

TRABAJO DE SERVICIOS DE RED E INTERNET DEL PRIMER TRIMESTRE

ANDRÉS FELIPE RIVERA RIAÑO

2º ASIR

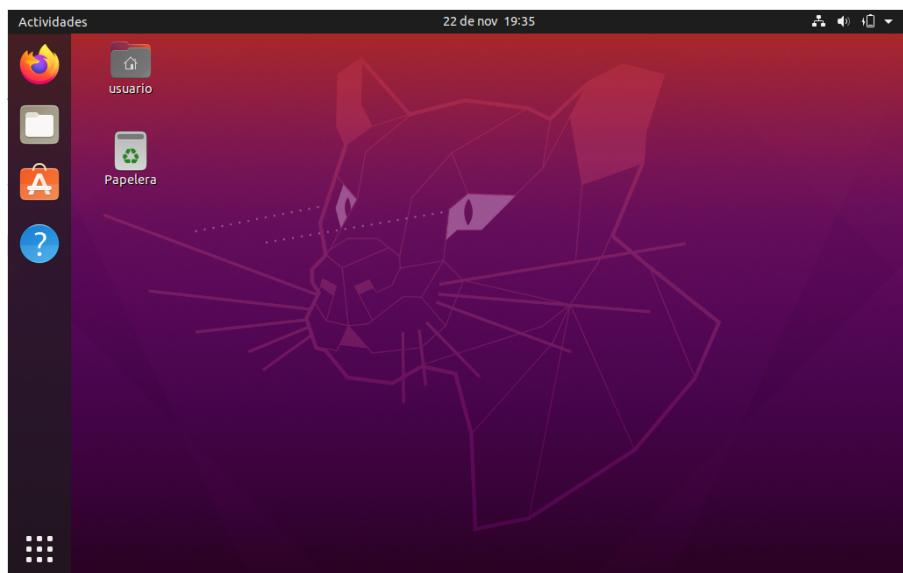
DESCRIPCIÓN DEL TRABAJO

Vamos a instalar un servidor web interno para un instituto. Se Pide:

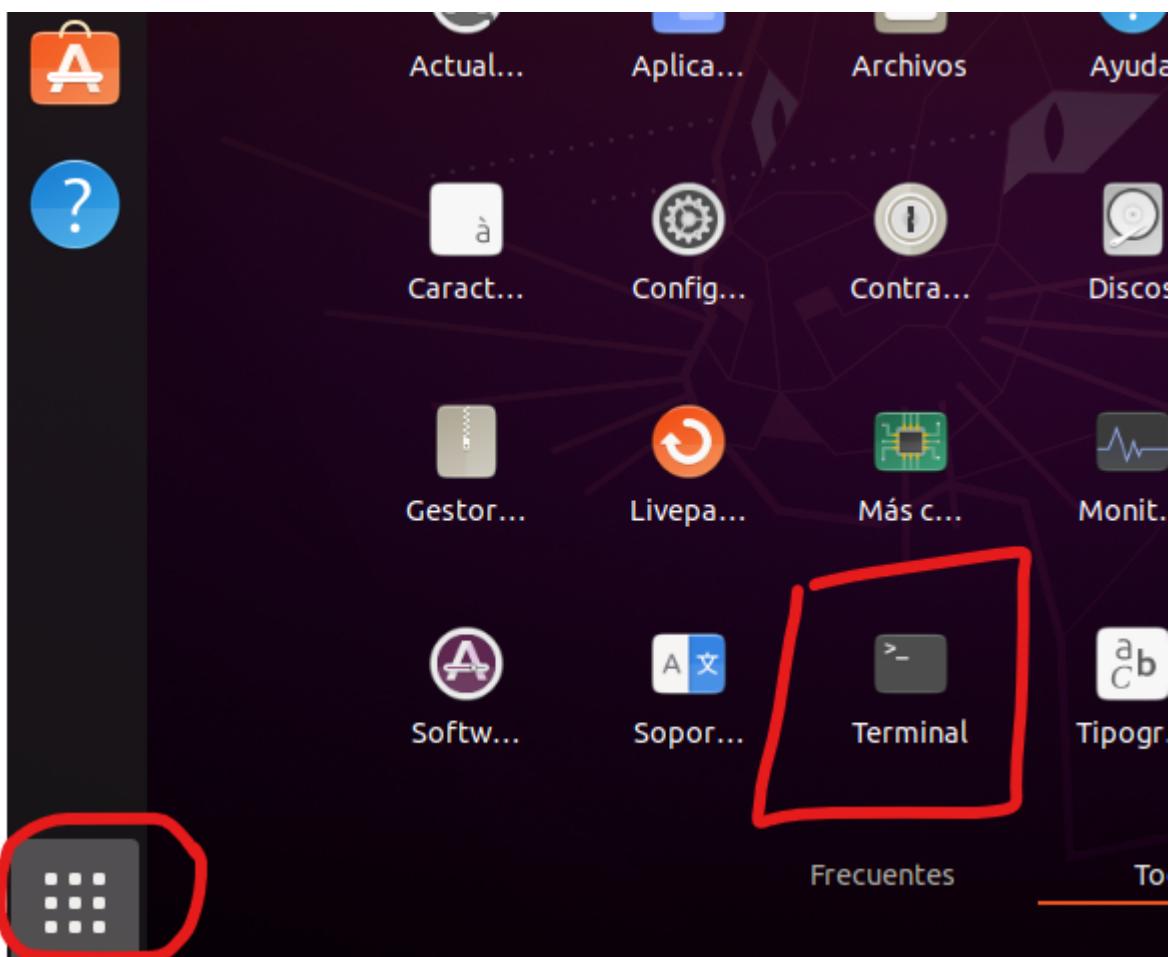
1. Instalación del servidor web Apache. Usaremos dos dominios mediante el archivo hosts: “centro.intranet” y “departamentos.centro.intranet”. El primero servirá el contenido mediante Wordpress y el segundo una aplicación en Python.
2. Activar los módulos necesarios para ejecutar php y acceder a MySQL.
3. Instala y configura Wordpress.
4. Activar el módulo “wsgi” para permitir la ejecución de aplicaciones Python
5. Crea y despliega una pequeña aplicación Python para comprobar que funciona correctamente.
6. Adicionalmente protegeremos el acceso a la aplicación Python mediante autenticación.
7. Instala y configura Awstats.
8. Instala un segundo servidor de tu elección (Nginx, Lighttpd) bajo el dominio “servidor2.centro.intranet”. Debes configurarlo para que sirva en el puerto 8080 y haz los cambios necesarios para ejecutar PHP. Instala PHPMyAdmin.

1. INSTALACIÓN DEL SERVICIO APACHE.

Una vez instalada y configurada mi máquina virtual con Ubuntu 20.04, se verá así:



Lo primero que vamos a configurar es la pila LAMP(Linux, Apache, MySQL y PHP). Para acceder a “Terminal” voy a la parte inferior del Escritorio y busco la aplicación. La ejecuto:



Lo primero que hay que realizar es la actualización de los paquetes de servidor. Se escribe el siguiente comando:

```
sudo apt update
```

Me pedirá la contraseña del usuario root y descargará los paquetes nuevos. Se pedirá confirmación, introduciendo “S” en el teclado e INTRO:

```
o run a command as administrator (user "root"), use "sudo <
ee "man sudo_root" for details.

usuario@usuario-VirtualBox:~$ sudo apt update
[sudo] contraseña para usuario:
[1/1] http://es.archive.ubuntu.com/ubuntu focal InRelease
[2/1] http://security.ubuntu.com/ubuntu focal-security InRelease
[3/1] http://es.archive.ubuntu.com/ubuntu focal-updates InRelease
[4/1] http://es.archive.ubuntu.com/ubuntu focal-backports InRelease
[5/1] http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages
[6/1] http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages
[7/1] http://security.ubuntu.com/ubuntu focal-security/main i386 Packages
[8/1] http://security.ubuntu.com/ubuntu focal-security/main all Packages
[9/1] http://es.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages
[10/1] http://security.ubuntu.com/ubuntu focal-security/main all Packages
[11/1] http://security.ubuntu.com/ubuntu focal-security/main Translation-en
[12/1] http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages
[13/1] http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages
[14/1] http://security.ubuntu.com/ubuntu focal-security/universe i386 Packages
[15/1] http://es.archive.ubuntu.com/ubuntu focal-updates/universe all Packages
[16/1] http://security.ubuntu.com/ubuntu focal-security/universe Translation-en
[17/1] http://security.ubuntu.com/ubuntu focal-security/universe Translation-en
```

