

Administración de un servidor Debian 10



VNiVERSiDAD
DE SALAMANCA

CAMPUS OF INTERNATIONAL EXCELLENCE

Pablo Jesús González Rubio

Sergio García González

Índice

Introducción	4
Configuración Inicial	4
Sin conexión a internet	4
Con conexión a internet	5
SSH	6
SFTP	7
MariaDB	8
Gestión de grupos	9
Apache2	9
Página principal	11
Ayuda	12
Status de los servicios	12
Formulario de login	12
Formulario de registro	15
Contraseña olvidada	19
Dashboard	22
Gestión usuario	22
Modificar datos	22
Modificar contraseña	22
Eliminar cuenta	22
Panel de administrador	22
Página web de los usuarios	22
Servidor de Correos	22
Postfix	22
Instalación	22
Configuración	24
Dovecot	25
Instalación	25

Configuración	25
Roundcube	28
Instalación	28
Uso con Gmail	29
Gestión del almacenamiento	37
Cuotas	37
Instalación de las cuotas	37
Backups	39
Seguridad	40
Fail2Ban	40
Tripwire	42
Monitorización	45
Web de status	45
Envío de correos al administrador	47
Panel de administración	50
Wordpress	50
Mumble	54

Introducción

En este trabajo vamos a mostrar toda información necesaria para la instalación y configuración de un servidor Linux. En nuestro caso toda la configuración mostrada en este informe sirve para una arquitectura muy concreta, en nuestro caso la arquitectura arm64v8 ya que es la que usa la raspberry pi 3B +.

El objetivo de esta práctica es que cualquier usuario pueda tener un servidor Linux de forma sencilla (teniendo algún conocimiento sobre Linux antes). Para ello dentro de los ficheros que proporcionamos ofrecemos un script que permite instalar y configurar prácticamente todo de forma automática.

Configuración Inicial

Sin conexión a internet

Añadimos el usuario admin y al crearlo le asignamos una contraseña.

```
adduser admin
```

Cambiamos la contraseña al usuario Root para poder acceder al él desde el usuario admin.

```
passwd
```

Configuramos los locales para poder utilizar los teclado con mapeado español.

```
echo "export LANGUAGE=es_ES.UTF-8  
export LANG=es_ES.UTF-8  
export LC_ALL=es_ES.UTF-8" >> /root/.bashrc
```

Le cambiamos el nombre al servidor.

```
echo "piratebay" > /etc/hostname
```

Se configura el FQDN para que los servicios que utilizan el fichero hosts como referencia, cojan el dominio completo. Para ello podemos ejecutar este comando.

```
echo "127.0.0.1    nonuser.onthewifi.com localhost piratebay  
::1      nonuser.onthewifi.com localhost piratebay ip6-localhost ip6-loopback  
ff02::1    ip6-allnodes
```

```
ff02::2      ip6-allrouters" > /etc/hosts
```

Se configura SSH para permitir la conexión remota sin interfaz gráfica. La explicación completa se encuentra en el [siguiente apartado](#).

Configuramos el adaptador de red para tener una IP estática y poder acceder siempre con la misma IP. Para ello añadimos el siguiente contenido al fichero “/etc/network/interfaces.d/eth0”:

```
auto eth0
iface eth0 inet static
    address 192.168.1.10
    netmask 255.255.255.0
    gateway 192.168.1.1
```

Añadimos el servidor de nombres de Cloudfare al fichero “/etc/resolv.conf”:

```
nameserver 1.1.1.1
```

El enunciado pedía cambiar el **nivel de ejecución**, lo cual para un servidor al que se accede de forma remota por consola de comandos es razonable. Para poder habilitar el servidor en modo multiusuario no gráfico podemos poner por defecto el nivel de ejecución 3 o el 4. Para ello podemos ejecutar el siguiente comando de Systemd:

```
systemctl set-default runlevel3.target
```

Con conexión a internet

Una vez esté la configuración básica en el servidor Raspberry Pi con Debian 10, la conectamos por Ethernet y la encendemos. Lo primero que debemos hacer es actualizar el servidor.

```
apt update -y && apt upgrade -y
```

Acto seguido instalaremos librerías que el resto de paquetes tanto de CPAN para Perl como otras utilidades necesitarán.

```
apt install net-tools locales jq sudo build-essential pkg-config libgd-dev git cmake cpanminus  
ufw libpam0g libpam0g-dev libmariadb-dev -y
```

Una vez instalado el paquete para los idiomas, podemos editar el fichero /etc/locale.gen, descomentando nuestro idioma “es_ES-UTF8”. O bien ejecutar el siguiente comando.

```
sed -i '/es_ES.UTF-8/s/^#//g' /etc/locale.gen
```

Una vez declarados los locales, tenemos que generarlos. En la configuración sin internet ya los configuramos por defecto, de tal forma que al reiniciar el servidor, los idiomas ya estarán operativos.

```
locale-gen es_ES.UTF-8
```

Ya instalada la herramienta “sudo”, lo siguiente es añadir al usuario “admin” al grupo sudoers.

```
usermod -aG sudo admin
```

Por último, podemos configurar la fecha y hora, pues viene desconfigurada por defecto.

```
timedatectl set-timezone "Europe/Madrid"
```

SSH

El servicio SSH permite conectarnos de forma remota a un servidor mediante consola de comandos para poder gestionarlo. En nuestro caso vino por defecto pero para instalarlo bastaría con ejecutar el comando:

```
apt install ssh -y
```

Para configurar el servicio tenemos que modificar el fichero “sshd_config” en el directorio de configuración de SSH, es decir, “/etc/ssh”.

En este fichero, los parámetros más importantes que debemos modificar son los siguientes:

- Puerto: El parámetro “port” permite configurar en qué puerto escuchará el servicio a las peticiones, siendo el puerto por defecto el 22. Es una buena práctica de seguridad el cambiar el puerto para que los atacantes lo tengan más complicado a la hora de intentar entrar en nuestro servidor. Nosotros hemos elegido el puerto 2222, el cual es otro puerto común para SSH.
- Tiempo de gracia para el login: El parámetro “LoginGraceTime” permite configurar el número de segundos, minutos (ej.: 1m) u horas (ej.: 1h) que tendrá el usuario para poder iniciar sesión en cada intento de contraseña antes de que se le cierre la conexión. Nosotros la hemos configurado a 1 minuto: “LoginGraceTime 1m”.
- No permitir la entrada de Root: El parámetro “PermitRootLogin” permite indicar si el usuario podrá iniciar sesión como Root. Es importante no permitir esto para evitar posibles ataques por fuerza bruta. Es mucho más recomendable iniciar sesión como un usuario normal, y luego escalar privilegios a Root. Nosotros la hemos desabilitado: “PermitRootLogin no”.
- Número máximo de intentos de inicio de sesión: El parámetro “MaxAuthTries” permite seleccionar el número máximo de intentos de inicio de sesión antes de que SSH le cierre la conexión al usuario. Establecer este límite es importante para evitar que un posible atacante pueda atacar por fuerza bruta sin ningún límite el servidor. En nuestro caso lo hemos configurado a 3 intentos: “MaxAuthTries 3”.
- Número máximo de sesiones: El parámetro “MaxSessions” permite configurar el número máximo de sesiones que se podrán tener abiertas simultáneamente. Esto en nuestro caso es recomendable establecerlo para definir el número máximo de dispositivos que se pueden conectar a la vez. En nuestro caso lo hemos configurado a 4, pues pensamos en un dispositivo principal y uno secundario por cada administrador del servidor (ej.: torre y portátil): “MaxSessions 4”.

Para autoconfigurar todos estos parámetros desde consola sin tener que modificar el fichero con un editor, y así poder ejecutarlo en un script usamos la utilidad “sed”:

```
sed -i '/#Port 22/c\Port 2222' /etc/ssh/sshd_config
sed -i '/#LoginGraceTime 2m/c\LoginGraceTime 1m' /etc/ssh/sshd_config
sed -i '/#MaxAuthTries 6/c\MaxAuthTries 3' /etc/ssh/sshd_config
sed -i '/#MaxSessions 10/c\MaxSessions 4' /etc/ssh/sshd_config
```

SFTP

El servicio SSH del paquete OpenSSH viene con SFTP por defecto, para poder utilizarlo no hace falta instalar nada adicional. El enunciado pedía que el usuario al entrar al sistema por SFTP entre directamente al directorio “public_html” donde se guarda la página web, para poder hacer esto hay que “enjaular” por así decirlo, al usuario en la carpeta “public_html”, de tal forma que no pueda salir de ahí; si lo consiguiera podrían darse problemas de seguridad.

Uno de los problemas más graves que podrían ocasionarse si saliera de ese directorio, es que el usuario podría borrar su carpeta de correo pudiendo dejar inhabilitada su función de correo y luego tendría que pedir ayuda al administrador; de esta forma nos aseguramos de que no se den estas circunstancias.

Para hacer efectivo el sistema de jaulas debemos añadir al final de la configuración de SSH (“/etc/ssh/sshd_config”) el fragmento de código siguiente:

```
Match Group usuarios
    ChrootDirectory /home/%u/public_html
    X11Forwarding no
    AllowTcpForwarding no
    PermitTunnel no
    AllowAgentForwarding no
    ForceCommand internal-sftp
```

Esto nos permitirá que todos los usuarios dentro del grupo “usuarios” (que es al que añadimos los usuarios cuando los creamos) no puedan acceder a otra carpeta que la especificada en el parámetro “ChrootDirectory”. Los demás parámetros son de seguridad, para evitar que el usuario pueda enrutar tráfico y utilizar ventanas gráficas.

Una vez hecho esto, si hiciéramos SFTP no nos dejaría acceder y daría un error, esto es porque tenemos que cambiarle al directorio “home” del usuario la propiedad, de tal forma que sea el usuario Root el que posea el fichero, y deberemos hacer lo mismo con la carpeta “public_html”; de esta forma el usuario no podrá acceder a niveles superiores pues no es propietario de la carpeta.

Es importante que los **ficheros** de estas carpetas Sí que pertenezcan al usuario pues si no, se darían lugar a fallos en la gestión del correo (ya que como he mencionado anteriormente, los ficheros del correo electrónico los guardamos dentro de la carpeta de cada usuario; para una explicación detallada de por qué hacemos esto, lo explicamos en su [apartado correspondiente](#)) y no podría tampoco editar los ficheros de su página ni subir unos nuevos.

Para automatizar esto se pueden utilizar los siguientes comandos (son parte de la creación del usuario en el script “dirlookup”):

```
user="miUsuario"  
  
chown root:root /home/$user  
chmod 755 /home/$user  
  
chown -R $user:usuarios /home/$user/*  
chmod 775 /home/$user/*  
  
chown root:root /home/$user/public_html/  
chmod 755 /home/$user/public_html/  
chmod 775 /home/$user/public_html/*
```

Una vez hecho esto, el usuario podría acceder sin mayor problema a su directorio “public_html” desde SFTP, manteniendo las medidas de seguridad adecuadas.

```
Símbolo del sistema - sftp -P 2222 pablo@nonuser.onthewifi.com  
Microsoft Windows [Versión 10.0.19042.1052]  
(c) Microsoft Corporation. Todos los derechos reservados.  
C:\Users\pjgr2>sftp -P 2222 pablo@nonuser.onthewifi.com  
pablo@nonuser.onthewifi.com's password:  
Connected to nonuser.onthewifi.com.  
sftp> ls  
css index.html js  
sftp> pwd  
Remote working directory: /  
sftp>
```

MariaDB

Por defecto se instala con el usuario y contraseña de root.

```
admin@admin@piratebay: ~$ sudo mariadb
sudo: unable to resolve host piratebay: Name or service not known
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 52
Server version: 10.3.27-MariaDB-0+deb10u1 Debian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

```
apt install mariadb-server -y

# Cambiar contraseña para root
systemctl stop mysql
systemctl stop mariadb
mysqld_safe --skip-grant-tables --skip-networking &
mysql -u root < ficheros/mariadb/createDB.sql

# Crear base de datos Usuarios y tabla Datos
mysql --user=root --password=admin < ficheros/mariadb/createDB.sql
mysql --user=root --password=admin usuarios < ficheros/mariadb/createTable.sql
```

Para la instalación de MariaDB ejecutamos los siguientes comandos que tenemos en el script “installServer.sh”.

Lo que hacemos es instalar MariaDB y tras instalarla parar su servicio para poder realizar la configuración del usuario root de forma segura.

Una vez hemos realizado lo anterior debemos crear las bases de datos y las tablas de usuarios que necesitamos para el portal, como pueden ser la base de datos para el usuario o la base de datos de wordpress.

Gestión de grupos

Todos los usuarios una vez se añadan se encontrarán en el grupo “usuarios”. Este grupo es creado con el siguiente comando:

```
groupadd usuarios
```

Podemos ver su gid leyendo el fichero “/etc/group”, lo cual después de haber añadido al usuario “admin” nos muestra un gid de 1001, el cual utilizaremos para el formulario de registro, pues todos los usuarios que se registren pertenecerán al grupo “usuarios”.

Apache2

Para instalar el servidor de páginas web hemos instalado Apache. Aunque también hubiéramos podido instalar Nginx, hemos preferido Apache por la versatilidad de sus módulos.

Para instalar el servidor web Apache podemos ejecutar el siguiente comando.

```
apt install apache2 -y
```

Una vez instalado, con ir a la página “<http://localhost:80>” nos mostraría la pantalla inicial de Apache. Si quisiéramos modificar esta página, podemos editar los ficheros del directorio “/var/www/html”; en los siguientes apartados explicaremos cómo hemos hecho la página y cómo hemos gestionado los inicios de sesión y la propia gestión del usuario desde la página.

Para configurar dónde buscará Apache los ficheros HTML y demás podemos editar el fichero “/etc/apache2/sites-available/000-default.conf”, en concreto el parámetro “DocumentRoot” como se ve a continuación.

```
<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 nonuser.onthewifi.com

    ServerAdmin admin@nonuser.onthewifi.com
    DocumentRoot /var/www/html

    # 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

    # 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>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

En el fichero “/etc/apache2/apache2.conf” hemos cambiado la parte que gestiona los permisos de visualización de la página, en concreto hemos cambiado los parámetros por defecto que son los siguientes:

```
<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>
```

Por estos:

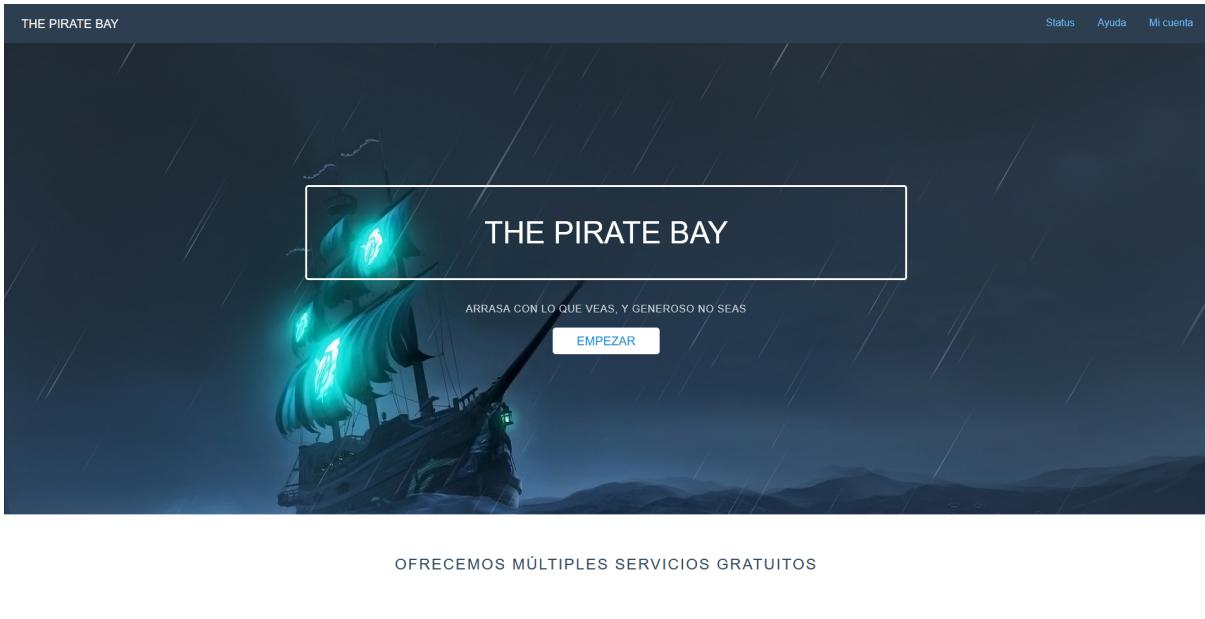
```
<Directory />
    Options FollowSymLinks
    AllowOverride all
</Directory>

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

<Directory /var/www/>
    Options FollowSymLinks
    AllowOverride all
    Order allow,deny
    allow from all
</Directory>
```

Esto lo hemos hecho para permitir que se ejecuten scripts CGI sin ningún tipo de error.

Página principal



A screenshot of a services section on the website. It features a dark blue header with the "THE PIRATE BAY" logo and navigation links for "Status", "Ayuda", and "Mi cuenta". Below the header, the text "OFRECEMOS MÚLTIPLES SERVICIOS GRATUITOS" is centered. Four service options are listed in a grid: "Servidor de correos" (with a mail icon), "Almacenamiento para Webs" (with a folder icon), "Creación de Blogs con WordPress" (with a blog icon), and "Servidor VOIP Mumble" (with a phone icon). Each service has a brief description below it. Further down, there's a section titled "SERVICIO SFTP" with a yellow folder icon labeled "SFTP" and a small circular progress bar.

CONTACTO

Contacta con nosotros

Puedes contactar con nosotros a través del siguiente correo admin@nonuser.onthewifi.com

Las imágenes superiores refieren a la pantalla inicial de la página “nonuser.onthewifi.com”. El archivo correspondiente a la pantalla inicial es el “html/index.html”.

Desde esta pantalla se puede acceder al estatus de los servicios, a la página de ayuda y al inicio de sesión.

Si se ha iniciado sesión anteriormente y la sesión sigue siendo válida, el apartado “Mi cuenta” redirigirá al Dashboard en el caso de los usuarios normales, y al panel de administración en el caso de ser el usuario “admin”.

Ayuda



Bienvenido al servidor The Pirate Bay, servidor con uso exclusivo educativo. Para poder utilizar esta plataforma debe crearse una cuenta, para ello diríjase al apartado Mi Cuenta y desde allí al apartado de registro.

Una vez se haya registrado puede gozar de nuestros múltiples servicios que le ofrecemos desde poder alojar su página web hasta el uso de nuestro exclusivo servicio de voz.

Si quiere modificar cualquier apartado de su cuenta, se tiene que dirigir al apartado de su Dashboard y desde allí al apartado Ajustes. Una vez este allí ya puede modificar sus datos, cambiar su contraseña o eliminar su cuenta.

Le recordamos que cuando modifique los datos no puede usar un correo que ya use otro usuario ni puede dejar campos en blanco.

