Con el comando quotaon -p -a podemos verificar que existen:

```
root@garciapozoit:/etc/tripwire# quotaon -p -a
group quota on /mnt/Pquota (/dev/loop0) is on
user quota on /mnt/Pquota (/dev/loop0) is on
```

4.2.11 Rsync

Para realizar las copias de seguridad solicitadas en el sistema, hemos utilizado la herramienta Rsync. Esta herramienta permite encriptar las transferencias de datos y además supone un ahorro ya que únicamente copia aquellos archivos que han sido modificados.

En nuestro sistema realizamos copias de seguridad de la carpeta /home, de la carpeta /usr/lib/cgi-bin, de la carpeta /var/www/GarciaPozoIT y de la carpeta /var/lib/mysql. De esta forma se guardan los directorios home de los usuarios, los html, los archivos perl y la base de datos.

El comando para realizar la copia de seguridad es el siguiente:

```
rsync -avg /origen /copia
```

4.2.12 Estadísticas: ac y sa

Para llevar a cabo algunas estadísticas del sistema se han utilizado los comandos ac y sa. El comando sa ya viene instalado por defecto pero el comando ac hay que instalarlo con apt-get install acct. Para nuestro sistema obtenemos las estadísticas de los usuarios que más tiempo han estado conectados al sistema, las estadísticas de actividad del sistema para todos los usuarios y las estadísticas de actividad del sistema para el usuario. Estas salidas de los comandos se almacenan luego en ficheros.

```
ac -p > /home/estadisticas/est_Tiempo.log 2>&1
/usr/sbin/sa -a > /home/estadisticas/est_TiempoCPU.log 2>&1
/usr/sbin/sa -u > /home/estadisticas/est_Usuario.log 2>&1
```

4.2.13 Crontab

El archivo crontab permite realizar tareas periódicas en unos momentos específicos. Para nuestro sistema, hemos utilizado esta herramienta para:

- Ejecutar cada minuto el script de perl que detecta el registro de nuevos usuarios y les manda el correo electrónico de confirmación.
- Ejecutar todos los días a las 23:55 las copias de seguridad con rsync.
- Ejecutar todos los días a las 20:00 el control de ficheros con tripwire.
- Ejecutar todos los días a las 19:15 las estadísticas.

```
*/1 * * * * cd /usr/lib/cgi-bin; perl comprobarSolicitudesAlta.pl
55 23 * * * rsync -avg /home /home/copiaDeSeguridad/usuarios
55 23 * * * rsync -avg /var/lib/mysql /home/copiaDeSeguridad/baseDeDatos
55 23 * * * rsync -avg /usr/lib/cgi-bin /home/copiaDeSeguridad/cgi-bin
55 23 * * * rsync -avg /var/www/GarciaPozoIT /home/copiaDeSeguridad/html
00 20 * * * sh /home/informesTripwire/ejecutarTripwire.sh
15 19 * * * sh /home/estadisticas/estadisticas.sh
```

4.2.14 Certificado https

Para que la conexión sea segura a nuestro servidor al conectarnos por un navegador web hemos instalado un certificado https para nuestro servidor.

Para generar el certificado SSL gratis hemos utilizado la herramienta openssl. Para instalarla se ejecuta el comando apt install openssl y se habilita los módulos ssl y rewrite con las sentencias: a2enmod ssl a2enmod rewrite.

A continuación, se modifica el fichero de apache2.conf añadiendo las siguientes líneas:


```
<Directory /var/www/GarciaPozoIT>
    AllowOverride ALL
</Directory>
```

Para generar el certificado basta con ejecutar el siguiente comando: `openssl req -new -newkey rsa:4096 -x509 -sha256 -days 365 -nodes -out apache.crt -keyout apache.key`.

Al ejecutar el comando se le solicita información que no hace falta especificar pero sí es importante indicar bien el nombre del servidor: en nuestro caso la ip del servidor.