Una vez actualizado, se procede a la instalación de Apache con el siguiente comando:

```
sudo apt install apache2
```

```
Se pueden actualizar 123 paquetes. Ejecute "sudo apt upgrade" para
usuario@usuario-VirtualBox:~$ sudo apt install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libcurl4 liblua5.2-0
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-cust
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1-ldap
  libcurl4 liblua5.2-0
0 actualizados, 10 nuevos se instalarán, 0 para eliminar y
Se necesita descargar 2.055 kB de archivos.
Se utilizarán 8.651 kB de espacio de disco adicional después
¿Desea continuar? [S/n]
```

Una vez instalado, toca comprobar el Firewall o Cortafuegos. Se introduce el siguiente comando para ver el listado de aplicaciones:

```
sudo ufw app list
```

```
Procesando dispositivos para libpc-bpf (2.51-ubuntu2.2)
usuario@usuario-VirtualBox:~$ sudo ufw app list
Aplicaciones disponibles:
 Apache
 Apache Full
 Apache Secure
 CUPS
usuario@usuario-VirtualBox:~$
```

Valido a Apache en el Firewall:

```
sudo ufw allow in "Apache"
```

```
CUPS
usuario@usuario-VirtualBox:~$ sudo ufw allow in "Apache"
Reglas actualizadas
Reglas actualizadas (v6)
```

Compruebo que el Firewall está activo:

```
sudo ufw status
```

```
usuario@usuario-VirtualBox:~$ sudo ufw status
Estado: inactivo
usuario@usuario-VirtualBox:~$
```

En mi caso, hay que habilitarlo:

```
sudo ufw enable
```

```
Estado: inactivo
usuario@usuario-VirtualBox:~$ sudo ufw enable
El cortafuegos está activo y habilitado en el arranque del sistema
usuario@usuario-VirtualBox:~$ sudo ufw status
Estado: activo

Hasta          Acción      Desde
-----        -----      -----
Apache          ALLOW      Anywhere
Apache (v6)     ALLOW      Anywhere (v6)
```

A continuación, procedo a la Introducción de dominios en host:

```
sudo nano /etc/hosts
```

Y coloco mis dos dominios:

127.0.0.1	centro.intranet
127.0.0.1	departamentos.centro.intranet

```
GNU nano 4.8                               /etc/hosts
127.0.0.1      localhost
127.0.1.1      usuario-VirtualBox
127.0.0.1      centro.intranet
127.0.0.1      departamentos.centro.intranet

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Pulso CONTROL + O e INTRO para guardar cambios y CONTROL + X para salir de Nano.

Ahora le toca el turno a MySQL, escribiendo el siguiente comando para instalarlo:

```
sudo apt install mysql-server
```

```
usuario@usuario-VirtualBox:~$ sudo apt install mysql-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7
  libhtml-template-perl libmecab2 mecab-ipadic mecab-ipadic-utf8
  mysql-client-core-8.0 mysql-server-8.0 mysql-server-core-8.0
Paquetes sugeridos:
  libipc-sharedcache-perl mailx tinyca
Se instalarán los siguientes paquetes NUEVOS:
  libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7
  libhtml-template-perl libmecab2 mecab-ipadic mecab-ipadic-utf8
  mysql-client-core-8.0 mysql-server mysql-server-8.0 mysql-server-core-8.0
0 actualizados, 16 nuevos se instalarán, 0 para eliminar y 12
Se necesita descargar 31,5 MB de archivos.
Se utilizarán 262 MB de espacio de disco adicional después de la instalación.
¿Desea continuar? [S/n]
```

Una vez instalado, iniciamos:

```
sudo mysql_secure_installation
```

Que permite establecer la contraseña de root para acceder de forma segura a las bases de datos:

```
usuario@usuario-VirtualBox:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y

There are three levels of password validation policy:

LOW      Length >= 8
MEDIUM   Length >= 8, numeric, mixed case, and special characters
STRONG   Length >= 8, numeric, mixed case, special characters and dictionary file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0
Please set the password for root here.