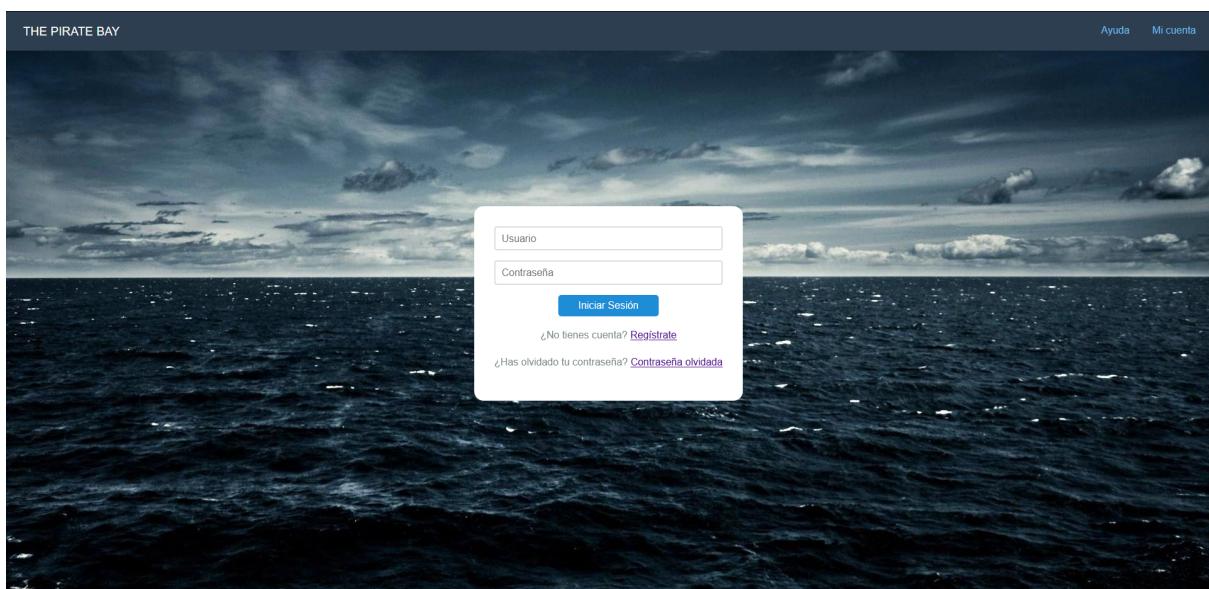
Copyright © 2021, The Pirate Bay
All rights reserved.

La imagen superior se refiere a la pantalla de ayuda (“nonuser.onthewifi.com/ayuda.html”). El archivo correspondiente a la pantalla inicial es el “html/ayuda.html”.

Status de los servicios

El apartado de status se explica en Monitorización.

Formulario de login



La gestión del Login se hace mediante dos scripts CGI, “login.cgi” que muestra la interfaz y dependiendo de si se había iniciado sesión con anterioridad redirige al usuario a una pantalla o a otra, y el script “logged.cgi” que gestiona el inicio de sesión.

El script realmente importante es el de “logged” así que es el que explicaremos en detalle.

```
use CGI;
use CGI::Cookie;
use CGI::Session;
use Authen::PAM;
use POSIX;
use utf8;

$cgi = CGI->new;
$username = $cgi->param(-name => 'username', -value => $cgi->param('email'));
$password = $cgi->param(-name => 'password', -value => $cgi->param('password'));
$service = "passwd";

# This "conversation function" will pass
# $password to PAM when it asks for it.
sub my_conv_func {
    my (@res);
    while (@_) {
        my $code = shift;
        my $msg = shift;
        my $ans = "";

        $ans = $username if ($code == PAM_PROMPT_ECHO_ON());
        $ans = $password if ($code == PAM_PROMPT_ECHO_OFF());

        push @res, (PAM_SUCCESS(), $ans);
    }
    push @res, PAM_SUCCESS();
    return @res;
}

# Initialize PAM object
if (!ref($pamh = new Authen::PAM($service, $username, \&my_conv_func))) {
    print "Authen::PAM init failed\n";
    exit 1;
}

# Authenticate with PAM
my $res = $pamh->pam_authenticate;

# Return success or failure
if ($res == PAM_SUCCESS()) {
    if ($username eq "admin") {
        my $session = new CGI::Session;
        $session->save_param($cgi);
        $session->expires("+1h");
        $session->flush();
        print $session->header(-location => "admin.cgi");
    }
    else{
        # https://www.youtube.com/watch?v=qtRRXy2oNUQ
        my $session = new CGI::Session;
        $session->save_param($cgi);
        $session->expires("+1h");
        $session->flush();
        print $session->header(-location => "dashboard.cgi");
    }
}
```

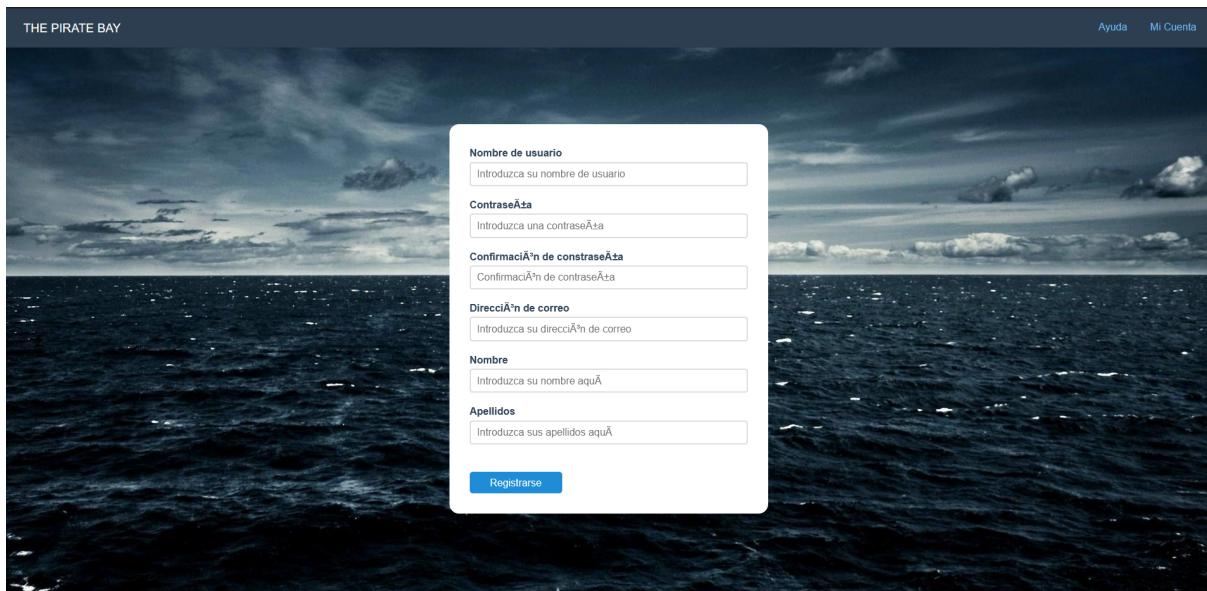
```
    }
} else {
    print $cgi->redirect("https://nonuser.onthewifi.com/cgi-bin/login.cgi");
}
```

El script recoge los parámetros “email” y “password” del script “login.cgi” y autentifica al usuario mediante PAM con la librería Authen::PAM de CPAN. Si el usuario no existe o el usuario/contraseña del usuario son erróneos la función devuelve falso y se redirige al usuario a la pantalla de login otra vez. En caso de autenticarse correctamente se le crea una sesión y se le redirige a la pantalla-script “dashboard.cgi”.

Para que el script pueda autenticar al usuario mediante PAM se necesita permisos de lectura al fichero shadow para comprobar la contraseña. Lo que hemos hecho ha sido añadir “www-data” al grupo “shadow” de tal forma que pueda leer el fichero.

```
sudo usermod -aG shadow www-data
```

Formulario de registro



En el apartado de registro se hacen varias cosas por detrás:

- Se crea el usuario
- Se crea su carpeta en el directorio /home
- Se asigna la Quota correspondiente al usuario creado
- Gestionar los permisos de los directorios para evitar problemas con el correo
- Se copian los datos de /etc/skel a los directorios de los usuarios
- Se crea el usuario en la base de datos
- Se manda un correo con los datos del usuario

La interfaz que se muestra con el formulario corresponde al script “cgi-bin/register.cgi”, el script que gestiona parte de lo mencionado arriba es el script “cgi-bin/**registered.cgi**” y un script (“**dirlookup**”) que se ejecuta cada segundo que gestiona la otra parte.

Un aspecto importante que tratamos es que no se permite que ningún nuevo usuario se pueda registrar con un nombre de usuario o correo electrónico ya registrados, para evitar posibles incidentes.

En el archivo passwd debemos darle permisos especiales.

```
sudo chmod 664 /etc/passwd
```

Para que el register.cgi pueda funcionar correctamente y registre un usuario desde la página web, tenemos que añadir al usuario www-data al grupo shadow lo cual hicimos en el apartado de login.

Además debemos darle permisos de escritura al fichero shadow al grupo www-data, para que el script pueda funcionar correctamente y no de un error interno del servidor Apache, para ellos utilizamos el siguiente comando:

```
sudo chmod 660 /etc/shadow
```

Tenía permisos de lectura y escritura para el usuario, y sólo de lectura para grupos, entonces le añadimos el de escritura para www-data.

Para poder crear al usuario nuevo, su propia carpeta en el directorio “/home”, hemos tenido que cambiar el grupo del directorio y cedérselo a www-data, y luego darle permisos de escritura a este directorio. De esta forma, el comando “mkdir” de Perl nos crea la carpeta.

```
sudo chgrp www-data /home  
sudo chmod 775 /home
```

Los scripts son los siguientes.

registered.cgi

```
#!/usr/bin/perl -w

use warnings;
use utf8;

use CGI;
use CGI::Cookie;
use CGI::Session;

use Linux::usermod;
use File::Copy::Recursive;

use SQL::Abstract;
use DBI;

use Email::MIME;
use Email::Sender::Simple qw(sendmail);
use MIME::Words qw(:all);

$cgi = CGI->new;
```

```

# Valores página web
$username = lc($cgi->param('username'));
$password = $cgi->param('password1');
$email = $cgi->param('email');
$name = $cgi->param('name');
$surname = $cgi->param('surname');

# Base datos
$usuarioDB = "root";
$claveDB = "admin";
$DB = "usuarios";
$tabla = 'datos';

# Usuario Sistema
$directory = "/home/$username";
$group = "1001";
$shell = "/usr/sbin/nologin";

# Creamos el objeto para SQL Abstract
my $sql = SQL::Abstract->new;

# Conectamos con la BD.
$dbh = DBI->connect("DBI:MySQL:$DB:localhost", $usuarioDB, $claveDB) or print "\nError al abrir la base de datos.\n";

my %where = (email => $email);
my ($stmt, @bind) = $sql->select($tabla, 'usuario', \%where);
my $sth = $dbh->prepare($stmt);
$sth->execute(@bind);
$query = $sth->fetchrow_array;
if ($query eq ""){
    # Datos usuarios
    my %data = (
        usuario => $username,
        nombre => $name,
        apellidos => $surname,
        email => $email,
    );
    # Genera Query para insertar
    my ($stmt, @bind) = $sql->insert($tabla, \%data);

    # Ejecutar Query
    my $sth = $dbh->prepare($stmt);
    $sth->execute(@bind);

    # Desconectamos de la BD.
    $dbh->disconnect or warn "\nFallo al desconectar.\n";
}

Linux::usermod->add($username, $password, $group, $directory, $shell);
($name, $pass, $uid, $gid, $quota, $comment, $gcos, $dir, $shell, $expire) = getpwnam($username);

$chownFile="/var/www/nameNew/$username";
open(FH, '>', $chownFile) or print "Failed to create empty: $!\n";
close(FH);

```

```

# Mandar correo
$BODY = "HTML con información del usuario";
my $message = Email::MIME->create(
    header_str => [
        From => '"The Pirate Bay" <admin@nonuser.onthewifi.com>',
        To => $email,
        Subject => "¡Registrado satisfactoriamente!",
        Charset => 'utf-8',
        Encoding => 'B',
        'Content-Type' => 'text/html',
    ],
    attributes => {
        encoding => 'base64',
        charset => 'UTF-8',
    },
    body_str => $BODY,
);
sendmail($message);

# Se crea una nueva
$session = CGI::Session->new();
$session->save_param($cgi);
$session->expires("+1h");
$session->flush();
print $session->header(-location => "dashboard.cgi");
}

else {
    # Desconectamos de la BD.
    $dbh->disconnect or warn "\nFallo al desconectar.\n";
    print $cgi->header;
    print qq(HTML con mensaje de ERROR:Ya existe un usuario con ese correo);
}

```

En el script CGI de arriba si no hay ningún usuario con el correo especificado en el sistema, se permite la creación de ese usuario: se inserta una fila en la base de datos, se crea su usuario en el sistema (en el passwd y en el shadow) añadiéndolo al grupo “usuarios”, se crea un fichero que gestionará el servicio “dirlookup”, se manda un correo al usuario con sus datos y finalmente se le crea una sesión al usuario y se le redirige al usuario a la pantalla del dashboard.

Si el usuario ya existe sea porque hay un usuario con ese correo, o sea un usuario del sistema como pueda ser “admin” o “root”, le saltará un error y no le dejará registrarse.

dirlookup:

```
#!/bin/sh

while true; do
    dirNew="/var/www/nameNew/"
    dirDel="/var/www/nameDel/"

    soft="50M"
    hard="80M"

    sleep 1
    # New Users
    for file in $(ls $dirNew); do
        mkdir /home/$file
        chmod 755 /home/$file
        cp -r /etc/skel/* /home/$file/

        # Permitir chroot en SFTP
        chown root:root /home/$file
        chmod 755 /home/$file

        chown -R $file:usuarios /home/$file/*
        chmod 775 /home/$file/*

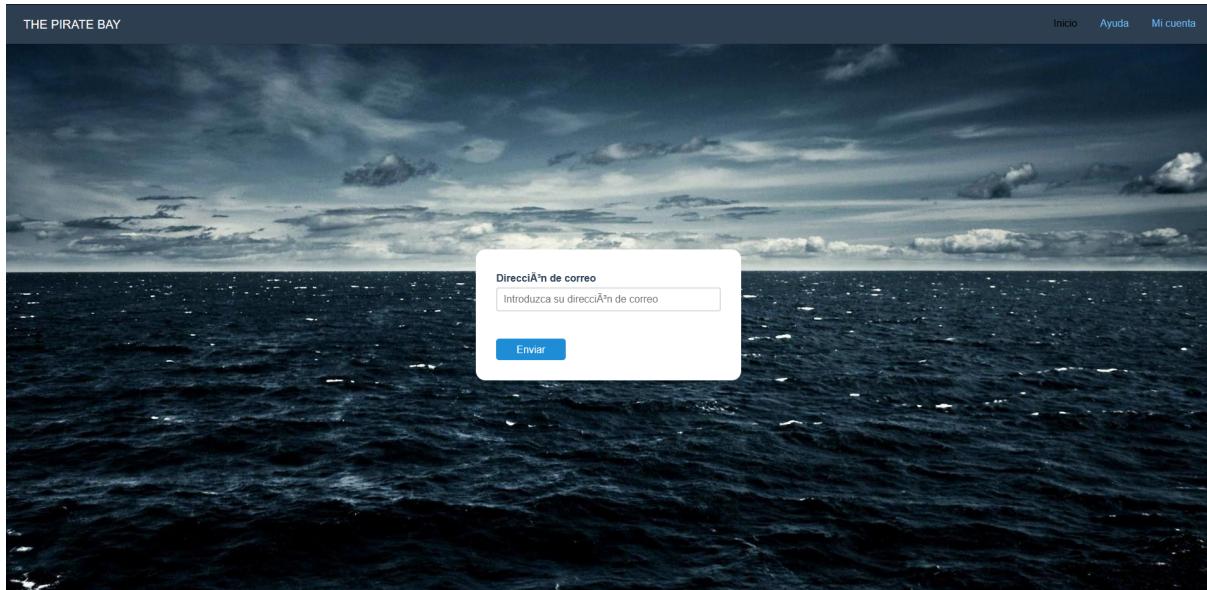
        chown root:root /home/$file/public_html/
        chmod 755 /home/$file/public_html/
        chmod 775 /home/$file/public_html/*

        echo "[NEW][$(date "+%H:%M:%S %d-%m-%Y")] $file" >> /home/admin/usuarios.log
        rm "$dirNew$file"
        setquota -u $file $soft $hard 0 0 /
    done
    # Deleted Users
    for file in $(ls $dirDel); do
        echo "[DEL][$(date "+%H:%M:%S %d-%m-%Y")] $file" >> /home/admin/usuarios.log
        rm "$dirDel$file"
        rm -r /home/$file
    done
done
```

Este script gestiona el crear el directorio del usuario; asignarle los permisos para permitir el chroot y poder seguir gestionando el correo y manejar la página web del usuario; y por último le asigna la quota.

Este script también gestiona la eliminación de la carpeta del usuario cuando en la página se especifica el borrar un usuario.

Contraseña olvidada



El usuario puede solicitar una contraseña nueva si no recuerda la suya y esta se le enviará por correo electrónico al email que especificó cuando creó el usuario.

Este apartado lo gestionan dos scripts: “forgot.cgi” que se encarga de la interfaz web, y “forgotten.cgi” que comprueba si existe un usuario con ese correo y de ser así, crea una contraseña nueva que se le manda al usuario por correo, después se muestra una pantalla para que revise su correo. Si el usuario no existe, se le muestra un HTML indicándole que el usuario no existe y un botón para volver al inicio.

A continuación se muestra el script “forgotten.cgi” pues es el más interesante desde el aspecto de la Administración del Sistema.

forgotten.cgi:

```
#!/usr/bin/perl -w

use warnings;
use Linux::usermod;
use CGI;
use utf8;
use SQL::Abstract;
use DBI;
use Email::MIME;
use Email::Sender::Simple qw(sendmail);
use MIME::Words qw(:all);
use Crypt::RandPasswd;

$usuarioDB = "root";
$claveDB = "admin";
$DB = "usuarios";
$tabla = 'datos';
```

```

$qq = CGI->new;
$email = $qq->param('email');

$minlen = 12;
$maxlen = 20;

# Gestión base datos
my $sql = SQL::Abstract->new();
$dbh = DBI->connect("DBI:MariaDB:$DB:localhost", $usuarioDB, $claveDB) or print "\nError al abrir la
base de datos.\n";
my %where = ($email => $email);
my($stmt, @bind) = $sql->select($tabla, 'usuario', \%where);
my $sth = $dbh->prepare($stmt);
$sth->execute(@bind);
my $username = $sth->fetchrow_array;
$dbh->disconnect or warn "\nFallo al desconectar.\n";

if ($username ne "") {
    $password = Crypt::RandPasswd->letters( $minlen, $maxlen );
    $user = Linux::usermod->new($username);
    $user->set(password => $password);

    $BODY='HTML para el correo con contraseña nueva';
    my $message = Email::MIME->create(
        header_str => [
            From => '"The Pirate Bay" <admin@nonuser.onthewifi.com>',
            To => $email,
            Subject => "¡Contraseña cambiada!",
            Charset => 'utf-8',
            Encoding => 'B',
            'Content-Type' => 'text/html',
        ],
        attributes => {
            encoding => 'base64',
            charset => 'UTF-8',
        },
        body_str => $BODY,
    );
    sendmail($message);

    print $qq->header;
    print qq(HTML para decir que revise el correo);
}
else {
    print $qq->header;
    print qq(HTML usuario no existe);
}

```

Dashboard

The screenshot shows the dashboard for user "pablo". At the top, it says "BIENVENIDO PABLO". Below that, there are four service links:

- Servidor de correos**: "Accede a tu correo de forma fácil y rápida" with a "Continuar" button.
- Almacenamiento para Webs**: "Crea un espacio para tu web (Almacenamiento 5MB)" with a "Crear" button.
- Creación de Blogs con WordPress**: "Crea tus blogs personalizados y de forma rápida con WordPress" with a "Crear" button.
- Servicio de VOIP Mumble**: "Hable con sus amigos fácilmente con nuestro servicio de voz Mumble" with fields for "Dirección" (nonuser.onthewifi.com) and "Puerto" (64738), and a "Descargar" button.