Por último se debe modificar el fichero `/etc/apache2/sites-enabled/GarciaPozoIT.com.conf` para que quede de la siguiente manera:

```
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # 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) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    # ServerName www.garciapozoit.com

    ServerName 192.168.1.89

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/GarciaPozoIT

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    RewriteEngine on

    RewriteCond %{HTTPS} !=on

    RewriteRule ^/?(.*) https://%{SERVER_NAME}/$1 [R=301,L]

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

```

<VirtualHost *:443>

    #ServerName www.garciapozoit.com
    ServerName 192.168.1.89

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/GarciaPozoIT

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    SSLEngine on

    SSLCertificateFile /etc/apache2/certs/apache.crt

    SSLCertificateKeyFile /etc/apache2/certs/apache.key

</VirtualHost>

```

5. Ficheros fuente PERL

En esta sección se explicará de manera general el funcionamiento de los distintos archivos perl en los que se basa el sistema. Únicamente se destacarán los aspectos más relevantes.

Primero, destacamos a continuación algunos aspectos que se repiten en la mayoría de los programas:

Acceder a la base de datos

```

my $database = 'DBGarciapozoit';
my $host = 'localhost';
my $username = 'admin';
my $password = 'labii';

my $dbh = DBI->connect("DBI:MariaDB:database=$database;host=$host",$username,$password,{RaiseError=>1,PrintError=>0});

my $sentencial = $dbh->prepare('select nombreUsuario from usuario where nombreUsuario=?');
$sentencial->execute($usuario);
if($sentencial->fetchrow()){
    print "Location: http://192.168.1.89/registroErroneo.html\n\n";
    $dbh->disconnect();
    exit(0);
}

```

Se utiliza el módulo DBI para conectarse a la base de datos en mariadb en la que mantenemos nuestro almacenamiento. Para conectarse a la base de datos hay que especificar el nombre de la base de datos, el host, el usuario con el que acceder y su contraseña. Con la instancia a la conexión de la base de datos con DBI para ejecutar una sentencia, únicamente sería indicarla en el método prepare sustituyendo las variables por interrogantes, luego ejecutarla con el método execute listando las variables con la que ejecutar la sentencia y por último, si es necesario, obtener las filas resultantes de la sentencia. Al final de las consultas hay que desconectarse con el método disconnect. (IBM, 2019)

Obtener los parámetros de un formulario en html

```
my $cgi= new CGI;

my $nombre = decode('utf8', $cgi -> param("nombre"));
my $apellidos = decode('utf8', $cgi -> param("apellidos"));
my $direccion = decode('utf8', $cgi -> param("direccion"));
my $correo = decode('utf8', $cgi -> param("correo"));
my $usuario = decode('utf8', $cgi -> param("usuario"));
my $contrasena = decode('utf8', $cgi -> param("contrasena"));
my $contrasena2 = decode('utf8', $cgi -> param("contrasena2"));
```

Para obtener los campos introducidos en un formulario en html se utiliza el módulo CGI de CPAN (Johnson, n.d.). Con este módulo gestionamos las peticiones post de un html al querer mandar el formulario. Para obtener el argumento deseado se debe especificar dentro de *param* el valor de la etiqueta *name* del html.

Sesiones en perl con CGI::Session():

Al iniciar sesión se registra una nueva sesión con el usuario como parámetro para que pese a cambiar de páginas y de scripts se siga manteniendo la sesión de ese usuario.

Al crearla le damos como parámetro el nombre de usuario que ha iniciado sesión y también le indicamos que expire en una hora. (Ruzmetov & Stosberg, n.d.)

```
my $sesion = CGI::Session->new();
$sesion->param('usuario', $usuario);
$sesion->expires("+1h");
$sesion->flush();
$dbh->disconnect();
```

En las siguientes pantallas a las que accede el usuario se recupera la sesión. Se comprueba si la sesión no tiene algún argumento y si es así es que no se ha iniciado sesión entonces no se puede acceder a las pantallas del usuario y se redirige a la pantalla de inicio de sesión. Se comprueba también si la sesión ha expirado y si es así se redirige también a la pantalla de inicio de sesión.