New password:
Re-enter new password:

Estimated strength of the password: 25
Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) : y
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.
```

Se introduce una contraseña segura y en cada uno de los mensajes que nos van saliendo, le introducimos "y" para irlos aceptando.

```
Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
- Dropping test database...
Success.

- Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
usuario@usuario-VirtualBox:~$
```

Una vez completado, se inicia MySQL con:

```
sudo mysql
```

```
All done!
usuario@usuario-VirtualBox:~$ sudo mysql
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.27-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> exit
Bye
usuario@usuario-VirtualBox:~$
```

Y para salir, introducimos:

```
exit
```

Para la instalación de PHP, introduzco el siguiente comando:

```
sudo apt install php libapache2-mod-php php-mysql
```

```
usuario@usuario-VirtualBox:/etc$ sudo apt install php libapache2-mod-php php-mysql
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libapache2-mod-php7.4 php-common php7.4 php7.4-cli php7.4-common php7.4-json php7.4-mysql
  php7.4-opcache php7.4-readline
Paquetes sugeridos:
  php-pear
Se instalarán los siguientes paquetes NUEVOS:
  libapache2-mod-php libapache2-mod-php7.4 php php-common php-mysql php7.4 php7.4-cli php7.4-
  php7.4-json php7.4-mysql php7.4-opcache php7.4-readline
0 actualizados, 12 nuevos se instalarán, 0 para eliminar y 125 no actualizados.
Se necesita descargar 4.144 kB de archivos.
Se utilizarán 18,5 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Una vez instalado, se comprueba la versión con:

```
php -v
```

```
usuario@usuario-VirtualBox:/etc$ php -v
PHP 7.4.3 (cli) (built: Oct 25 2021 18:20:54) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
usuario@usuario-VirtualBox:/etc$ █
```

2. INSTALACIÓN Y CONFIGURACIÓN DE WORDPRESS.

Primero, tenemos que instalar curl, ya que con esta herramienta, podremos descargarlo y configurarlo:

```
sudo apt install curl
```

```
usuario@usuario-VirtualBox:/etc$ sudo apt install curl
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  curl
0 actualizados, 1 nuevos se instalarán, 0 para eliminar
Se necesita descargar 161 kB de archivos.
Se utilizarán 412 kB de espacio de disco adicional desp
Des:1 http://es.archive.ubuntu.com/ubuntu focal-updates
Descargados 161 kB en 1s (133 kB/s)
Seleccionando el paquete curl previamente no seleccionado
(Leyendo la base de datos ... 165500 ficheros o directorios)
Preparando para desempaquetar .../curl_7.68.0-1ubuntu2.7.
Desempaquetando curl (7.68.0-1ubuntu2.7) ...
Configurando curl (7.68.0-1ubuntu2.7) ...
Procesando disparadores para man-db (2.9.1-1) ...
usuario@usuario-VirtualBox:/etc$
```

El primer paso que daremos es preparatorio. WordPress utiliza MySQL para administrar y almacenar el sitio y la información del usuario. Ya instalamos MySQL, pero debemos crear una base de datos y un usuario para que use WordPress.

Para comenzar, inicie sesión en la cuenta root de MySQL (administrativa) emitiendo este comando (tenga en cuenta que este no es el usuario root de su servidor):

```
sudo mysql -u root -p
```

```
usuario@usuario-VirtualBox:/etc$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.27-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> |
```

En la base de datos, puede crear una base de datos exclusiva para que WordPress la controle. Puede ponerle el nombre que quiera, pero usaremos el nombre “centro” en esta guía. Cree la base de datos para WordPress escribiendo lo siguiente:

```
CREATE DATABASE centro DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;
```

```
root@centro-OptiPlex-5070-1: ~# mysql> CREATE DATABASE centro DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;
Query OK, 1 row affected, 2 warnings (0,02 sec)

mysql>
```