At the bottom, it says "Copyright © 2021, The Pirate Bay All rights reserved."

La imagen superior representa el dashboard del usuario “pablo”.

Esta página la gestiona el script “dashboard.cgi” que mediante la información que guardamos en la sesión, podemos obtener el nombre de usuario para mostrar el texto de “Bienvenido” y el comando de SFTP.

Desde esta pantalla se puede acceder a los otros servicios de los que dispone el servidor.

Gestión usuario

The screenshot shows the user management page for user "pablo". It includes links for managing account data and deleting the account:

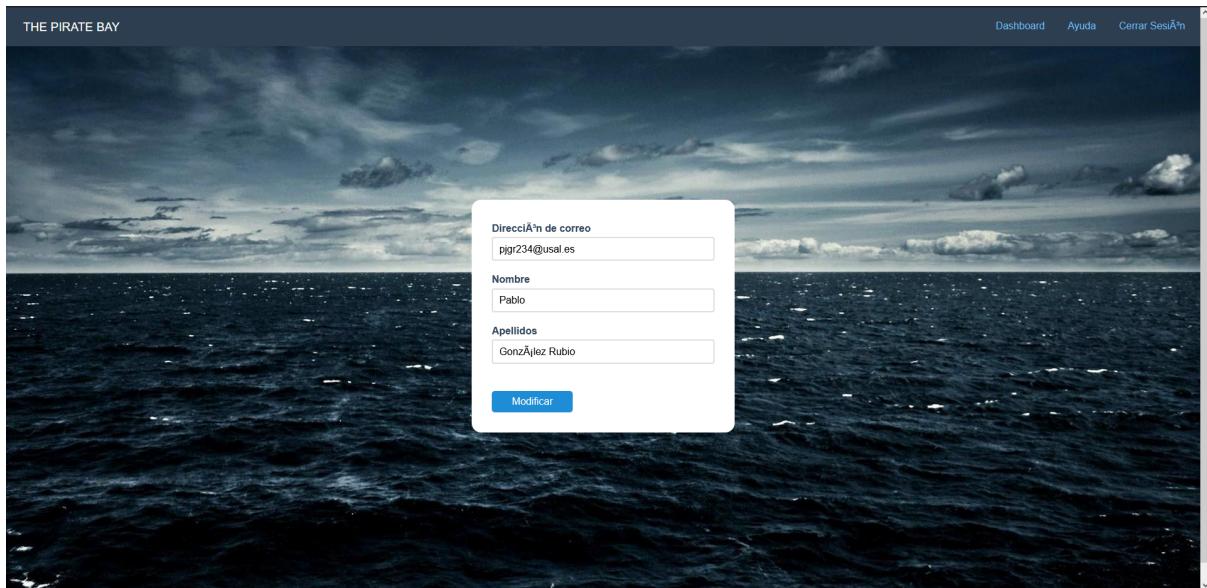
- "Modificar mis datos" with a "Modificar" button.
- "Cambiar contraseña" with a "Cambiar contraseña" button.
- "Eliminar cuenta" with a "Eliminar" button.

At the bottom, it says "Copyright © 2021, The Pirate Bay All rights reserved."

La imagen superior representa la gestión del usuario.

Esta página la gestiona el script “account.cgi” y permite acceder a los otros scripts “modify.cgi” para modificar los datos del usuario, “password.cgi” para cambiar la contraseña y “delete.cgi” para eliminar la cuenta.

Modificar datos



La imagen superior representa la gestión del usuario.

Esta página la gestiona el script “modify.cgi” que accede a la base de datos antes de presentar la pantalla para mostrarle al usuario los datos que tiene guardados para que sepa qué tiene guardado y pueda modificarlo con más facilidad.

Una vez cambie los datos y le de a “Modificar”, el script “modified.cgi” se ejecuta y coge los datos mediante el campo “name” de los campos de entrada. Este script simplemente actualiza la fila de la base de datos con los datos nuevos.

El código de “modified.cgi” es el siguiente:

```
#!/usr/bin/perl -w

use warnings;
use Linux::Usermod;
use CGI;
use CGI::Session;
use utf8;
use SQL::Abstract;
use DBI;

$q = CGI->new;

my $session = new CGI::Session;
$session->load();
my @autenticar = $session->param;
my $username = $session->param("username");
$usuarioDB = "root";
$claveDB = "admin";
$DB = "usuarios";
$tabla = 'datos';
```

```

$email = $q->param('email');
$name = $q->param('name');
$surname = $q->param('surname');

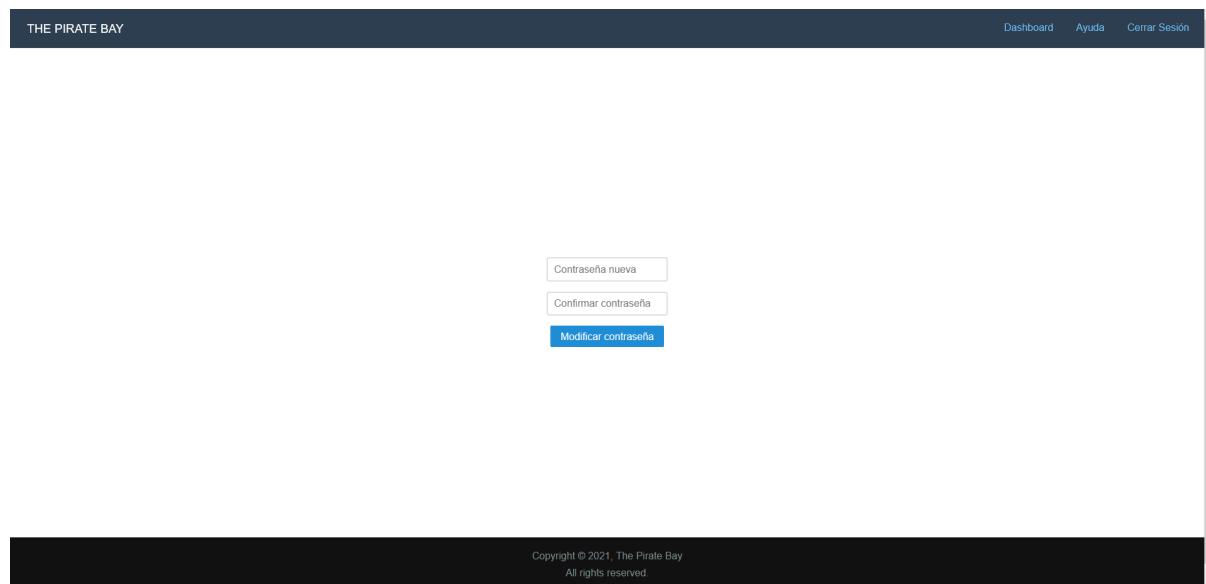
if (@autenticar eq 0) {
    $session->delete();
    $session->flush();
    print $q->redirect("https://nonuser.onthewifi.com/");
} elsif ($session->is_expired) {
    $session->delete();
    $session->flush();
    print $q->redirect("https://nonuser.onthewifi.com/");
} else {

    # Gestión base datos
    my $sql = SQL::Abstract->new;
    $dbh = DBI->connect("DBI:MySQL:$DB:localhost", $usuarioDB, $claveDB) or print "\nError al abrir la
base de datos.\n";

    my %where = (email => $email);
    my($stmt, @bind) = $sql->select($tabla,'usuario', \%where);
    my $sth = $dbh->prepare($stmt);
    $sth->execute(@bind);
    $query = $sth->fetchrow_array;
    if ($query eq $username){
        my %data = (
            nombre => $name,
            apellidos => $surname,
            email => $email,
        );
        my %where = (
            usuario => $username
        );
        my($stmt, @bind) = $sql->update($tabla, \%data, \%where);
        my $sth = $dbh->prepare($stmt);
        $sth->execute(@bind);
        $dbh->disconnect or warn "\nFallo al desconectar.\n";
        print $q->redirect ("https://nonuser.onthewifi.com/cgi-bin/account.cgi");
    }
    else{
        $dbh->disconnect or warn "\nFallo al desconectar.\n";
        print $q->header;
        print qq(HTML usuario con correo introducido ya existe);
    }
}

```

Modificar contraseña



La imagen superior representa la gestión del usuario.

Esta página la gestiona el script “password.cgi” que una vez introducidos las contraseñas, al seleccionar “Modificar contraseña” se ejecuta el script “passwd.cgi” que coge las contraseñas por el “name” del campo de entrada y cambia las contraseñas una vez seleccionado el usuario del sistema.

El código “passwd.cgi” es el siguiente:

```
#!/usr/bin/perl -w

use warnings;
use Linux::usermod;
use CGI;
use CGI::Session;
use utf8;

$q = CGI->new;

my $session = new CGI::Session;
$session->load();
my @autenticar = $session->param;
my $username = $session->param("username");

if (@autenticar eq 0) {
    $session->delete();
    $session->flush();
    print $q->redirect("https://nonuser.onthewifi.com/");
} elsif ($session->is_expired) {
    $session->delete();
    $session->flush();
    print $q->redirect("https://nonuser.onthewifi.com/");
```

```

}

else {
    $password = $q->param('password1');
    $user = Linux::usermod->new($username);
    $user->set(password => $password);
    print $q->redirect ("https://nonuser.onthewifi.com/cgi-bin/account.cgi");
}

```

Eliminar cuenta

El botón de “eliminar cuenta” en la gestión del usuario activa el script “delete.cgi” que coge el nombre del usuario de la sesión y hace los siguientes pasos:

- Elimina al usuario del sistema
- Elimina la fila correspondiente en la base de datos
- Crea el fichero con el nombre del usuario para que el servicio “dirlookupd” mencionado anteriormente lo elimine.

El código es el siguiente:

```

#!/usr/bin/perl -w

use warnings;
use Linux::usermod;
use CGI;
use CGI::Session;
use utf8;
use SQL::Abstract;
use DBI;

# Base datos
$usuarioDB = "root";
$claveDB = "admin";
$DB = "usuarios";
$tabla = "datos";

$q = CGI->new;

# Gestión sesión
my $session = new CGI::Session;
$session->load();
my @autenticar = $session->param;
my $username = $session->param("username");

```

```

if (@autenticar eq 0) {
    $session->delete();
    $session->flush();
    print $q->redirect("https://nonuser.onthewifi.com/");
} elsif ($session->is_expired) {
    $session->delete();
    $session->flush();
    print $q->redirect("https://nonuser.onthewifi.com/");
} else {
    print $q->header;

    # Se borra al usuario
    Linux::usermod->del($username);

    # Se añade fichero para que servicio lo borre
    $delUser="/var/www/nameDel/$username";
    open(FH, '>', $delUser) or print "Failed to create empty: $!\n";
    close(FH);

    # Gestión Base Datos
    my $sql = SQL::Abstract->new;
    $dbh = DBI->connect("DBI:MariaDB:$DB:localhost", $usuarioDB, $claveDB) or die "\nError al
abrir la base de datos.\n";
    my %where = (
        usuario => $username
    );
    my($stmt, @bind) = $sql->delete($tabla, \%where);
    my $sth = $dbh->prepare($stmt);
    $sth->execute(@bind);
    $dbh->disconnect or warn "\nFallo al desconectar.\n";

    # Se borra la sesión
    $session->delete();
    $session->flush();
    print qq(HTML usuario borrado exitosamente);
}

```

Panel de administrador

El [panel de administrador](#) se explica en el apartado Monitorización.

Página web de los usuarios

Para que los usuarios puedan tener su propia página web, es necesario habilitar el módulo “userdir” de Apache. Podemos hacerlo con el siguiente comando:

```
a2enmod userdir
```

A partir de este momento los usuarios podrán editar y disponer de su página web en el directorio “public_html” de su carpeta personal, a la que podrán acceder mediante SFTP para poder gestionar su página.

Suponiendo que el usuario Pablo tiene subida una página web, podrá acceder desde un enlace similar a este “midominio.com/~pablo”; en nuestro caso sería “nonuser.onthewifi.com/~pablo”.

Esto es así porque en la configuración del módulo userdir (en el directorio “/etc/apache2/mods-enabled”) se especifica la carpeta “public_html” dentro de cada carpeta de los usuarios, pero podría usarse otra.

Let's Encrypt

Hemos empleado para cifrar las conexiones tanto de la página web, como del servidor de correos la plataforma Let's Encrypt, que nos proporciona un certificado válido y refutado para que no aparezca en el buscador que es una página de reputación sospechosa, ya que el certificado SSL podríamos haberlo hecho nosotros con OpenSSL.

Para instalarlo hemos hecho uso de “certbot”, la cual es una herramienta que con un comando es capaz de facilitarnos el certificado para nuestro dominio.

Para ello necesitamos instalarla:

```
apt update && apt install certbot python-certbot-apache -y
```

Y acto seguido le especificamos que queremos el certificado para Apache:

```
certbot --apache --redirect -d nonuser.onthewifi.com -m admin@nonuser.onthewifi.com  
--agree-tos
```

Una vez aceptados los términos del servicio, nos automatizará la instalación del certificado, modificando a su vez los ficheros de configuración de la página web: /etc/apache2/sites-enabled/000-default.conf y creará uno nuevo para el certificado SSL.

Con esto tendríamos una página web totalmente cifrada y con un certificado de reputación válido.

Servidor de Correos

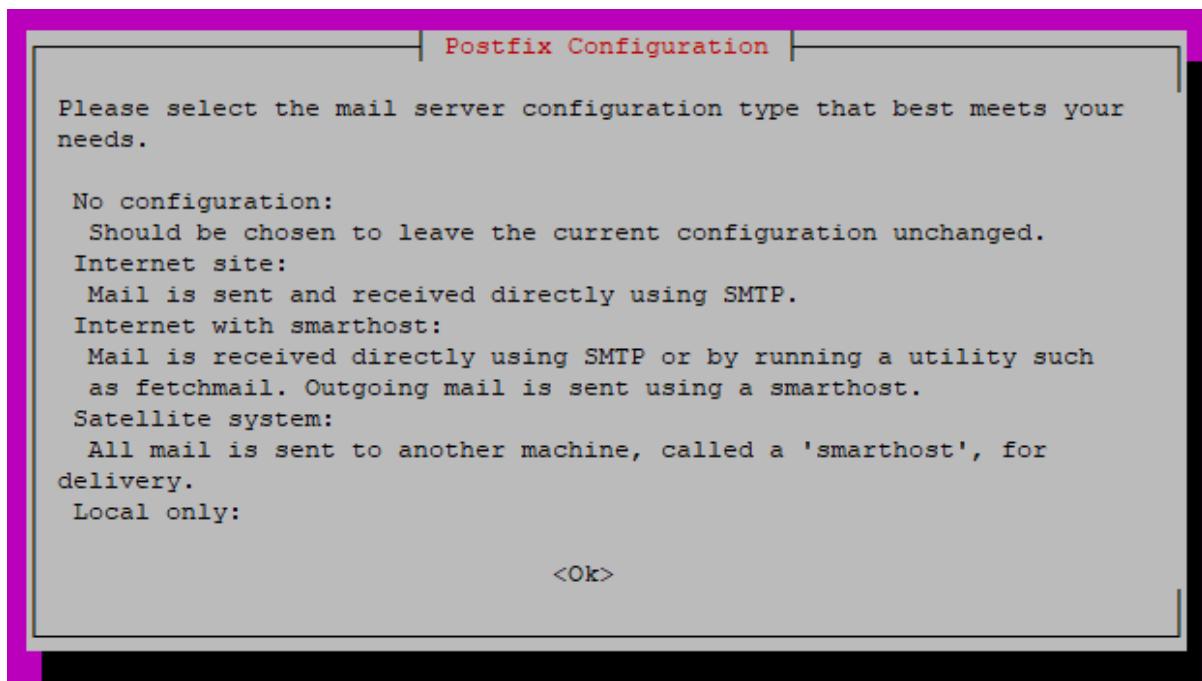
Postfix

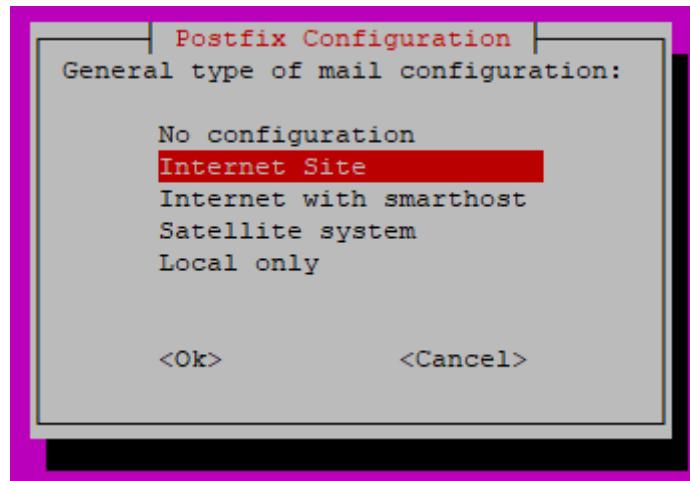
Instalación

Para instalarlo comprobamos que tengamos las últimas versiones de las aplicaciones y a continuación instalamos postfix.

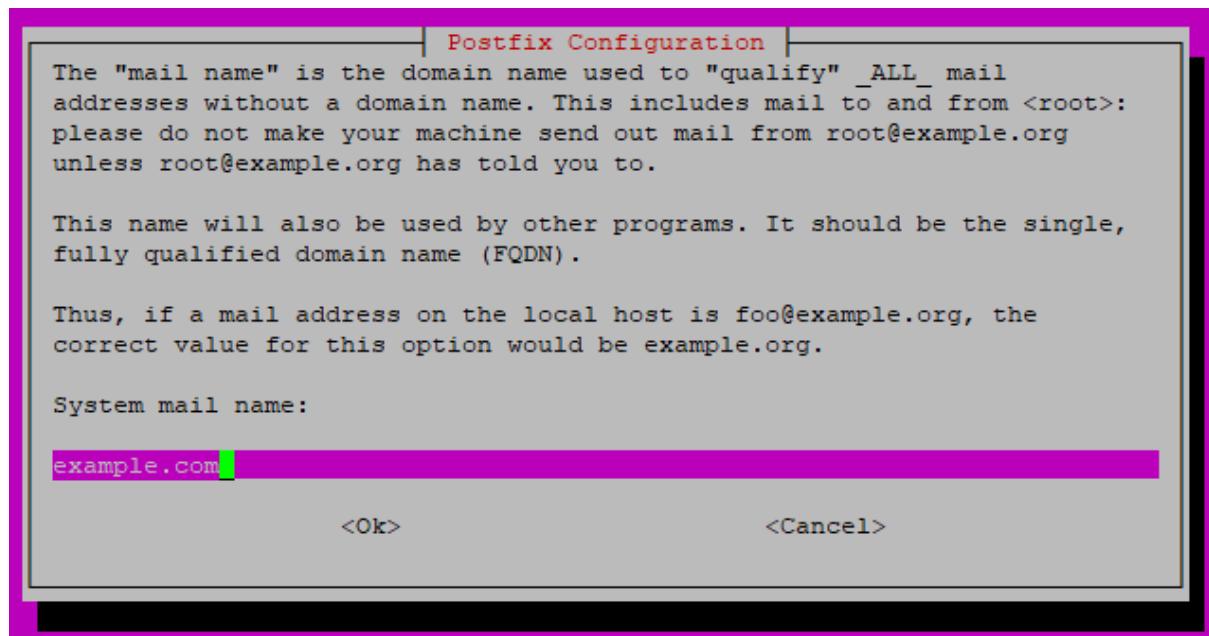
```
sudo apt update -y && sudo apt install postfix -y
```

En el instalador se nos dará varias opciones sobre cómo se pretende utilizar el servidor, a nosotros la que nos interesa es “Internet Site”, ya que seremos nosotros los que alojaremos el servicio.