```
my $sesion = new CGI::Session;

$sesion->load();

if($sesion->is_expired){
    $sesion->delete();
    $sesion->flush();
    print "Location: http://192.168.1.89/inicioSesion.html\n\n";
    exit(0);
}

my $usuario = decode('utf8', $sesion -> param("usuario"));

unless($usuario){
    $sesion->delete();
    $sesion->flush();
    print "Location: http://192.168.1.89/inicioSesion.html\n\n";
    exit(0);
}
```

Uso del módulo Template

Para personalizar un html según el usuario que sea utilizamos el módulo Template. Esto permite introducir variables en el html y sustituir estas variables en el script de perl. En el html se escribe la variable con el siguiente formato [% var1 %], y en el script de perl se da valor a esa variable de la siguiente manera \$insertar->{var1}=valor. Finalmente se carga la plantilla con esos valores dados y se recoge el html resultante en otra variable.

```
my $template = Template->new({ENCODING => 'utf8',});
my $var;
$var->{nombre_usuario}=$usuario;
$var->{clave}=int(rand(900000))+100000;
my $salida;
$template->process('correoConfirmacion.html',$var,\$salida);
```

5.1. registro.pl

Este fichero se ejecuta al seleccionar el botón de registrarse en la página de registro. En este script primero se recogen los distintos campos del formulario con el modulo cgi. A continuación, se comprueba si el nombre de usuario introducido existe en la base de datos. Si existe se redirige al usuario a la página de registro erróneo. Si no existe, se introduce una nueva fila en la tabla de usuarios en la base de datos con los datos del nuevo usuario.

5.2. comprobarSolicitudesAlta.pl

Este fichero comprueba si hay nuevos registros a los que no se le ha mandado un correo de confirmación y les manda el correo. Para ello primero busca en la base de datos aquellos usuarios con un estado de 1. El estado del usuario puede ser 1, 2, 3. 1 es cuando se ha registrado pero no se le ha enviado el correo de confirmación. 2 es cuando se le ha enviado el correo de confirmación pero todavía no lo ha validado. 3 es cuando ha validado su correo electrónico. Por tanto, a aquellos usuarios con estado=1 les manda un correo con un html personalizado incluyendo su usuario y una clave para validar su correo.

Para mandar el correo electrónico primero especificamos con el módulo Email::Sender::Transport::SMTPS el transporte, indicando el host, el puerto, el protocolo ssl, el usuario sals y su contraseña. (林, n.d.)

Para este programa hemos creado una cuenta de correo en gmail que únicamente se encarga de mandar correos de confirmación. Para permitir que este programa utilice esa cuenta se debe habilitar la verificación en dos pasos y generar una contraseña de aplicación. Esta será la que se debe especificar en el transporte. (*Troubleshooting sSMTP: Authorization Failed - Unix & Linux Stack Exchange*, 2020)

A continuación se crea el mensaje a enviar especificando el remitente, destinatario, asunto, el cuerpo y otros parámetros que queramos. Por último se envía el mensaje en el transporte especificado con el módulo Email::Sender::Simple y se actualiza el estado del usuario a 2.

```
my $transport = Email::Sender::Transport::SMTPS->new(
    host => 'smtp.gmail.com',
    port => 587,
    ssl => 'starttls',
    sasl_username => 'registrogarciaipozoit1@gmail.com',
    sasl_password => 'fbxnurnqgxxckzmf',
    debug => 0, # or 1
);

my $message = Email::MIME->create(
    header_str => [
        From => 'registrogarciaipozoit1@gmail.com',
        To => $correo,
        Subject => 'Confirma tu cuenta en GarciaPozo IT',
    ],
    attributes => {
        charset => 'UTF-8',
        encoding => 'quoted-printable',
        content_type => 'text/html',
    },
    body_str => $salida,
);

try {
    sendmail($message, { transport => $transport });
} catch {
    die "Error sending email: $_";
};
```

5.3. confirmarRegistro.pl

Este script se ejecuta para validar la cuenta de un usuario. Es ejecutado al seleccionar en la pantalla de confirmación de correo con la clave al seleccionar el botón de validar cuenta. Primero obtiene la clave registrada en la base de datos y la comprueba con la recibida. Si no coinciden se redirige al usuario a la pantalla de confirmar registro erróneo. Si si coinciden se actualiza el estado del usuario a 3, se crea su directorio home y public_html con los permisos 770. Se añade con el módulo Linux::usermod al nuevo usuario, se convierte en dueño de su home el usuario con chown y se establecen las cuotas límites con el módulo Quotas indicando como límite duro los 80 MB de espacio. Finalmente se redirige al usuario a la pantalla de registro confirmado.