En este caso, en vez de “example.com” nosotros escribimos nuestro dominio “nonuser.onthewifi.com”.



Configuración

En el fichero **/etc/postfix/main.cf** hemos cambiado/añadido lo siguiente:

```
myorigin = /etc/mailname
mydomain=nonuser.onthewifi.com
myhostname = nonuser.onthewifi.com

smtp_use_tls=yes
smtp_tls_security_level = may
smtpd_tls_cert_file = /etc/letsencrypt/live/nonuser.onthewifi.com/fullchain.pem
smtpd_tls_key_file = /etc/letsencrypt/live/nonuser.onthewifi.com/privkey.pem
smtpd_tls_security_level = may
smtpd_sasl_auth_enable = yes
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
home_mailbox = Maildir/
```

Es importante también que los correos se envíen encriptados, para ello hacemos uso de la misma encriptación que nos brinda “Let’s Encrypt”.

Para que los usuarios reciban los correos dentro de su carpeta personal en vez de en el directorio “/var/mail” hay que decirle a Postfix que utilice el tipo de correos Maildir en vez de MailBox, pues luego configuraremos esto también en Dovecot.

La idea de que sea “Maildir” en vez de “Mailbox”, es que el primero permite crear un fichero por cada correo recibido en vez de almacenarlos en un sólo fichero, lo que a nivel de gestión de ficheros es mucho más eficiente.

Para poder recibir correos del exterior se necesita hacer un mapeo y redirigir los correos con dirección “nonuser.onthe wifi.com” a los usuarios que se encuentran como “localhost”.

```
echo "@nonuser.onthewifi.com @localhost" > /etc/postfix/vmailbox
sudo postmap /etc/postfix/vmailbox
```

Para habilitar que otros clientes como Gmail puedan acceder a la configuración del servidor, tenemos que habilitar la siguiente configuración descomentando los parámetros del siguiente fichero.

Fichero **/etc/postfix/master.cf**

```
smtp  inet  n       -       y       -       -       smtpd
#smtp    inet  n       -       y       -       1       postscreen
#smtpd   pass -      -       y       -       -       smtpd
#dnsblog unix -     -       y       -       0       dnsblog
#tlsproxy unix -    -       y       -       0       tlsproxy
submission inet n    -       y       -       -       smtpd
-o syslog_name=postfix/submission
-o smtpd_tls_security_level=encrypt
-o smtpd_tls_wrappermode=no
-o smtpd_sasl_auth_enable=yes
-o smtpd_relay_restrictions=permit_sasl_authenticated,reject
-o smtpd_recipient_restrictions=permit_mynetworks,permit_sasl_authenticated,reject
-o smtpd_sasl_type=dovecot
-o smtpd_sasl_path=private/auth
-o smtpd_tls_auth_only=yes
-o smtpd_reject_unlisted_recipient=no
-o smtpd_client_restrictions=$mua_client_restrictions
-o smtpd_helo_restrictions=$mua_helo_restrictions
-o smtpd_sender_restrictions=$mua_sender_restrictions
-o smtpd_recipient_restrictions=
-o milter_macro_daemon_name=ORIGINATING
```

Dovecot

Instalación

```
sudo apt install dovecot-imapd -y
```

Configuración

Para modificar un fichero podemos usar el siguiente comando:

```
sudo nano fichero
```

Primero configuramos Dovecot para que los correos lleguen a la carpeta del usuario:

Fichero **/etc/dovecot/conf.d/10-mail.conf**

Y cambiamos la línea:

```
mail_location = mbox:~/mail:INBOX=/var/mail/%u
```

Por:

```
mail_location = maildir:~/Maildir
```

Lo siguiente que hacemos es habilitar SSL y configurar el apartado referente a Postfix:

Fichero **/etc/dovecot/conf.d/10-master.conf**

Descomentamos lo siguiente:

```
inet_listener imap {
    port = 143
}
```

```
inet_listener imaps {
    port = 993
    ssl = yes
}
```

El puerto estándar no cifrado de IMAP es el 143 (en Gmail utiliza el cifrado STARTTLS), pero es recomendable utilizar el cifrado SSL, el cual corresponde con el puerto 993 (en Gmail utiliza el cifrado SSL).

Habilitamos el puerto 587, utilizado para la salida de los mensajes:

```
service submission-login {  
    inet_listener submission {  
        port = 587  
    }  
}
```

En el apartado de “lsmtp” cambiamos lo siguiente

```
unix_listener lsmtp {  
    #mode = 0666  
}
```

Por esto:

```
unix_listener lsmtp {  
    mode = 0600  
    user = postfix  
    group = postfix  
}
```

Y cambiamos el apartado de Postfix, por defecto está esto

```
# Postfix smtp-auth  
#unix_listener /var/spool/postfix/private/auth {  
# mode = 0666  
#}
```

Lo cambiamos por esto

```
# Postfix smtp-auth  
unix_listener /var/spool/postfix/private/auth {  
    mode = 0666  
    user = postfix  
    group = postfix  
}
```

Ahora configuraremos algunos ajustes de autenticación.

Fichero **/etc/dovecot/conf.d/10-auth.conf**

Descomentamos la siguiente línea:

```
disable_plaintext_auth = yes
```

Y cambiamos la siguiente línea:

```
auth_mechanisms = plain
```

Sustituir el “login”:

```
auth_mechanisms = plain login
```

También debemos comprobar que SSL está habilitado para que las conexiones sean cifradas. Gmail necesita tener esto habilitado para poder utilizar nuestro servidor de correos.

Fichero **/etc/dovecot/conf.d/10-ssl.conf**

```
ssl_cert = </etc/letsencrypt/live/nonuser.onthewifi.com/fullchain.pem  
ssl_key = </etc/letsencrypt/live/nonuser.onthewifi.com/privkey.pem  
ssl = required  
ssl_prefer_server_ciphers = yes  
ssl_min_protocol = TLSv1.2
```

Por último queda comprobar que el método de autenticación es PAM.

Fichero **/etc/dovecot/conf.d/auth-system.conf.ext**

Y comprobamos que en “passdb” esté así:

```
passdb {  
    driver = pam  
    # [session=yes] [setcred=yes] [failure_show_msg=yes] [max_requests=<n>]  
    # [cache_key=<key>] [<service name>]  
    #args = dovecot  
    args = %s  
}
```

Roundcube

Instalación

```
sudo apt install roundcube -y
```

Para que podamos acceder a la interfaz del correo tendremos que hacer un enlace simbólico del directorio original de Roundcube a la página web.

```
sudo ln -s /usr/share/roundcube/ /var/www/html/webmail
```

Una vez configurado todo lo anterior podremos mandar mensajes desde dentro del sistema a los usuarios locales al mismo, y hacia afuera; además gracias al mapeo de direcciones que hicimos en Postfix, podremos recibir correos desde fuera de la red.

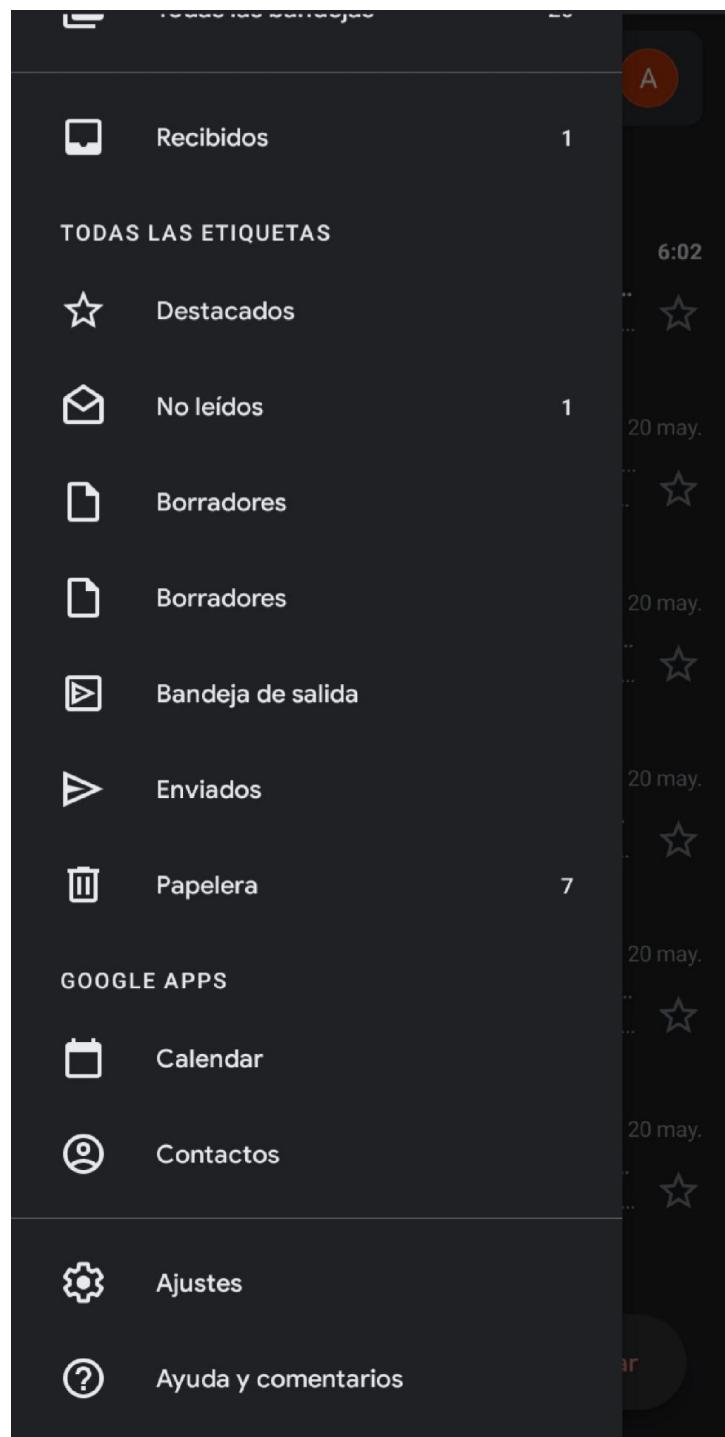
Uso con Gmail

Para usar el correo del usuario “admin” con Gmail debemos ir al apartado de nuestro móvil donde se encuentra el apartado de cuentas, en nuestro caso añadiremos una cuenta de correo personal IMAP.

En nombre de usuario pondremos “admin”, en la contraseña la propia del usuario (la del sistema) y en servidor pondremos nuestro dominio, en este caso “nonuser.onthewifi.com”.

A continuación se adjuntan capturas del proceso a seguir.

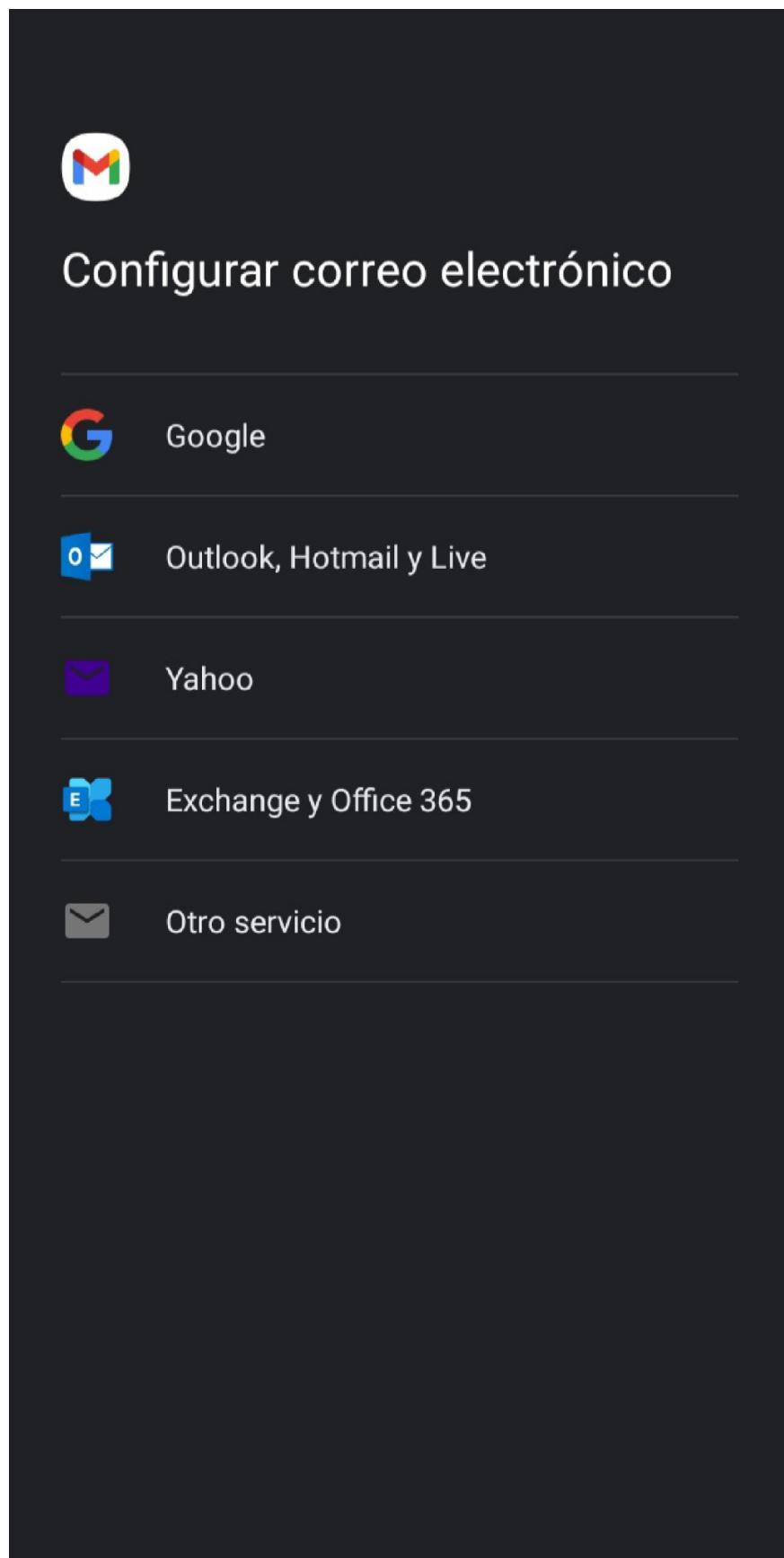
1º Paso: Buscar el apartado “ajustes” dentro del menú de Gmail.



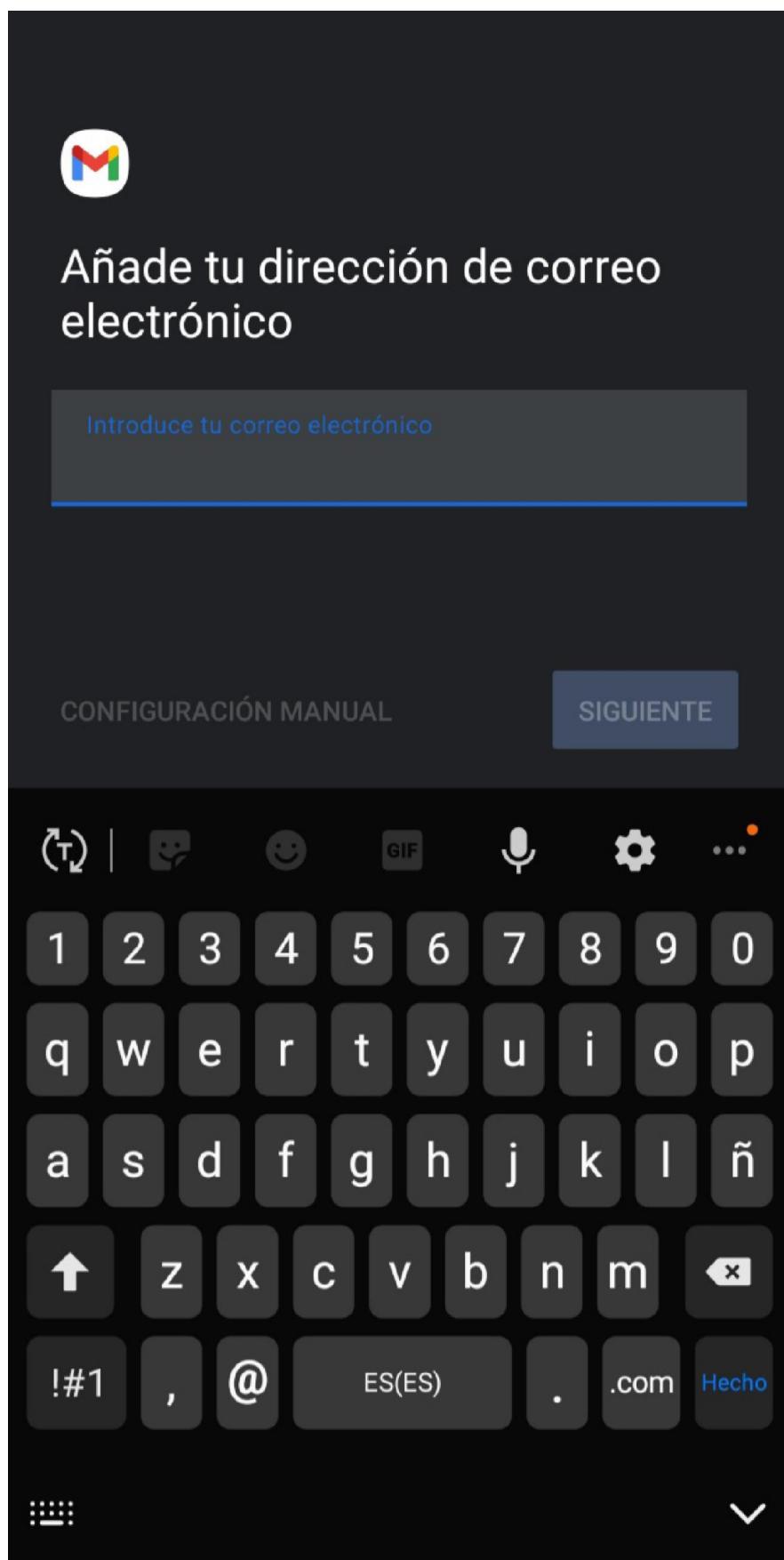
2º Paso: Seleccionar “Añadir cuenta”.