```
Linux::usermod->add($usuario,$contrasena,'',1001,'',$ruta,"/bin/bash");
Quota::setqlim("/dev/loop0",$uid,71680,81920,0,0,0);
```

5.4. inicioSesion.pl

Este fichero inicia la sesión del usuario en el sistema. Obtiene como parámetros del formulario el nombre de usuario y la contraseña. Obtiene de la base de datos la contraseña y el estado del usuario. Por tanto si el estado de ese usuario no es 3 se redirige a la página de inicio de sesión erróneo. Al comparar la contraseña introducida con la del usuario registrada en la base de datos si no coinciden también se redirige al usuario a la página de inicio de sesión erróneo. Lo mismo sucede si el usuario introducido no existe. Si la contraseña coincide y el usuario tiene un estado de 3 se inicia una nueva sesión y se configura como hemos explicado anteriormente. A continuación, se redirige al usuario a la página de servicios.

Los inicios de sesión tanto fallidos como correctos se registran en el fichero `/var/log/inicioWeb.log` junto con su hora.

5.5. serviciosUsuario.pl

Este script muestra los servicios contratados y no contratados del usuario logueado. Primero obtiene las variables correo, blog y web de la tabla usuarios para ver qué servicios tiene contratados. Según estos valores añade en una tabla o en otra los elementos html que corresponden utilizando una plantilla como hemos explicado anteriormente.

5.6. contratarCorreo.pl

Este script crea un nuevo correo para el usuario. Primero actualiza de la base de datos la variable correo para que sea 1, es decir, que está contratado. A continuación encripta la contraseña del usuario e inserta en la tabla de virtual_users los datos correspondientes. Finalmente vuelve a redirigir al usuario a la pantalla de serviciosUsuario para actualizar la pantalla.

5.7. finalizarCorreo.pl

Este script elimina el correo del usuario. Primero actualiza de la base de datos la variable correo para que sea 0, es decir, que no está contratado. A continuación, elimina de la tabla de virtual_users los datos de su correo y elimina también de la carpeta vmail su buzón correspondiente con el módulo `File::Path qw(rmtree)`. Finalmente vuelve a redirigir al usuario a la pantalla de serviciosUsuario para actualizar la pantalla.

5.8. contratarBlog.pl

Este script crea un nuevo correo para el usuario. Primero obtiene los datos necesarios del usuario para la creación del blog. A continuación, encripta la contraseña del usuario e inserta en la tabla de wp_users y wp_usermeta los datos correspondientes. Finalmente actualiza de la base de datos la variable blog para que sea 0 y vuelve a redirigir al usuario a la pantalla de serviciosUsuario para actualizar la pantalla.

5.9. finalizarBlog.pl

Este script elimina el blog del usuario. Actualiza de la base de datos la variable blog para que sea 0. También elimina de las tablas de wp_users y wp_usermeta los datos correspondientes al usuario. Finalmente vuelve a redirigir al usuario a la pantalla de serviciosUsuario para actualizar la pantalla.

5.10. contratarWeb.pl

Este script crea un nuevo espacio de almacenamiento web para el usuario. Primero se hace propietario de la carpeta paginasWebUsuarios en var/www para poder crear directorios y modificar permisos. Crea el directorio de páginas web del usuario y en él un index.html, la carpeta en la que el usuario debe meter las páginas webs y una carpeta dev para los logs de sftp.

Si el usuario no estaba registrado anteriormente añade en el fichero sshd_config unas líneas para restringir el acceso del usuario con sftp a solo su carpeta. Además en el fichero sftp.conf en rsyslog.d añade un nuevo socket para generar el log con la actividad de los usuarios. Esto solo hace falta hacerlo una vez.

A continuación, hace propietario al usuario de estas carpetas y devuelve al root la propiedad de la carpeta paginasWebsUsuario y le hace propietario también de la carpeta webs_\$usuario. Finalmente se reinician los servicios de ssh y rsyslog para que incorporen los cambios.

Actualiza de la base de datos la variable web para que sea 1, es decir, que está contratado. A continuación encripta la contraseña del usuario e inserta en la tabla de virtual_users los datos correspondientes. Finalmente vuelve a redirigir al usuario a la pantalla de serviciosUsuario para actualizar la pantalla.

5.11. finalizarWeb.pl

Este script elimina el correo del usuario. Actualiza de la base de datos la variable webpara que sea 0 y elimina la carpeta de webs del usuario. Finalmente vuelve a redirigir al usuario a la pantalla de serviciosUsuario para actualizar la pantalla.

5.12. ejecutarObtenerEstadoServicios.pl

Este script ejecuta un archivo de bash para obtener el estado de los servicios ya que hay que hacer llamadas al sistema con systemctl. Se ejecuta al seleccionar en el menú la sección de estado servicios.

5.13. obtenerEstadoServicios.sh

Este script obtiene el estado de los servicios postfix, dovecot, ssh, rsyslog, mariadb y apache2 y se lo manda como argumento a un archivo perl.

5.14. estadoSistema.pl

Este script muestra en pantalla el estado de los sistemas. Según el estado de los distintos servicios a partir de una plantilla modifica el color de la fila correspondiente al servicio y su icono.

5.15. verPerfil.pl

Este script muestra en pantalla los datos del usuario. Obtiene de la base de datos sus respectivos datos y personaliza una plantilla ya creada de ver perfil con los datos del usuario imprimiendo posteriormente por pantalla.

5.15. modificarPerfil.pl

Este script muestra en pantalla los datos del usuario pero en campos editables. Realiza lo mismo que el script anterior pero con otra plantilla de html que permite editar los campos.

5.16. guardarModificarPerfil.pl

Este script actualiza los datos del usuario en la base de datos. Se ejecuta al enviar el formulario de modificar perfil. Recoge de este formulario en html los distintos campos con los datos y los actualiza en la base de datos para ese usuario. Para finalizar redirige al usuario a la pantalla de ver perfil.

5.17. eliminarPerfil.pl

Este script borra todo rastro del usuario en el sistema. Primero comprueba que servicios tiene contratos y realiza las acciones respectivas para eliminarlos como hemos explicado en finalizarWeb.pl, finalizarCorreo.pl y finalizarBlog.pl. También elimina de la tabla de usuario su fila correspondiente, elimina su directorio home y el elimina al usuario del sistema con usermod->del.

5.18. contacto.pl

Este script muestra el html con la plantilla de contacto únicamente personalizándolo con el nombre del usuario.

5.19. ayuda.pl

Este script muestra el html con la plantilla de ayuda únicamente personalizándolo con el nombre del usuario.

5.20. cerrarSesion.pl

Este script elimina la sesión del usuario y guarda en el log de inicioWeb.log la salida del usuario del sistema con la fecha. Finalmente se redirige al usuario a la pantalla de inicio de sesión.

6. Conclusión

Concluiremos este informe documentativo de la solución implementada para un servidor de una empresa tecnológica emergente que ofrece diferentes servicios a sus usuarios destacando algunos de los aspectos.

El uso de perl, junto a html y css ha permitido un desarrollo web rápido, haciendo especial hincapié en el gran abanico de posibilidades que abre esta combinación de lenguajes permitiendo: acceso, escritura y recolección de datos de una base de datos, recolección de información de formularios de html, labores de movimiento y creación de ficheros y directorios en el sistema tanto de log como para uso corriente, modificación de ficheros de manera sencilla...

Además se ha presentado solución a todas las necesidades del enunciado, mediante el uso de las herramientas y ficheros descritos a lo largo del informe, destacando postfix y dovecot para la implementación del servidor de correo, wordpress para la creación de blogs y Nagios para la monitorización de los componentes del sistema. Y mediante el uso de MariaDB se ha sido capaz de organizar toda la información relevante y necesaria para los servicios ofrecidos de manera eficaz y segura.