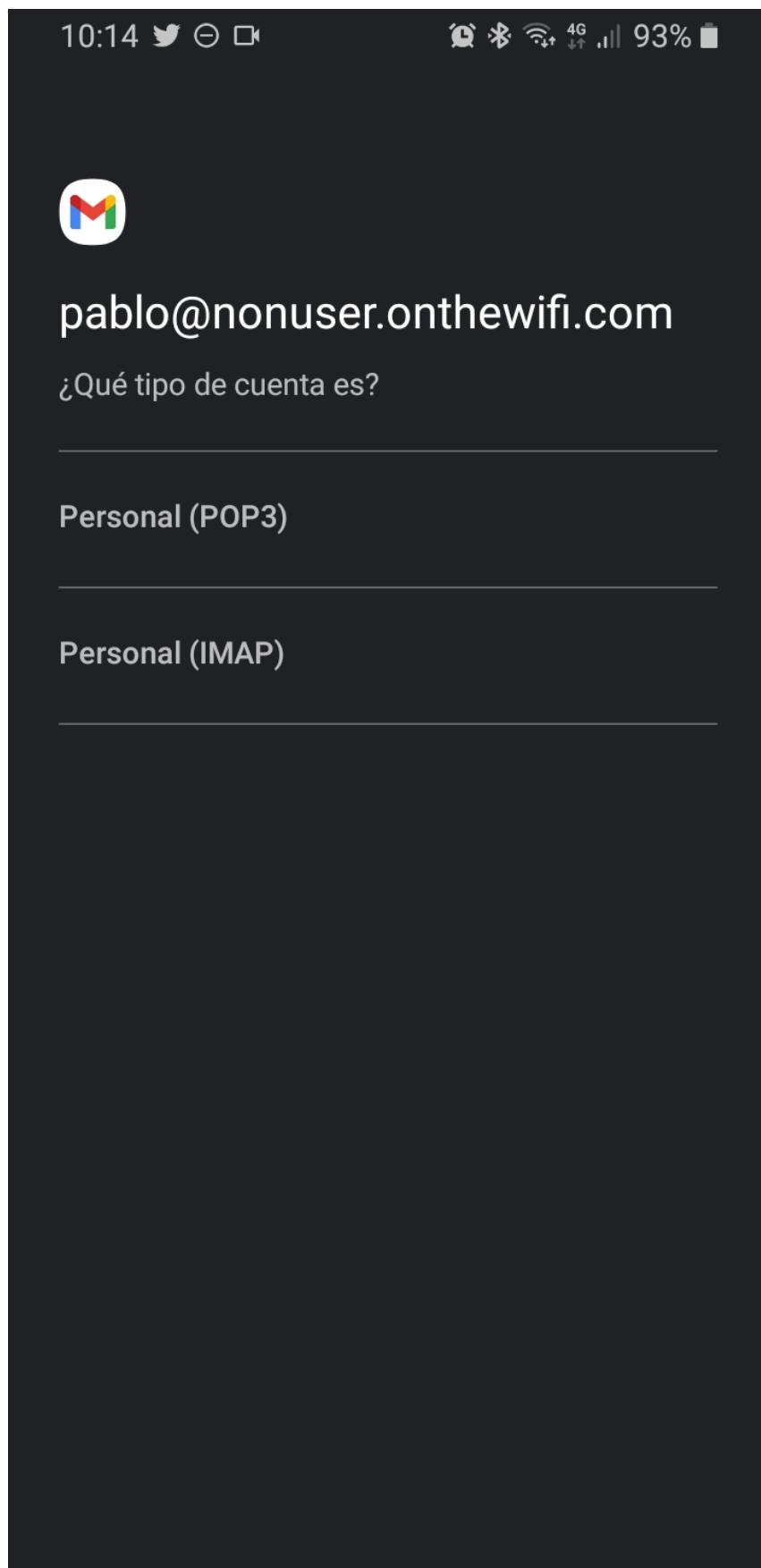
3º Paso: Seleccionar “Otro servicio”.



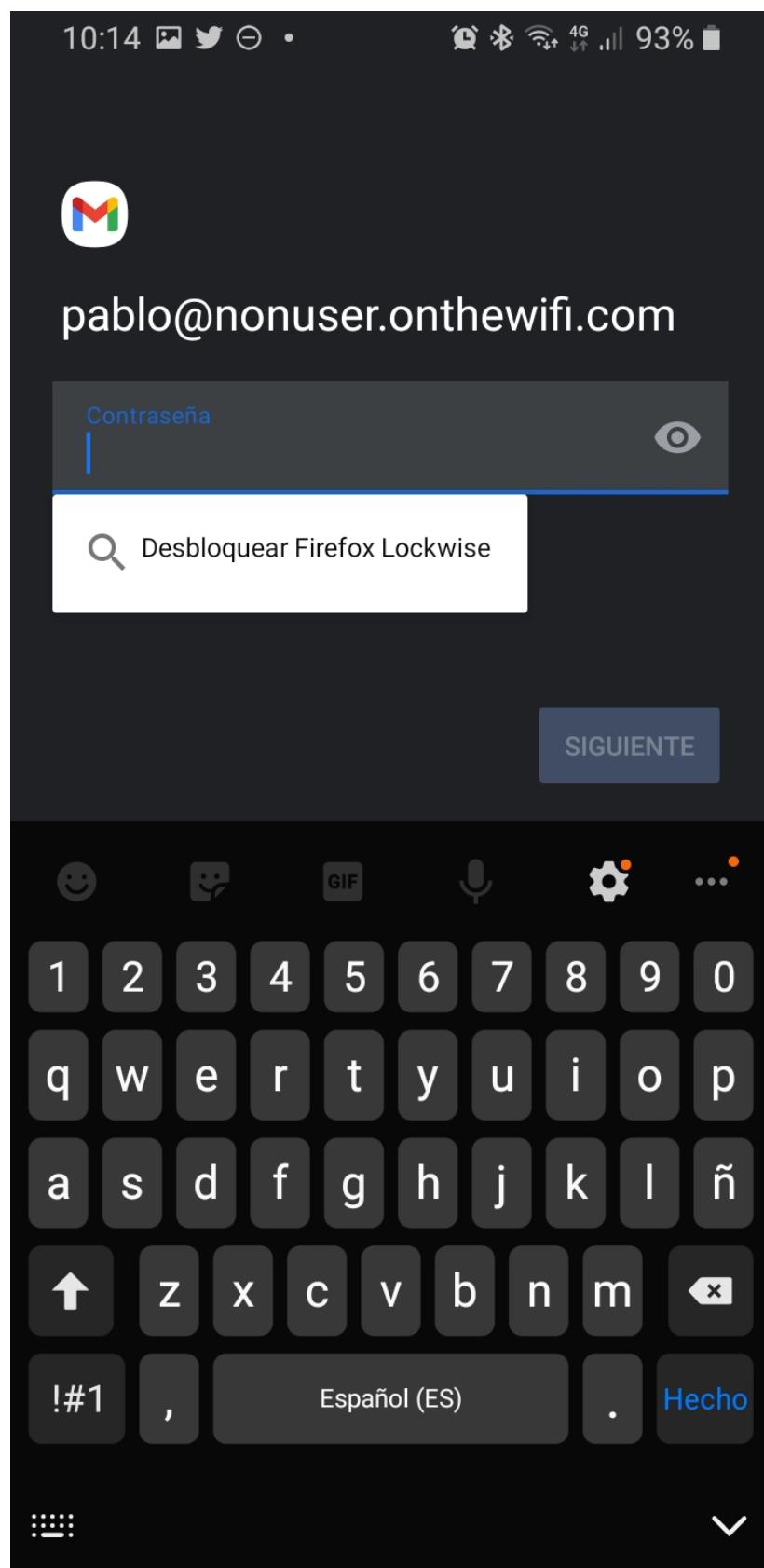
4º Paso: Introducir “nombreUsuario@dominio”, en este caso “pablo@nonuser.onthewifi.com”



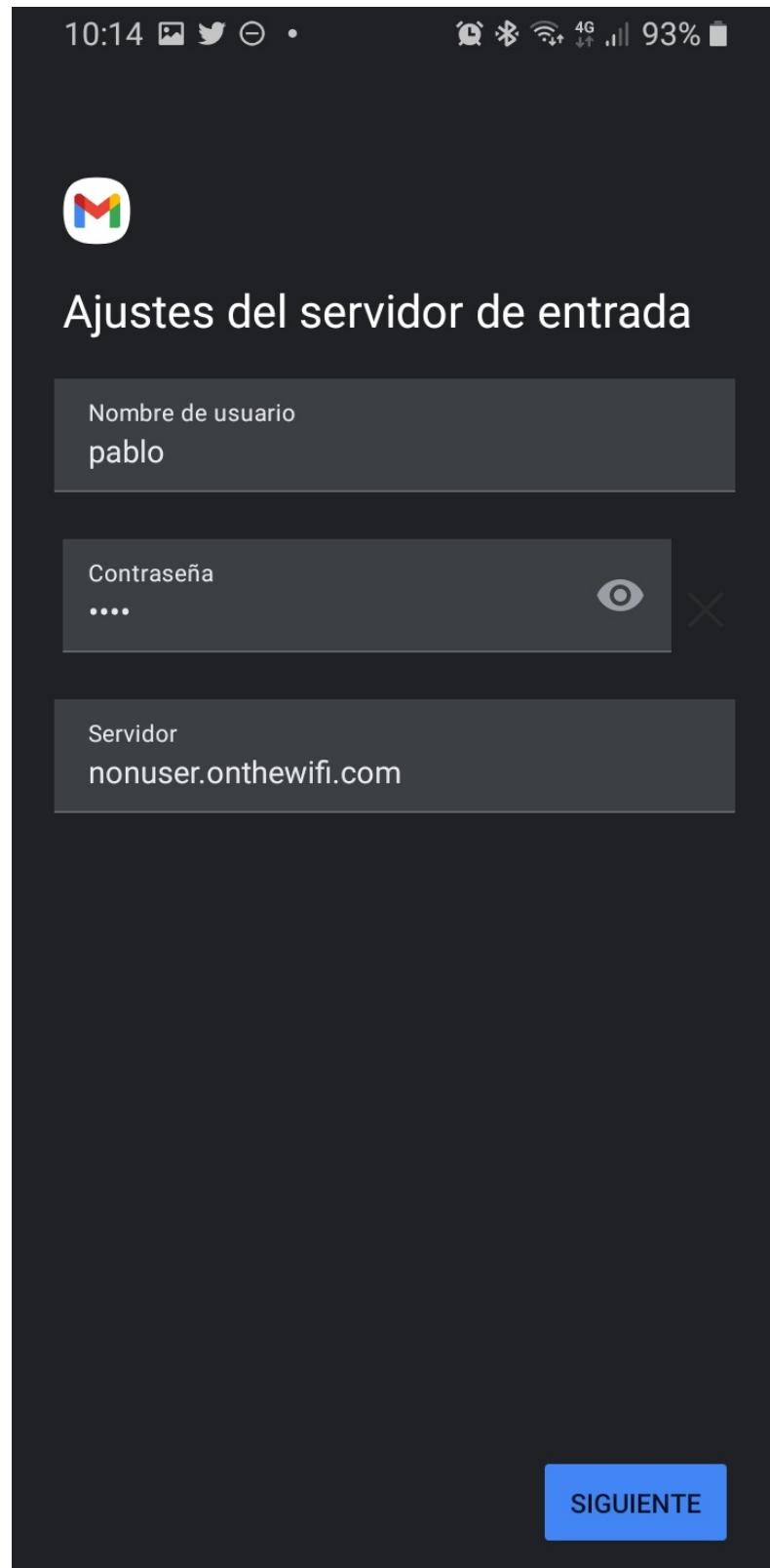
5º Paso: Seleccionar tipo de cuenta IMAP, pues así lo hemos configurado anteriormente.



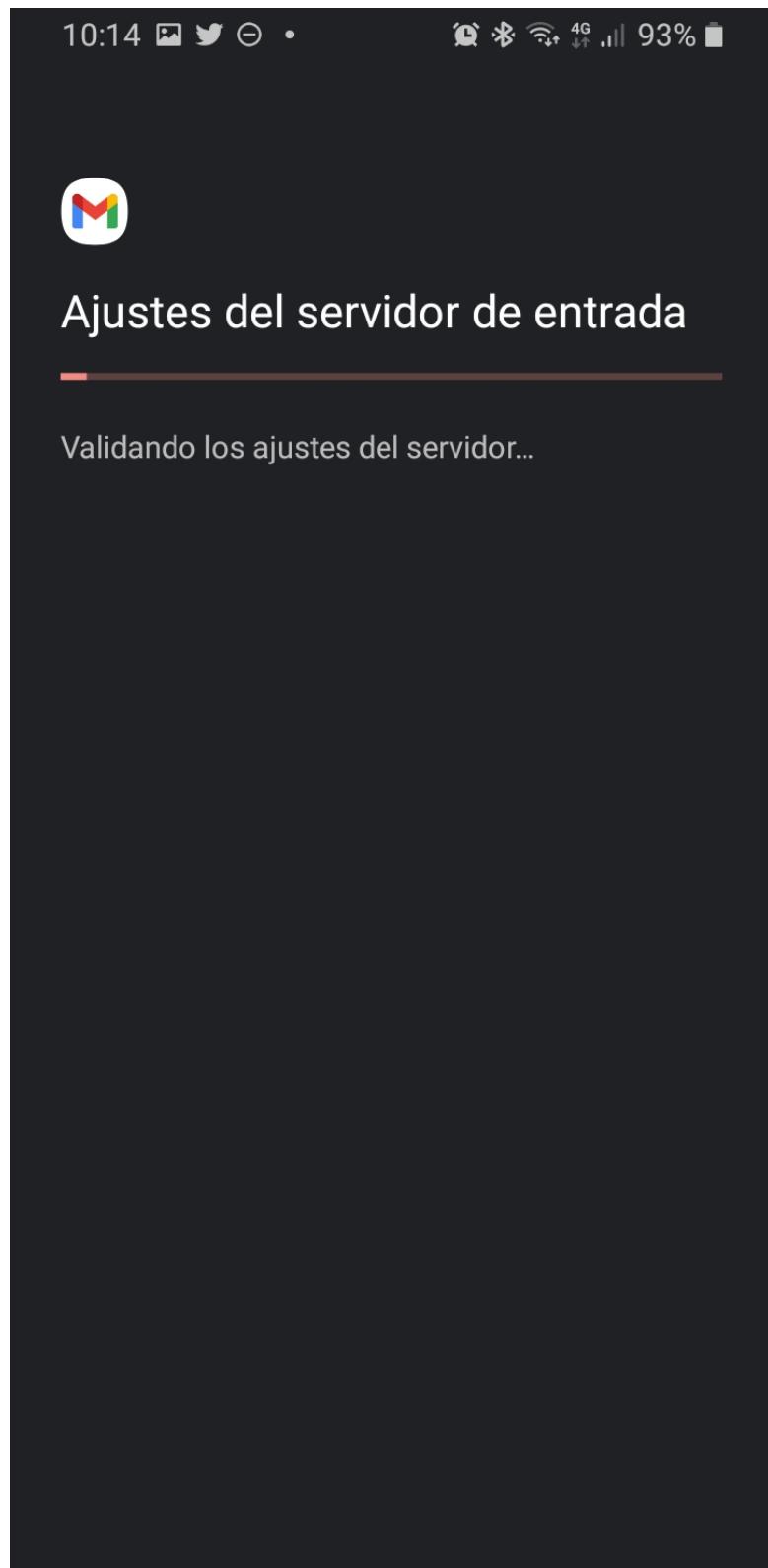
6º Paso: Introducir la contraseña del usuario.



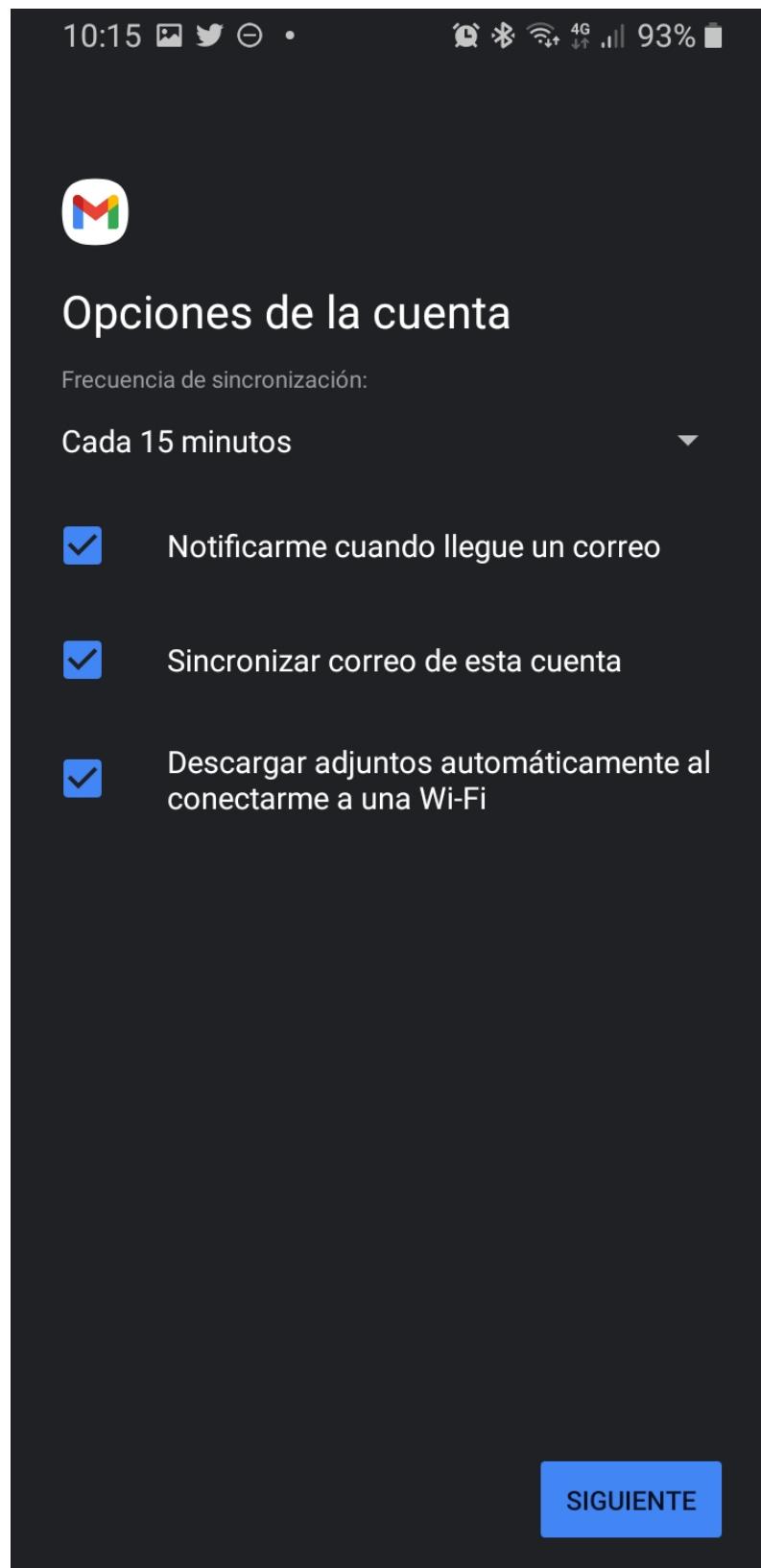
7º Paso: ¡MUY IMPORTANTE! Cambiar el nombre de usuario de “pablo@nonuser.onthewifi.com” a “pablo”. Nuestro servidor permite entrar únicamente con el nombre del usuario.



8º Paso: Una vez se haya clicado en “siguiente”, Gmail validará que los ajustes del servidor sean correctos. De no ser así, pedirá detalles más técnicos como pueda ser el puerto por el que te conectas y el tipo de cifrado que se utiliza, dependiendo de cómo elegimos POP3 o IMAP.



9º Paso: Ya prácticamente hemos terminado, se muestran unos ajustes para que el usuario seleccione lo que más le interese y una vez se haga clic en “siguiente”, la cuenta ya quedaría configurada y todos los correos que llegaran al servidor, llegarían también al dispositivo móvil.



Gestión del almacenamiento

Cuotas

Almacenamiento tope de 80 Mb.

Instalación de las cuotas

Lo primero que debemos hacer para instalar las cuotas en el sistema de ficheros es instalar las herramientas necesarias. Para ellos ejecutamos los siguientes comandos:

```
apt update -y && apt install quota -y
```

Una vez instalado debemos configurar el sistema de archivos, para ello nos tenemos que modificar el fichero `/etc/fstab` con el editor que queramos, en nuestro caso nano.

```
# The root file system has fs_passno=1 as per fstab(5) for automatic fsck.  
LABEL=RASPIROOT / ext4 rw 0 1  
# All other file systems have fs_passno=2 as per fstab(5) for automatic fsck.  
LABEL=RASPIFIRM /boot/firmware vfat rw 0 2
```

Nos aparecerá algo como lo que vemos y debemos dejar el archivo como el siguiente:

```
# The root file system has fs_passno=1 as per fstab(5) for automatic fsck.  
LABEL=RASPIROOT / ext4 rw,usrquota,grpquota 0 1  
# All other file systems have fs_passno=2 as per fstab(5) for automatic fsck.  
LABEL=RASPIFIRM /boot/firmware vfat rw 0 2
```

Vuelva a montar el sistema de archivos para que las nuevas opciones surtan efecto:

```
sudo mount -o remount /
```

Antes de activar finalmente el sistema de cuotas, debemos ejecutar manualmente el comando `quotacheck` una vez:

```
sudo quotacheck -gum /
```

Este comando crea los archivos `/aquota.user` y `/aquota.group`. Estos archivos contienen información sobre los límites y el uso del sistema de archivos, y deben existir antes de activar la supervisión de cuotas.

Ahora si queremos añadir una cuota de 80MB a cada usuario para que no pueda usar más de ese espacio, debemos añadir el siguiente comando en el servicio que utilizamos para crear los usuarios.

```
user="miUsuario"  
soft="50M"  
hard="80M"  
setquota -u $user $soft $hard 0 0 /
```

Para asignar estas quotas lo que hacemos es utilizar el comando anteriormente citado en un servicio llamado `dirlookupd` que hace que el usuario registrado se de alta prácticamente al instante.

Backups

Para poder realizar copias de seguridad de forma diaria hemos hecho uso de un cronjob y para realizarlas de forma automatizada hemos hecho uso de un script. La herramienta que se ha utilizado de cara a las copias de seguridad ha sido rsync.

Para que se ejecute diariamente hemos habilitado una entrada en el crontab que ejecute nuestro script todas las mañanas a las 4 de la mañana pues pensamos que era el horario con menos usuarios activos, y la copia sería incremental, de tal forma que sólo se guardarían los cambios realizados en el mismo día.

Para introducir la entrada en el crontab sin editar el correspondiente fichero hemos hecho uso del siguiente comando:

```
(crontab -l 2>/dev/null; echo "0 4 * * * /usr/bin/backup") | crontab -
```

Respecto a la propia copia de seguridad hemos empleado el script “backup”, que contiene los siguientes comandos:

```
#!/bin/bash

rsync -avzh /home/ /backups/
rsync -avzh /root/ /backups/

rsync -avzh /etc/ /backups/

rsync -avzh /var/www/ /backups/
rsync -avzh /lib/cgi-bin /backups/

rsync -avzh /lib/systemd/system/ /backups/
rsync -avzh /usr/bin/status /backups/
rsync -avzh /usr/bin/dirlookup /backups/
rsync -avzh /usr/bin/monitor /backups/
```

Para que funcione, le damos permisos de ejecución al script.

```
chmod +x /usr/bin/backup
```

Pensamos que en el caso de nuestro sistema, información relevante a salvaguardar sería (por orden de arriba hacia abajo):

- Los directorios de los usuarios
- El directorio de Root pues es donde hemos guardado Mumble en este caso
- Los archivos de configuración del sistema
- La página web
- Los scripts cgi-bin que gestionan la página web
- Los servicios que hemos creado y los correspondientes scripts que utilizamos

Todos estos ficheros se guardarán en el directorio “backups” que hemos creado con anterioridad en la raíz del sistema.

Para poder mandar estos ficheros a otro servidor bastaría con ejecutar rsync de la siguiente forma:

```
rsync -ravzh /etc/ username@remote_host:destination_directory
```

En nuestro caso no lo hemos hecho pues no disponíamos de un servidor al que mandar las copias de seguridad.

Seguridad

Fail2Ban

Instalamos Fail2Ban:

```
sudo apt install fail2ban -y
```

Para configurarlo es tan sencillo como copiar un fichero y modificarlo para gestionar cuánto tiempo estará baneada una IP, que nos mande un correo con datos de WHOIS y las líneas de log relevantes.

```
sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```

Y en ese fichero cambiamos ciertos valores para que sean los siguientes:

```
ignoreip = 127.0.0.1/8
bantime = 31536000
findtime = 86400
maxretry = 3
usedns = warn
destemail = admin@localhost
sendername = Fail2Ban
banaction = iptables-multiport
mta = sendmail
protocol = tcp
chain = INPUT
action = $(banaction_cf_mwl)s
```

Ignoramos nuestro propio localhost de tal forma que no nos banee, indicamos que a la persona baneada esté en la cárcel durante un año. Si esa persona hace 3 intentos erróneos en el mismo día se le banea. Una vez se banea a un usuario se nos manda un correo con los datos relevantes de una petición WHOIS a la IP del atacante y las líneas de log relevantes.

Luego en el directorio “jail.d” hay un archivo llamado “defaults-debian.conf” en el que incluimos el siguiente contenido para poder habilitar las cárceles:

```
[sshd]
enabled = true

[apache-auth]
enabled = true

[apache-badbots]
enabled = true

[apache-noscript]
enabled = true

[apache-overflows]
enabled = true

[apache-nohome]
enabled = true

[apache-botsearch]
enabled = true

[apache-fakegooglebot]
enabled = true

[apache-modsecurity]
enabled = true

[apache-shellshock]
enabled = true

[dovecot]
enabled = true

[postfix]
enabled = true

[postfix-rb]
enabled = true

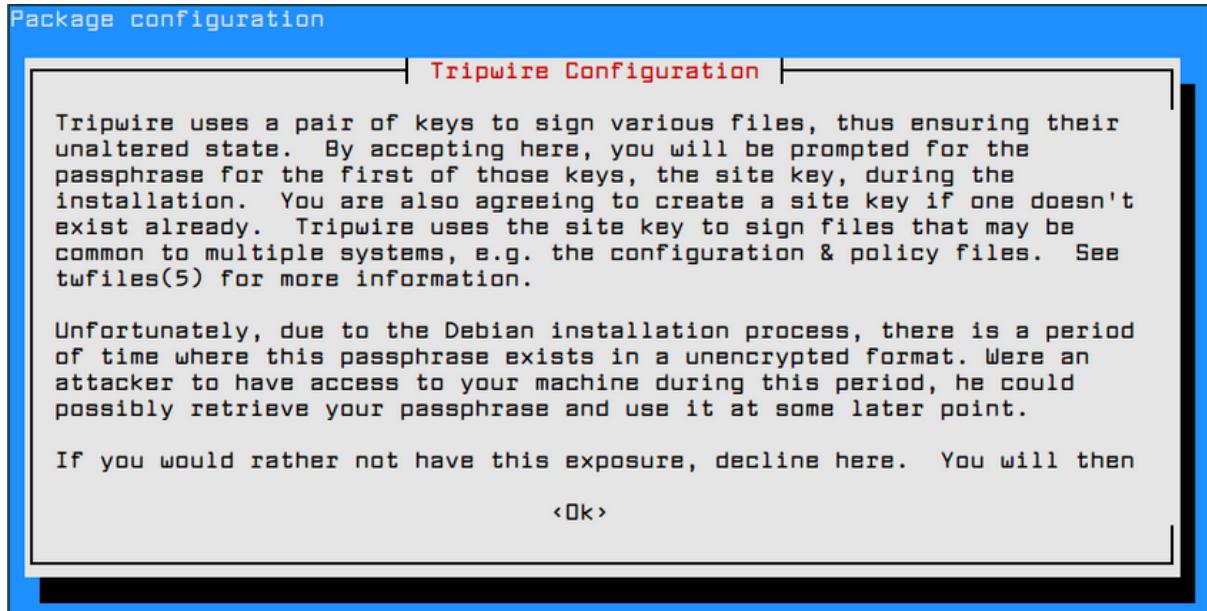
[postfix-sasl]
enabled = true

[roundcube-auth]
enabled = true
```

Tripwire

Tripwire es un software de seguridad que comprueba la integridad de los datos y monitorea y alerta sobre cambios en estos. Para instalarlo podemos ejecutar el siguiente comando.

```
apt install tripwire -y
```



Para habilitar tripwire y que así genere sus bases de datos y empiece a funcionar de continuo podemos ejecutar el siguiente comando:

```
tripwire --init
```

```
[root@centos ~]# tripwire --init
Please enter your local passphrase:
Parsing policy file: /etc/tripwire/tw.pol
Generating the database...
*** Processing Unix File System ***
### Warning: File system error.
### Filename: /usr/sbin/fixrmtab
### No such file or directory
### Continuing...
```

Una vez se ha configurado tripwire, se puede comprobar la modificación de ficheros con el siguiente comando:

```
tripwire --check
```

```
No existe el fichero o el directorio
19. File system error.
    Filename: /root/.addressbook.lu
    No existe el fichero o el directorio
20. File system error.
    Filename: /root/.addressbook
    No existe el fichero o el directorio
21. File system error.
    Filename: /root/.Xresources
    No existe el fichero o el directorio
22. File system error.
    Filename: /root/.Xauthority
    No existe el fichero o el directorio
23. File system error.
    Filename: /root/.ICEauthority
    No existe el fichero o el directorio
24. File system error.
    Filename: /proc/28524/fd/3
    No existe el fichero o el directorio
25. File system error.
    Filename: /proc/28524/fdinfo/3
    No existe el fichero o el directorio
26. File system error.
    Filename: /proc/28524/task/28524/fd/3
    No existe el fichero o el directorio
27. File system error.
    Filename: /proc/28524/task/28524/fdinfo/3
    No existe el fichero o el directorio
28. File system error.
    Filename: /proc/29369
    No existe el fichero o el directorio
-----  
Backups
Respecto a las copias de seguridad de forma diaria hemos hecho uso de un cronjob y para realizarlas de forma automatizada hemos hecho uso de un script. La herramienta que se ha utilizado para las copias de seguridad ha sido rsync.  
Respecto a las copias de seguridad de forma diaria hemos habilitado una entrada en el crontab que ejecute nuestro script todas las mañanas a las 4 de la mañana pues pensamos que era el horario con menos usuarios activos, y la copia seria incremental, de tal forma que sólo se guardarian los cambios realizados en el mismo día.  
Respecto a la propia copia de seguridad hemos empleado el script "backup", que contiene los siguientes comandos:  
-----  
*** End of report ***  
  
Open Source Tripwire 2.4 Portions copyright 2000-2018 Tripwire, Inc. Tripwire is a registered trademark of Tripwire, Inc. This software comes with ABSOLUTELY NO WARRANTY; for details use --version. This is free software which may be redistributed or modified only under certain conditions; see COPYING for details.  
All rights reserved.  
Integrity check complete.  
root@piratebay:~#
```

En la imagen superior se muestra parte del informe de Tripwire.

Este informe es tan extenso, que se recomienda redirigir la salida por pantalla (STDOUT) a un fichero para poder manejar y tratar mejor la información que nos brinda.

Esto se podría hacer con el siguiente comando.

```
tripwire --check > informe.txt
```

Monitorización

Rsyslog

Ya que en la práctica se nos pedía que guardamos un log con los intentos fallidos y exitosos de accesos y lo gestionaremos con Rsyslog hicimos la siguiente función que tenemos dentro del InstallServer.sh.

```
# Logs de Apache
echo 'module(load="imfile" PollingInterval="10")
input(type="imfile"
      File="/var/log/apache2/access.log"
      Tag="[APACHE]"
      Severity="error"
      Facility="local6")
local6.error          /home/admin/access.log' >
/etc/rsyslog.d/02-apache.conf

# Logs de correos
echo "mail.*      -/home/admin/access.log" >> /etc/rsyslog.conf

# Logs de autenticación por PAM (SSH, SFTP y Apache)
echo "auth,authpriv.*           -/home/admin/access.log" >>
/etc/rsyslog.conf

systemctl restart rsyslog
}'
```

Mediante estas función y gracias al servicio de Rsyslog lo que hacemos es guardar los accesos de apache2, los de el webmail y todos las autenticaciones relacionadas con los servicios de SSH, STFP y Apache.

De esta forma proporcionamos un fichero “.log” en la carpeta home del admin que le permite conocer todos aquellos accesos que se han producido dentro de su servidor.

Web de status



La imagen superior refiere al estado de los servicios del servidor, a esta página se puede acceder en la dirección “nonuser.onthewifi.com/cgi-bin/status.cgi”.

Para la gestión de la página web se hace uso del script “status.cgi” y para la monitorización de los servicios se hace uso de un script que se ejecuta cada 15 minutos mediante un cronjob.

El script que monitoriza los servicios es el siguiente (“servicios/status”). Este script se ejecuta como Root.

```
#!/bin/bash

# Variables
dir="/var/www/status/"
declare -a services=("apache2" "sshd" "postfix" "dovecot" "mariadb" "quotaon" "dirlookupd"
"statusd" "mumble")

for service in "${services[@]}"
do
    (systemctl -q is-active $service && echo 1 || echo 0) > $dir$service
done
```

Básicamente lo que hace es iterar en la lista de servicios preguntando a systemd si está activo, de ser así mete un 1 en el fichero correspondiente a su nombre, si no es así introduce un 0.

Para incluirlo en el crontab, podemos editar el con “crontab -e” o automatizarlo con el siguiente comando:

```
(crontab -l 2>/dev/null; echo "*/15 * * * * /usr/bin/status") | crontab -
```

El status se muestra en la página mediante el siguiente fragmento de código (“cgi-bin/status.cgi”):

```
#!/usr/bin/perl -w
use warnings;
use CGI;
use File::Slurp;
use Data::Dumper::Simple;
use strict;
use utf8;

my $q = CGI->new;

my $apache = read_file('/var/www/status/apache2');
my $sshd = read_file('/var/www/status/sshd');
my $postfix = read_file('/var/www/status/postfix');
my $dovecot = read_file('/var/www/status/dovecot');
my $mariadb = read_file('/var/www/status/mariadb');
my $mumble = read_file('/var/www/status/mumble');

my @services = ($apache, $sshd, $sshd, $postfix, $dovecot, $mariadb, $mumble);
my @servicios = ("Web", "SSH", "SFTP", "SMTP", "IMAP", "Database", "VOIP");
my $i = 7;
...
for ($i = 0; $i < @services; $i++){
    my $servicio = $servicios[$i];
    my $label = 'label-success';
    my $label2 = 'Operational';
    if ($services[$i] != 1){
        $label = 'label-danger';
        $label2 = 'Not Operational';
    }
    print qq(
<div class="list-group-item">
    <h4 class="list-group-item-heading">
        $servicio service
        <a href="#" data-toggle="tooltip" data-placement="bottom" title="$servicio">
            <i class="fa fa-question-circle"></i>
        </a>
    </h4>
    <p class="list-group-item-text">
        <span class="label $label">$label2</span>
    </p>
</div>
    );
}
}
```

Se itera en los servicios comprobando el valor de los ficheros, si el fichero contiene un 1, se imprime como operacional, si contiene un 0 como no operacional.

Envío de correos al administrador

Se ha programado un envío periódico de información sobre el sistema al administrador mediante correo electrónico. El script que recoge la información y manda el correo a su vez es el siguiente:

```
#!/usr/bin/perl -w

use CGI;
use warnings;
use strict;
use Filesys::DiskUsage qw/du/;
use Proc::ProcessTable;

use SQL::Abstract;
use DBI;

use Email::MIME;
use Email::Sender::Simple qw(sendmail);
use MIME::Words qw(:all);

# Base datos
my $usuarioDB = "root";
my $claveDB = "admin";
my $DB = "usuarios";
my $tabla = 'datos';

# Gestión base datos
my $sql = SQL::Abstract->new;
my $dbh = DBI->connect("DBI:MySQL:$DB:localhost", $usuarioDB, $claveDB) or print "\nError al abrir la
base de datos.\n";

my($stmt, @bind) = $sql->select($tabla, 'usuario');
my $sth = $dbh->prepare($stmt);
$sth->execute(@bind);

# Creamos el fichero
my $file = "/tmp/informe.txt";
open(my $fp, '>', $file) or die "Error al abrir el archivo $file";

# Información sobre el uso del disco
print $fp "\n#####\n# Uso del disco #\n#####\n";
my @usuarios;
while (@usuarios = $sth->fetchrow_array){
    my $usuario = $usuarios[0];
    print $fp "Nombre del usuario: ";
    printf $fp "$usuario\n";
    my $dir = "/home/" . $usuario . "/";
    my $tam = du ( { 'human-readable' => 1 } , $dir );
    printf $fp "%-20s %-20s" . "\n", "Espacio usado", "Directorio";
    printf $fp "%-20s %-20s" . "\n", "$tam", "$dir";
}
# Desconexión de la base de datos
$dbh->disconnect or warn "\nFallo al desconectar.\n";

# Información sobre los procesos
```

```

print $fp "\n#####\n# Procesos #\n#####\n\n";
my $t = new Proc::ProcessTable;
my $p;
my $linea;
printf $fp '%-6s %-10s %-10s %-10s %-50s' . "\n", "PID", "STAT", "%MEM", "CPU", "COMMAND";
foreach $p ( @{$t->table} ){
    printf $fp '%-6d %-10s %-10f %-10f %-50s' .
    "\n", $p->pid, $p->state, $p->pctmem, $p->pctcpu, $p->cmndline;
}

# Información de memoria
print $fp "\n#####\n# Informacion de memoria
#\n#####\n\n";

my $datosmeminfo = "";
open(my $fh, '<', "/proc/meminfo");
{
    local $/;
    $datosmeminfo = <$fh>;
}
close($fh);
print $fp $datosmeminfo;
close($fp);

my $body;
my $filename = "/tmp/informe.txt";
open($fh, '<', $filename) or die "cannot open file $filename";
{
    local $/;
    $body = <$fh>;
}
close($fh);