Por último recordar la importancia que tienen los permisos de los directorios y documentos, si no se lleva un control riguroso de estos y por ejemplo se le dan permisos de modificación y ejecución a ficheros o directorios indebidos para usuarios indebidos podría llevar al posible acceso ante un hackeo a determinados lugares de nuestro sistema que no deberían ser accesibles.

7. Referencias

林, F. (n.d.). *Email::Sender::Transport::SMTPS - Email::Sender joins Net::SMTPS -*

metacpan.org. MetaCPAN. Retrieved May 22, 2023, from

<https://metacpan.org/pod/Email::Sender::Transport::SMTPS>

Bootstrap 4 confirm account welcome email template Example. (n.d.). BBBootstrap.

Retrieved May 21, 2023, from <https://bbbootstrap.com/snippets/bootstrap->

[bootstrap-4-confirm-account-email-template-31081089](https://bbbootstrap.com/snippets/bootstrap-4-confirm-account-email-template-31081089)

Bootstrap Registration form - free examples & tutorial. (n.d.). Material Design for Bootstrap. Retrieved May 21, 2023, from

<https://mdbootstrap.com/docs/standard/extended/registration/>

Bootstrap snippet. Confirmation email. (2016, October 24). Bootdey. Retrieved May 21, 2023, from <https://bootdey.com/snippets/view/Confirmation-email#html>

Can you require two form fields to match with HTML? (n.d.). Stack Overflow.

Retrieved May 21, 2023, from <https://stackoverflow.com/questions/9142527/can-you-require-two-form-fields-to-match-with-html>

Cristian Cuesta. (2022, 31 marzo). *Debian 11, servidor de correo con postfix, dovecot y mariadb* [Vídeo]. YouTube. https://www.youtube.com/watch?v=Epzdzpsr_Pg

Directory - Nagios Exchange. (s. f.).

https://exchange.nagios.org/directory/Plugins/Operating-Systems/Linux/check_disk--2D-%25

Equipo Bootstrap. (n.d.). Bootstrap · The most popular HTML, CSS, and JS library in the world. Retrieved May 21, 2023, from <https://getbootstrap.com/>

Font Awesome 5 Introduction. (n.d.). W3Schools. Retrieved May 21, 2023, from https://www.w3schools.com/icons/fontawesome5_intro.asp

Homann, R. (s. f.). check_mem - Nagios Exchange.

https://exchange.nagios.org/directory/Plugins/Operating-Systems/Linux/check_mem/details

How to Install Nagios Open Source Monitoring on Debian 11 - GreenCloud Documentation. (2023, 12 abril). GreenCloud Documentation.

<https://green.cloud/docs/how-to-install-nagios-open-source-monitoring-on-debian-11/>

HTML input pattern Attribute. (n.d.). W3Schools. Retrieved May 21, 2023, from https://www.w3schools.com/tags/att_input_pattern.asp

IBM. (2019, May 7). . IBM. Retrieved May 22, 2023, from <https://www.ibm.com/docs/es/db2/11.5?topic=p-database-connections>

Johnson, L. (n.d.). *CGI - metacpan.org*. MetaCPAN. Retrieved May 22, 2023, from <https://metacpan.org/pod/CGI>

Rai, S. (s. f.). *check_cpu_info - Nagios Exchange*. https://exchange.nagios.org/directory/Plugins/Operating-Systems/check_cpu_info/details

Ruzmetov, S., & Stosberg, M. (n.d.). *CGI::Session::Tutorial - Extended CGI::Session manual*. MetaCPAN. Retrieved May 22, 2023, from <https://metacpan.org/pod/CGI::Session::Tutorial>

Thunderbird — Haz el correo más fácil. (s. f.). Thunderbird. <https://www.thunderbird.net/es-ES/>

Troubleshooting sSMTP: Authorization failed - Unix & Linux Stack Exchange. (2020, June 27). Unix Stack Exchange. Retrieved May 22, 2023, from <https://unix.stackexchange.com/questions/595410/troubleshooting-ssmtp-authorization-failed>

WordPress.com. (s. f.). *WordPress.com: crea un sitio, vende tus productos, escribe un blog y muchas opciones más*. <https://wordpress.com/es/>

W3.CSS Home. (n.d.). W3Schools. Retrieved May 21, 2023, from

<https://www.w3schools.com/w3css/default.asp>

W3.CSS Templates. (n.d.). W3Schools. Retrieved May 21, 2023, from

https://www.w3schools.com/w3css/w3css_templates.asp