# Enviar correo al administrador
my $message = Email::MIME->create(
    header_str => [
        From => "'The Pirate Bay' <admin@nonuser.onthewifi.com>",
        To => 'admin@nonuser.onthewifi.com',
        Subject => "Informe de monitorizacion",
    ],
    attributes => {
        content_type => 'text/plain',
        encoding => 'base64',
        charset => 'UTF-8',
    },
    body_str => $body,
);
sendmail($message);

```

Recoge datos del uso de disco de los usuarios (para ello recoge la lista de usuarios de la base de datos), información sobre los procesos e información sobre la memoria, y la manda por correo al administrador “admin@nonuser.onthewifi.com”.

El script “monitor” se almacena en el directorio “/usr/bin” con permisos de ejecución.

Para permitir que se ejecute periódicamente hemos habilitado una entrada en crontab de tal forma que se manda a las 8 de la mañana para que el administrador disponga de información sobre el estado del servidor nada más levantarse.

Para poder gestionar la entrada del crontab desde un script podemos usar el siguiente comando:

```
(crontab -l 2>/dev/null; echo "0 8 * * * /usr/bin/monitor") | crontab -
```

Panel de administración

The screenshot shows the administrative interface for The Pirate Bay. At the top, there's a dark header bar with "THE PIRATE BAY" on the left and "Ajustes", "Ayuda", and "Cerrar Sesión" on the right. Below the header, the main content area has a title "BIENVENIDO ADMIN". A search bar labeled "Usuario" and a red "Eliminar cuenta" button are visible. There's also a blue "Refrescar estadísticas de uso" button. Three main sections are present: "Disk Usage per User", "Process Information", and "Memory Information", each with a "Open Here!" link.

Desde el panel de administración se permite eliminar usuarios y revisar la misma información que se manda en el reporte periódico por correo. Podríamos haber implementado más funcionalidades como la activación/desactivación de servicios, la edición de datos de usuarios o el bloquear una cuenta en específico con algo más de tiempo.

La información que se muestra en pantalla se gestiona en el script “cgi-bin/admin.cgi”, el cual es una mezcla del script que elimina usuarios y del script que monitoriza el servidor.

Wordpress

Para la solicitud automática de Blogs, hemos utilizado la herramienta de Wordpress con la que cualquier usuario puede solicitar la creación automática de un Blog.

Para instalar el software debemos seguir los siguientes pasos:

- 1) Descargar el paquete comprimido de WordPress con el siguiente comando:

```
wget https://es.wordpress.org/latest-es_ES.tar.gz
```

- 2) Ahora descomprimimos el paquete en /var/www/html

```
sudo tar xf latest-es_ES.tar.gz -C /var/www/html
```

- 3) El siguiente paso será cambiar los permisos de la carpeta para que le pertenezca a www-data

```
sudo chown www-data:www-data /var/www/html/wordpress/ -R
```

- 4) Lo siguiente que debemos hacer es proceder a la instalación de WordPress a través de la página web:

Debemos seguir los pasos que nos va indicando WordPress y rellenamos los campos de información para el usuario y la base de datos.





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="wordpressdb"/>	El nombre de la base de datos que quieres usar con WordPress.
Nombre de usuario	<input type="text" value="adminwp"/>	El nombre de usuario de tu base de datos.
Contraseña	<input type="password" value=""/>	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=""/>	Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.



No se ha podido escribir en el archivo wp-config.php.

Puedes crear manualmente el archivo wp-config.php y pegar el siguiente texto.

```
<?php  
/**  
 * The base configuration for WordPress  
 *  
 * The wp-config.php creation script uses this file during the  
 * installation. You don't have to use the web site, you can  
 * copy this file to "wp-config.php" and fill in the values.  
 *  
 * This file contains the following configurations:  
 *  
 * * MySQL settings  
 * * Secret keys  
 * * Database table prefix  
 * * ABSPATH  
 */
```

Después de hacer esto, haz clic en «Ejecutar la instalación».

Bienvenido al famoso proceso de instalación de WordPress en cinco minutos! Simplemente completa la información siguiente y estarás a punto de usar la más enriquecedora y potente plataforma de publicación personal del mundo.

Información necesaria

Por favor, debes facilitarnos los siguientes datos. No te preocupes, siempre podrás cambiar estos ajustes más tarde.

Título del sitio

Nombre de usuario
Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.

Contraseña (JS%eWCW0g^ZV^6UI@f) Ocultar

Título del sitio The Pirate Bay

Nombre de usuario admin
Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.

Contraseña Débil Show

Importante: Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.

Confirma la contraseña Confirma el uso de una contraseña débil.

Tu correo electrónico admin@nonuser.onthewifi.com
Comprueba bien tu dirección de correo electrónico antes de continuar.

Visibilidad en los motores de búsqueda Disuadir a los motores de búsqueda de indexar este sitio
Depende de los motores de búsqueda atender esta petición o no.

Instalar WordPress

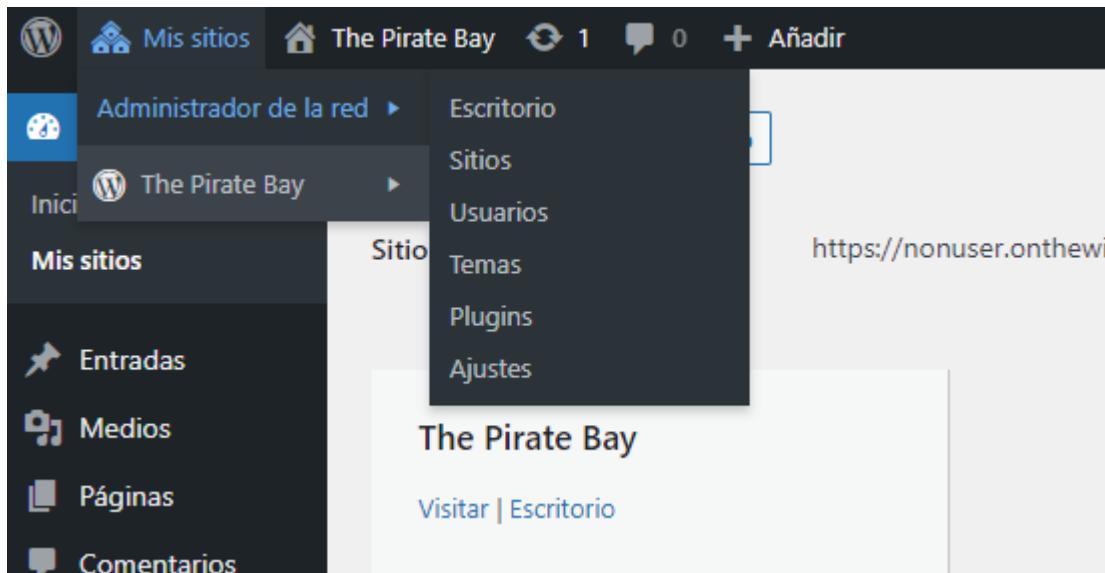
Una vez hemos hecho todo lo que se nos ha pedido y hemos llenado todos los campos le damos a Instalar WordPress y ya tendríamos WordPress funcional.

Ahora tenemos que configurar la segunda parte que es que WordPress permita a cada usuario solicitar su Sitio, para ello deberemos configurar correctamente Multisite en el panel de configuración y el archivo de configuración wp-config.php.

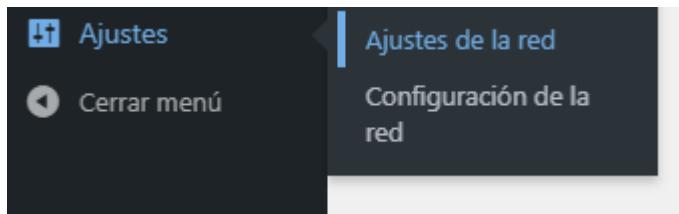
En el archivo de configuración añadiremos la siguiente línea para poder configurar Multisite en WordPress:

```
define('MULTISITE', true);
```

Para lo siguiente debemos ir al panel de configuración web de WordPress y a Administrador de la red.



Luego tenemos que dirigirnos a “Ajustes>Ajustes de la red”.



Una vez estemos allí debemos permitir que los usuarios puedan registrar sitios.

The screenshot shows the 'Ajustes de la red' (Network Settings) page. Under the 'Permitir nuevos registros' (Allow new registrations) heading, there are four radio button options:

- El registro está desactivado (Registration is disabled)
- Se pueden crear cuentas de usuario. (User accounts can be created.)
- Los usuarios conectados pueden registrar sitios nuevos (Connected users can register new sites)
- Pueden registrarse sitios y cuentas de usuario (Sites and user accounts can be registered)

A small note below states: Si el registro de usuarios está desactivado, establece como valor de NOBLOGREDIRECT en wp-config.php una URL donde redirigir a los visitantes que accedan a un sitio inexistente. (If user registration is disabled, set the value of NOBLOGREDIRECT in wp-config.php to a URL where to redirect visitors who access to a non-existent site.)

Y con esto ya queda configurado WordPress Multisite.

Ahora para que los usuarios puedan pedir su sitio automático lo que hemos hecho es que desde su dashboard redirigirlos a la página de solicitud de un sitio, como mostramos a continuación.

BIENVENIDO SERGIO

Servidor de correos

Accede a tu correo de forma fácil y rápida

[Continuar](#)

Almacenamiento para Webs

Crea un espacio para tu web
(Almacenamiento 5MB)Conéctate mediante SFTP con el
siguiente comando:

sftp -P 2222

sergio@nonuser.onthewifi.com

 Creación de Blogs con
WordPressCrea tus blogs personalizados y de forma
rápida con WordPress[Crear](#)

Servicio de VOIP Mumble

Hable con sus amigos fácilmente con
nuestro servicio de voz Mumble.

Dirección

Puerto

nonuser.onthewifi.com

64738

[Descargar](#)Copyright © 2021, The Pirate Bay
All rights reserved.

Consigue tu propia cuenta en The Pirate Bay en segundos.

Nombre de usuario:

(Deben tener como mínimo 4 caracteres, solo letras y números.)

Dirección de correo electrónico:

Envaremos los datos de registro a esta dirección de correo electrónico. Comprueba bien esta dirección antes de continuar.

 ¡Dame un sitio! Solo un nombre de usuario, por favor.[Siguiente](#)

Una vez le han puesto un nombre y un correo se les pedirá un nombre para su sitio, una vez lo tenga solo tendrán que activarlo para poder disfrutar de la magia de WordPress.

Mumble

Para esta práctica debido a que estamos usando una arquitectura arm64v8 no podíamos usar teamspeak, por lo que buscamos algo similar que pudiéramos usar. Primero probamos con el virtualizador Box64 pero no conseguimos buenos resultados con Teamspeak.

Por lo que pasamos a usar Mumble que es un servicio VOIP algo más simple que Teamspeak pero con prácticamente las mismas funcionalidades. Como seguimos con el mismo problema de que usamos una arquitectura arm64 tuvimos que usar docker porque con Box64 seguimos teniendo problemas. La ventaja que nos ofrecía docker es que nos permitía instalar Mumble sin apenas problemas y de forma sencilla, esto es debido a que docker proporciona contenedores independientes que se ejecutan dentro de una sola instancia de Linux.

Para la instalación y configuración de Docker y Mumble debemos seguir los siguientes pasos:

- 1) Descargamos el archivo de docker mediante el comando curl

```
curl -sSL https://get.docker.com/ | sh
```

- 2) Instalamos Docker (versión no gráfica) con el siguiente comando:

```
apt-get install docker-ce docker-ce-cli containerd.io -y
```

- 3) Tras instalar docker ya podemos usar sus comandos y ejecutaremos el siguiente comando para descargar e instalar la imagen de Mumble

```
docker run -d --name mumble -p 64738:64738 -p 64738:64738/udp ugeek/mumble:arm
```

- 4) Una vez hemos hecho esto ya tenemos instalado tanto Docker como Mumble ahora solo debemos configurar Mumble. Para ello ejecutamos el siguiente comando que nos permite entrar en una especie de terminal de docker.

```
docker exec -i -t --user root mumble sh
```

- 5) Ahora tendríamos que editar el fichero de configuración de Mumble y en nuestro caso docker por defecto solo cuenta con el editor de texto vi pero aparte de ese editor nosotros instalamos el editor de texto plano nano. Para hacer todo esto solo debemos ejecutar los siguientes comandos:

```
apk add nano  
nano /config/mumble-server.ini
```

- 6) Podemos configurar diferentes aspectos como la contraseña del server, el nombre de la sala, el máximo de usuarios, el puerto y la ip... En nuestro caso solo cambiamos el mensaje de bienvenida y el nombre de la sala.

```
# Password to join server.  
serverpassword=contraseña  
# The below will be used as defaults for new configured servers.  
# If you're just running one server (the default), it's easier to  
# configure it here than through D-Bus or lce.  
#  
# Welcome message sent to clients when they connect.  
welcometext=<br/>Bienvenido al servidor VOIP de The Pirate Bay  
<b>Arrrrrr</b><br/>Arrasa con lo que veas y generoso no seas!<br />"  
# To enable public server registration, the serverpassword must be blank, and  
# this must all be filled out.  
# The password here is used to create a registry for the server name; subsequent  
# updates will need the same password. Don't lose your password.  
# The URL is your own website, and only set the registerHostname for static IP  
# addresses.  
# Only uncomment the 'registerName' parameter if you wish to give your "Root"  
# channel a custom name.  
#  
registerName=The Pirate Bay
```

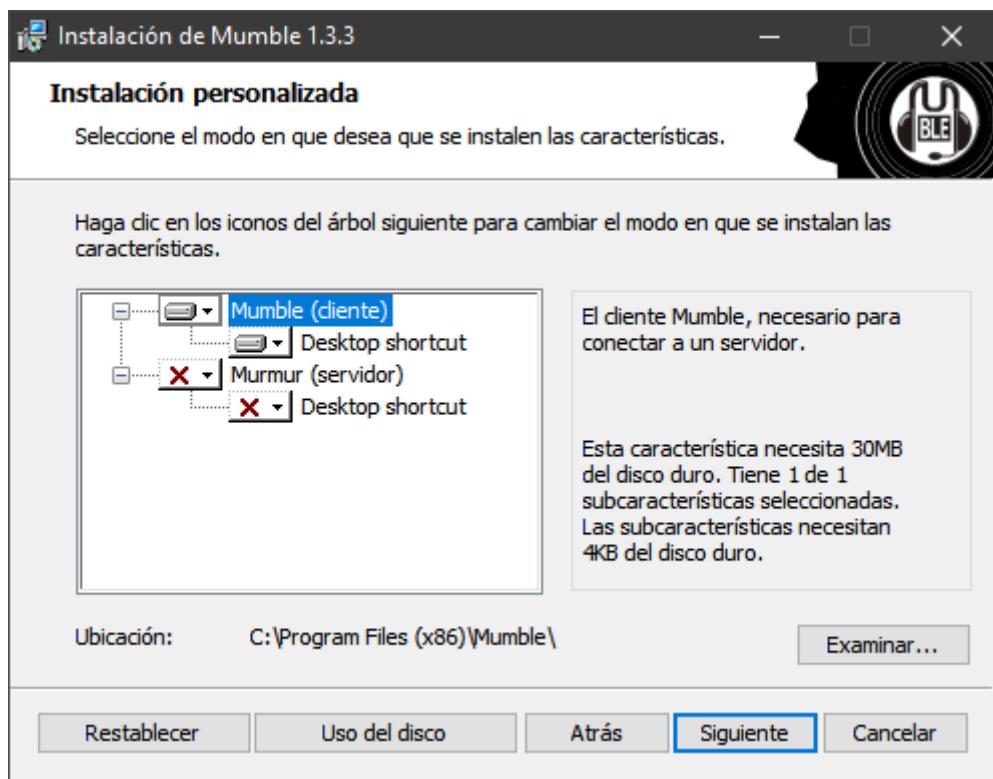
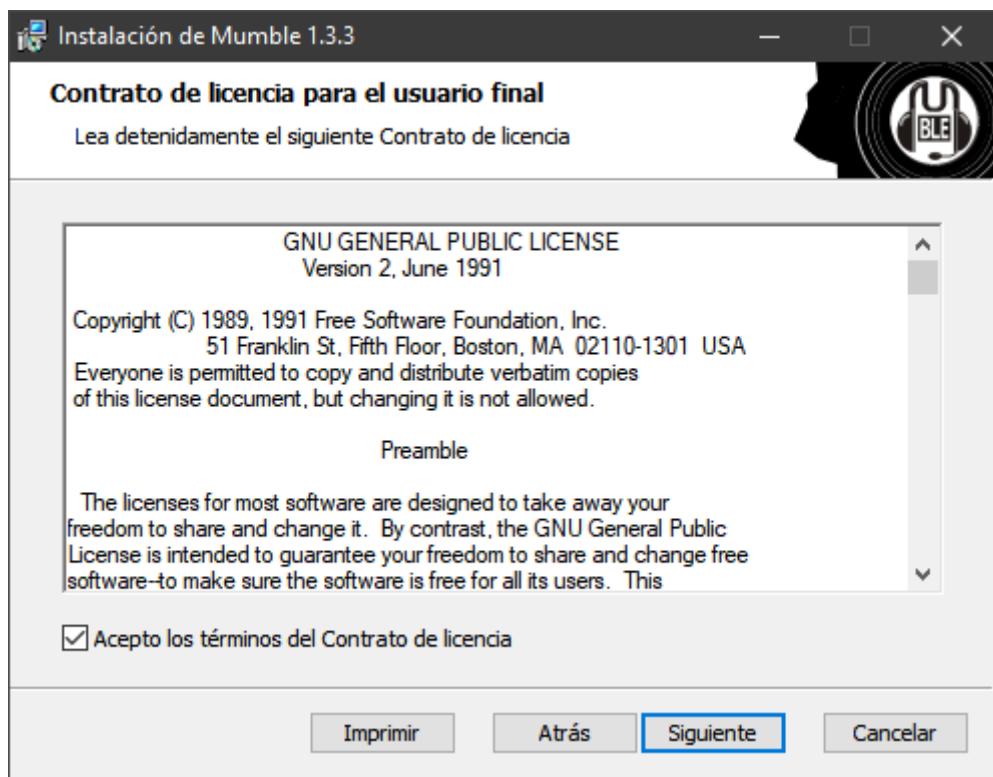
- 7) Ya tenemos todo listo para empezar a hablar con nuestros amigos por un servidor de VOIP propio.

Para la configuración del cliente de Mumble debemos irnos a la página oficial (<https://www.mumble.com/mumble-download.php>) y descargar el cliente que sea compatible con el sistema operativo que tengamos.

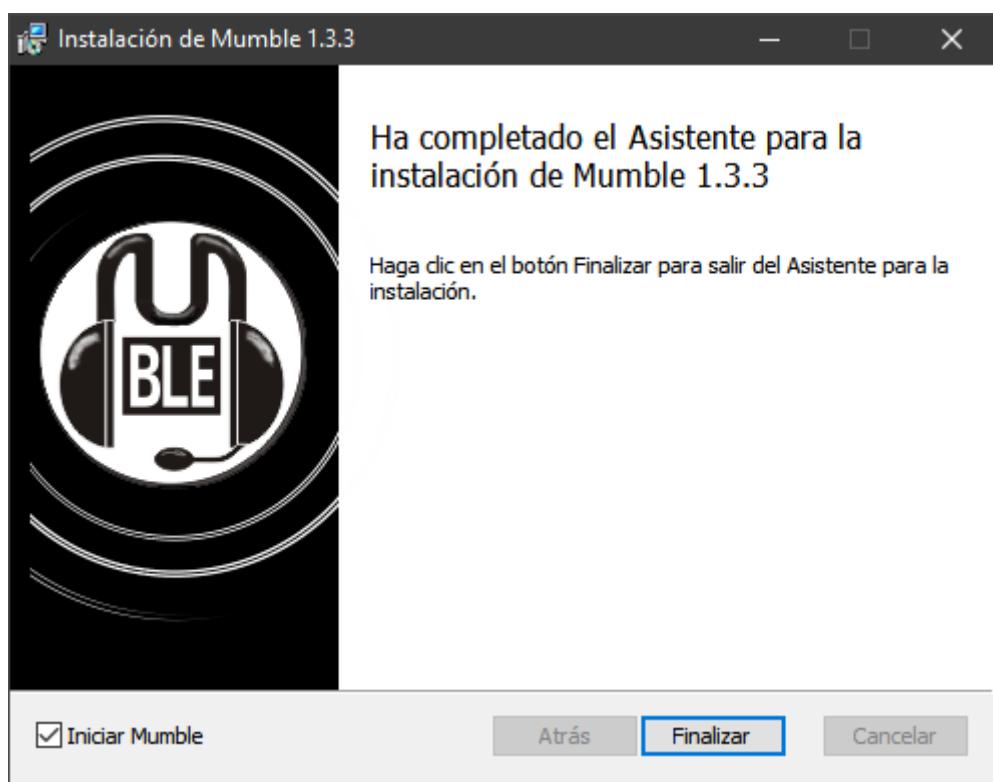
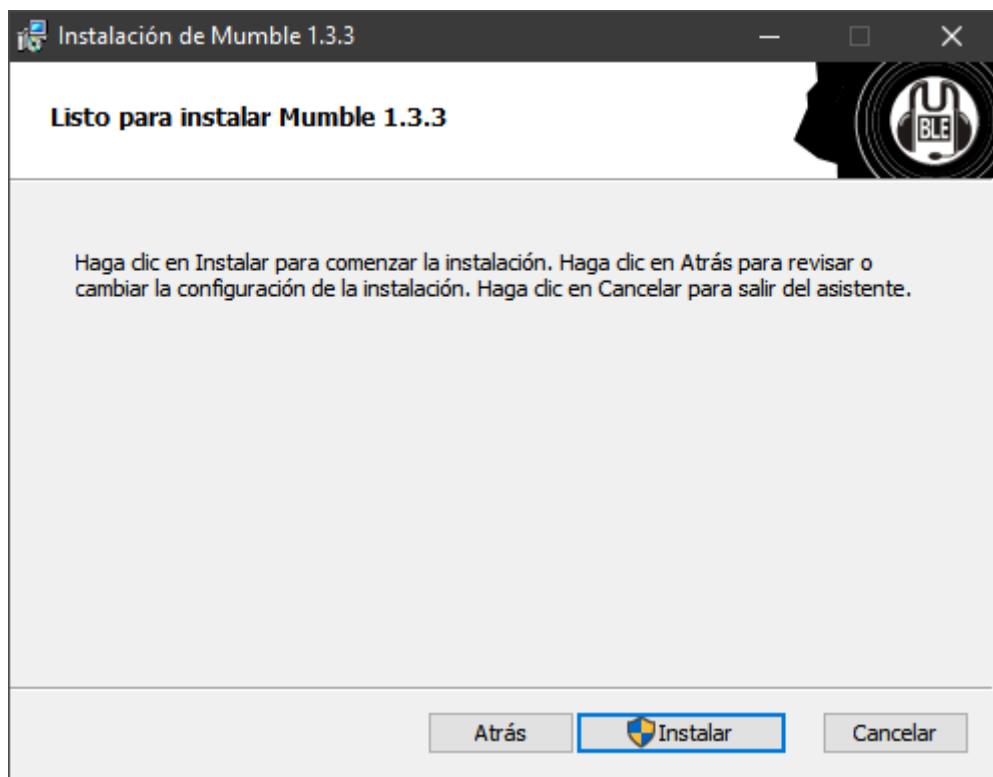


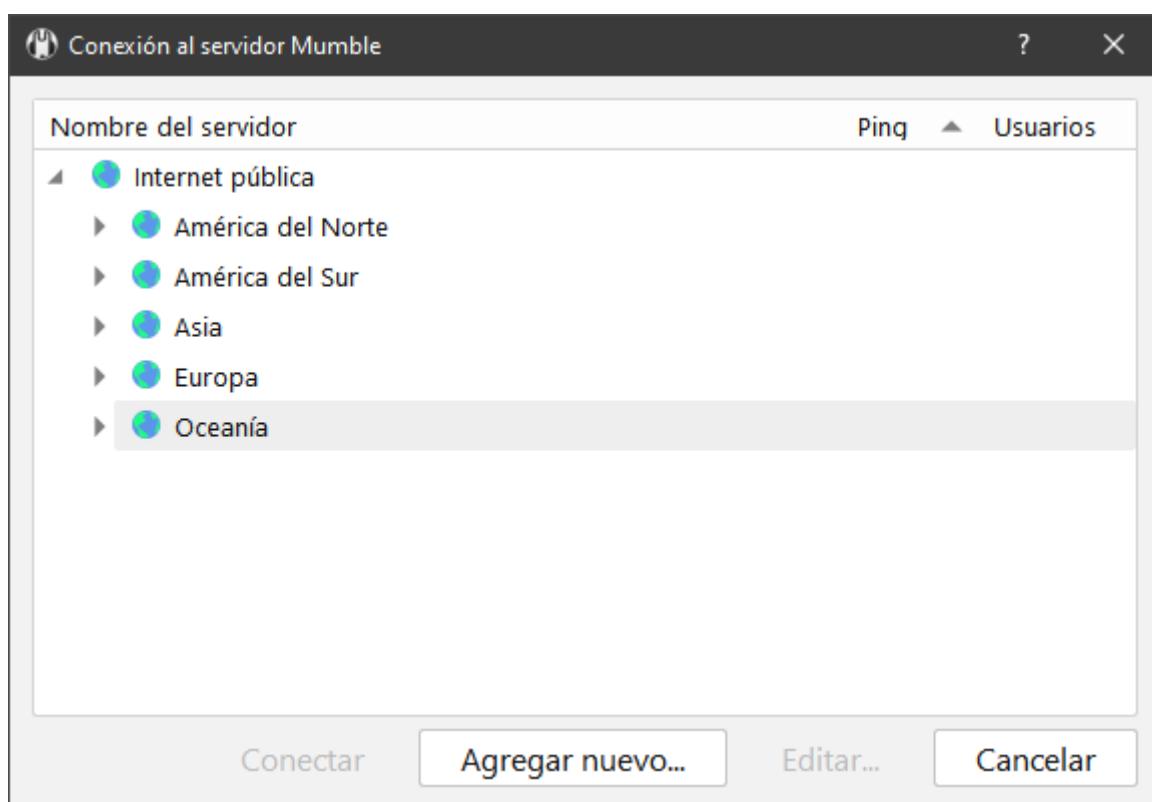
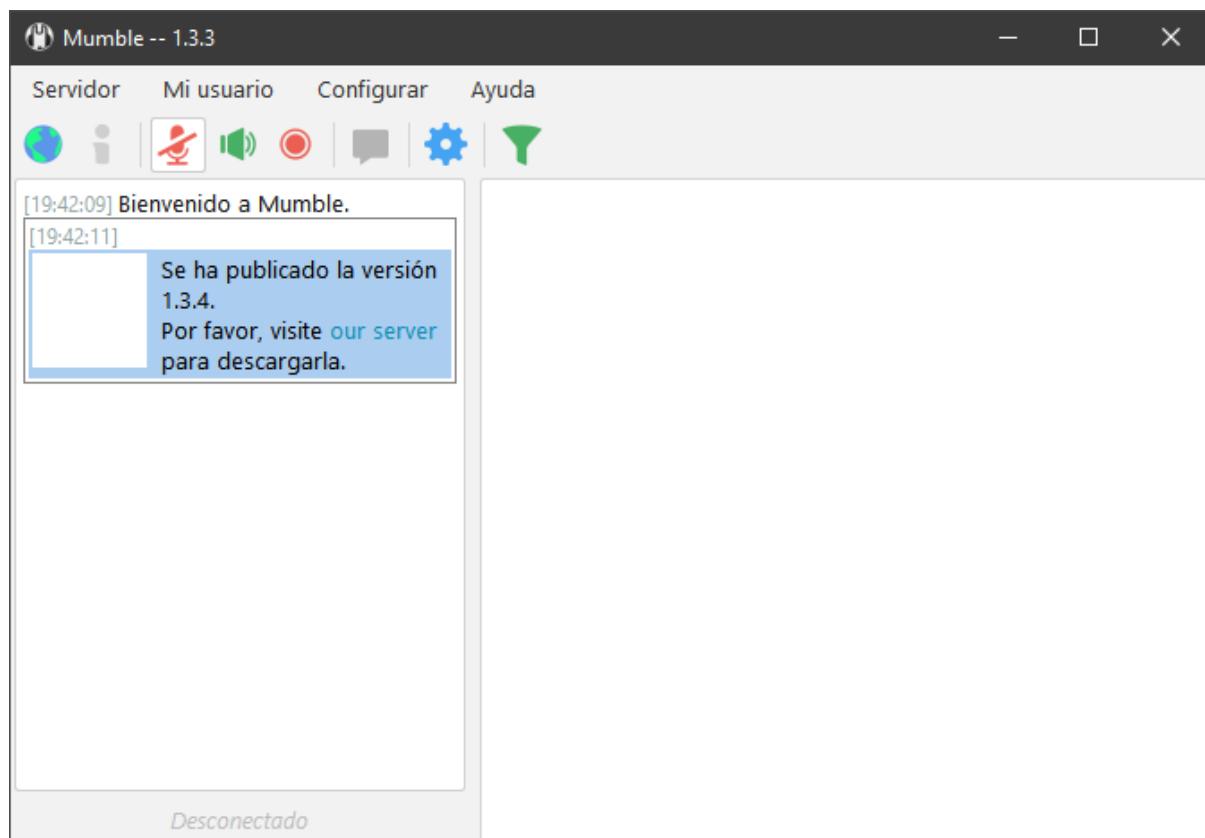
Una vez hayamos descargado Mumble tan solo debemos seguir los pasos de instalación.





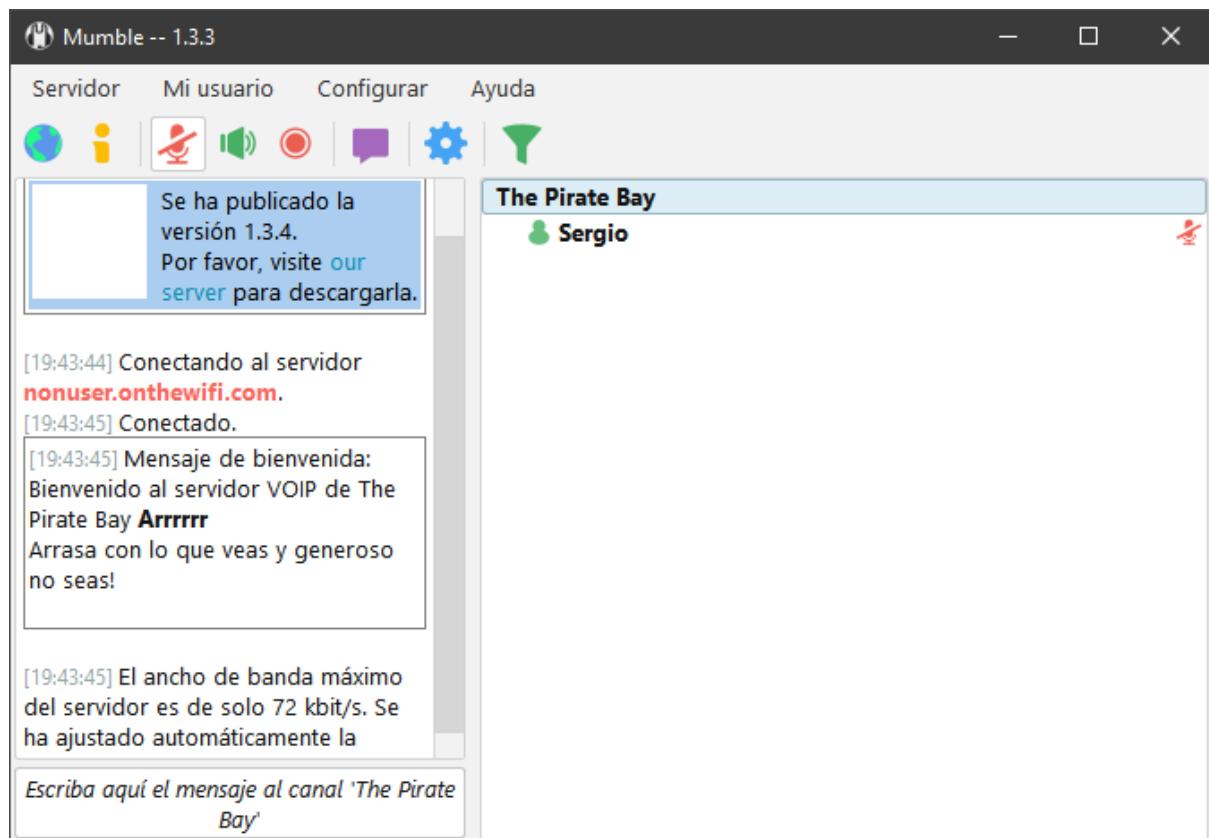
En la imagen de arriba indicaremos en la instalación que solo queremos instalar el cliente.







Una vez instalado debemos agregar el servidor VOIP indicando la dirección, el puerto y un nombre de usuario como se indica en la imagen de arriba.



Por último ya podemos disfrutar de nuestro increíble servicio VOIP para charlar por texto o por voz con nuestros colegas.

Conclusiones

En primer lugar, hemos conseguido una automatización prácticamente completa de la instalación de un servidor Linux que facilita la instalación a un usuario medio o uno con pocos recursos y conocimientos sobre el universo Linux.

Se han gestionado aspectos tan diversos como:

- La configuración base de un servidor Debian.
- La configuración de Quotas para gestionar el espacio límite del que dispone cada usuario.
- La gestión de logs cuando un usuario inicia sesión, de forma que se tengan todos en un mismo fichero de log, todo gestionado por rsyslog.
- La gestión de las copias de seguridad incrementales de forma periódica mediante rsync.
- El de un servidor web con toda la gestión por detrás de usuarios y sesiones, y su propia gestión de errores (la cual no es poca ya que hay que evitar posibles infiltraciones de seguridad).
- Un servidor SSH y SFTP.
- No solo eso sino que también hemos implementado un almacenamiento de webs que permitimos subir por SFTP.
- Un servicio de blogs Wordpress para permitir a los usuarios la creación automática de Blogs .
- Un servidor de correo (con sus 3 servicios SMTP, IMAP y webmail).
- Un servidor VOIP Mumble virtualizado con Docker.
- Un servicio Fail2Ban para evitar ataques masivos que se puedan dar (y los cuales se han dado) y que permite enviar correos cuando se ha baneado a un usuario.
- Un servicio Tripwire, el cual permite comprobar la integridad de los ficheros y controlar la modificación de estos de cara a un posible ataque.
- Servicios de monitorización que hemos creado nosotros, para que nos mandaran periódicamente un informe del estado del servidor.

A lo largo de la práctica hemos aprendido a gestionar los distintos servicios y cómo configurarlos. Ha sido un trabajo bastante bonito ya que nos permitirá en un futuro saber desenvolvernos si nos planteamos hacer un servidor en casa para las necesidades que pudiéramos tener, o en una empresa, de gestionar servicios a gran escala con multitud de clientes.

Referencias

- [▷ Cómo instalar WordPress en Debian 10 Buster ✓ \[2021\] Paso a paso](#)
- [Solved: Request for subsystem 'sftp' failed on channel 0 - Hewlett Packard Enterprise Community](#)
- [How do I change the roundcube logo to my own?](#)
- [Could I edit Roundcube Webmail Title?](#)
- [docker command not found even though installed with apt-get](#)
- [Mumble. El docker del servidor de VoIP de Software Libre.](#)
- [Creating your own Raspberry Pi TeamSpeak Server](#)
- <https://www.cyberciti.biz/tips/howto-write-perl-script-to-monitor-disk-space.html>
- <https://alvinalexander.com/perl/edu/articles/pl010003.shtml>
- <https://gonzalonavarro.es/blog/wordpress-multisite/>
- [Convert HA AWS Wordpress Install to Multisite - WordPress](#)
- [Installing Multiple WordPress Instances](#)
- [How to Install WordPress Multisite](#)
- [Bash Loop Through a List of Strings – Linux Hint](#)
- [linux shell command in Perl program - Unix & Linux Stack Exchange](#)
- [15 thoughts on “Monitor running processes with Perl”](#)
- [Linux + how to give only specific user to read the file](#)
- [How to set limit on directory size in Linux?](#)
- [SQL::Abstract - Generate SQL from Perl data structures - metacpan.org](#)
- [Digital Ocean Tutorial on "How To Reset Your MySQL or MariaDB Root Password"](#)
- [Tutorial CGI de Perl parte 5 - Crear sesiones con CGI::Session \(Demo sistema login\)](#)
- [CGI::Session::Tutorial - extended CGI::Session manual](#)
- [Securing the Linux filesystem with Tripwire](#)
- [Forward Apache Logs to Central Log Server with Rsyslog](#)
- [Managing Linux Logs - The Ultimate Guide To Logging](#)
- [Is it possible to grant users sftp access without shell access? If yes, how is it implemented? - Unix & Linux Stack Exchange](#)

- [Linux - Disk Space Management with Quotas • n0nuser](#)
- [Add Perl module relative to script](#)
- [Why does setuid not work? - Unix & Linux Stack Exchange](#)
- [Linux + how to give only specific user to read the file](#)
- [Authen::PAM::FAQ\(3pm\) — libauthen-pam-perl — Debian testing](#)
- [Authen::Simple::PAM - Simple PAM authentication - metacpan.org](#)
- [Apache Web Server • n0nuser](#)
- [Part 2: Install Dovecot IMAP server on Ubuntu & Enable TLS Encryption](#)
- [Configure SMTP with Gmail Using Postfix](#)
- [Change location of maildir](#)
- [How to Install Postfix, Dovecot, and Roundcube on Ubuntu 20.04](#)
- [Setup Dovecot with PAM authentication and SSL on CentOS - Experiencing Technology](#)
- [How To Install and Configure a Mail Server with Dovecot on Ubuntu 18.04](#)
- [Postfix not receiving mails from outside the localhost](#)
- [Install Let's Encrypt SSL on Ubuntu 20.04 with Apache or Nginx](#)
- [10 tips for Apache Security](#)
- [Secure Apache Web Pages with LDAP Authentication](#)
- [How to set up static IP address on Debian Linux 10/11](#)
- [How to configure Static IP on Debian](#)
- [How to setup a Static IP address on Debian Linux](#)