A continuación, crearemos una cuenta de usuario separada de MySQL que usaremos exclusivamente para realizar operaciones en nuestra nueva base de datos:

```
CREATE USER 'centrousuario'@'%' IDENTIFIED WITH mysql_native_password BY
'Centrousuario123_';
```

```
mysql> CREATE USER 'centrousuario'@'%' IDENTIFIED WITH mysql_native_password BY 'Centrousuario123_';
Query OK, 0 rows affected (0,02 sec)

mysql>
```

A continuación, deje saber a la base de datos que nuestro “centrousuario” debería tener acceso completo a la base de datos que configuramos:

```
GRANT ALL ON centro.* TO 'centrousuario'@'%';
```

```
mysql> GRANT ALL ON centro.* TO 'centrousuario'@'%';
Query OK, 0 rows affected (0,01 sec)

mysql>
```

Debemos eliminar los privilegios de modo que la instancia actual de MySQL sepa sobre los cambios recientes que hemos realizado:

```
FLUSH PRIVILEGES;
```

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,02 sec)
```

Mostrar aplicaciones

```
mysql> EXIT;
Bye
usuario@usuario-Vi
```

EXIT;

Ahora debemos realizar la instalación de herramientas adicionales de PHP. Primero debemos actualizar la base de datos de repositorio:

```
sudo apt update
```

```
usuario@usuario-VirtualBox:/etc$ sudo apt update
[sudo] contraseña para usuario:
Obj:1 http://es.archive.ubuntu.com/ubuntu focal In
Des:2 http://security.ubuntu.com/ubuntu focal-secu
Des:3 http://es.archive.ubuntu.com/ubuntu focal-up
Des:4 http://es.archive.ubuntu.com/ubuntu focal-ba
Descargados 328 kB en 1s (352 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se pueden actualizar 125 paquetes. Ejecute «apt lis
usuario@usuario-VirtualBox:/etc$ █
```

Podemos descargar e instalar algunas de las extensiones de PHP más populares para usarlas con WordPress escribiendo lo siguiente:

```
sudo apt install php-curl php-gd php-mbstring php-xml php-xmlrpc php-soap php-intl php-zip
```

```
0 actualizados, 125 nuevos se instalarán. Ejecute «apt list --upgradable» para verlos.
usuario@usuario-VirtualBox:/etc$ sudo apt install php-curl php-gd php-mbstring php-xml php-xmlrpc php-so
ap php-intl php-zip
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
 libonig5 libxmlrpc-epi0 libzip5 php7.4-curl php7.4-gd php7.4-intl php7.4-mbstring php7.4-soap
 php7.4-xml php7.4-xmlrpc php7.4-zip
Se instalarán los siguientes paquetes NUEVOS:
 libonig5 libxmlrpc-epi0 libzip5 php-curl php-gd php-intl php-mbstring php-soap php-xml php-xmlrpc
 php-zip php7.4-curl php7.4-gd php7.4-intl php7.4-mbstring php7.4-soap php7.4-xml php7.4-xmlrpc
 php7.4-zip
0 actualizados, 19 nuevos se instalarán, 0 para eliminar y 125 no actualizados.
Se necesita descargar 1.063 kB de archivos.
Se utilizarán 3.814 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] █
```

Necesitaremos reiniciar Apache para cargar estas nuevas extensiones:

```
sudo systemctl restart apache2
```

Ahora le toca el paso al ajuste de la configuración de Apache para permitir reemplazos y reescrituras .htaccess:

Actualmente, el uso de archivos .htaccess está desactivado. WordPress y muchos de sus complementos utilizan estos archivos de forma amplia para realizar ajustes de comportamiento del servidor web dentro del directorio.

Abra el archivo de configuración de Apache para centro.intranet:

```
cd /etc/apache2/sites-available
```

y

```
sudo nano centro.intranet.conf
```

Para permitir archivos .htaccess, debemos configurar la directiva AllowOverride dentro de un bloque Directory orientado a nuestro root de documentos. Agregue el siguiente bloque de texto dentro del bloque VirtualHost:

```
<VirtualHost *:80>
    ServerName centro.intranet
    ServerAlias www.centro.intranet
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/centro.intranet
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

```
GNU nano 4.8
<VirtualHost *:80>
    ServerName centro.intranet
    ServerAlias www.centro.intranet
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/centro.intranet
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Ahora, puede usar a2ensite para habilitar el nuevo host virtual:

```
sudo a2ensite centro.intranet
```

```
sudo systemctl reload apache2
```

```
usuario@usuario-VirtualBox:/etc/apache2/sites-available$ sudo nano centro.intranet.com
usuario@usuario-VirtualBox:/etc/apache2/sites-available$ sudo a2ensite centro.intranet
Enabling site centro.intranet.
To activate the new configuration, you need to run:
    systemctl reload apache2
usuario@usuario-VirtualBox:/etc/apache2/sites-available$ sudo systemctl reload apache2
usuario@usuario-VirtualBox:/etc/apache2/sites-available$ █
```

Puede ser conveniente deshabilitar el sitio web predeterminado que viene instalado con Apache. Es necesario hacerlo si no se utiliza un nombre de dominio personalizado, dado que, en este caso, la configuración predeterminada de Apache sobrescribirá su host virtual. Para deshabilitar el sitio web predeterminado de Apache, escriba lo siguiente:

```
sudo a2dissite 000-default  
systemctl reload apache2
```

```
usuario@usuario-VirtualBox:/etc/apache2/sites-available$ sudo a2dissite 000-default  
Site 000-default disabled.  
To activate the new configuration, you need to run:  
    systemctl reload apache2  
usuario@VirtualBox:/etc/apache2/sites-available$ systemctl reload apache2  
usuario@VirtualBox:/etc/apache2/sites-available$ █
```

Ahora toca habilitar reemplazos .htaccess, ya que actualmente, el uso de archivos .htaccess está desactivado. WordPress y muchos de sus complementos utilizan estos archivos de forma amplia para realizar ajustes de comportamiento del servidor web dentro del directorio.

Abra el archivo de configuración de Apache para “centro.intranet” con un editor de texto como nano.

```
sudo nano /etc/apache2/sites-available/centro.intranet.conf
```

Y añadimos:

```
<Directory /var/www/centro.intranet/>  
    AllowOverride All  
</Directory>
```

```
usuario@usuario-VirtualBox:/etc/apache2/sites-available$  
<VirtualHost *:80>  
    ServerName centro.intranet  
    ServerAlias www.centro.intranet  
    ServerAdmin webmaster@localhost  
    DocumentRoot /var/www/centro.intranet  
    ErrorLog ${APACHE_LOG_DIR}/error.log  
    CustomLog ${APACHE_LOG_DIR}/access.log combined  
  
    <Directory /var/www/centro.intranet/>  
        AllowOverride All  
    </Directory>
```

Cuando termine, guarde y cierre el archivo. En nano, puede hacer esto pulsando CTRL y X juntos, luego Y, y luego ENTER.

A continuación le toca el turno al **módulo de reescritura**. Podemos habilitar mod_rewrite para usar la característica de permalink de WordPress:

```
sudo a2enmod rewrite
```

```
usuario@usuario-VirtualBox:/etc/apache2/sites-available$ sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  systemctl restart apache2
usuario@usuario-VirtualBox:/etc/apache2/sites-available$ systemctl restart apache2
usuario@usuario-VirtualBox:/etc/apache2/sites-available$ █
```

Probamos la sintaxis de Apache:

```
sudo apache2ctl configtest
```

En tanto el resultado contenga Sintaxis OK, podremos continuar. Por último, reinicia Apache para implementar los cambios.

```
sudo systemctl restart apache2
```

Ahora que el software de nuestro servidor está configurado, podemos descargar y configurar WordPress.

Cambie a un directorio que permita la escritura (recomendamos uno temporal como /tmp) y descargue la versión comprimida.

```
cd /tmp
curl -O https://wordpress.org/latest.tar.gz
```

```
usuario@usuario-VirtualBox:/tmp$ curl -O https://wordpress.org/latest.tar.gz
% Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload Total   Spent    Left Speed
 60 14.3M    60 8960k     0      0  1188k      0  0:00:12  0:00:07  0:00:05 1242k
```

Extraiga el archivo comprimido para crear la estructura de directorios de WordPress:

```
tar xzvf latest.tar.gz
```

Moveremos estos archivos a nuestro root de documentos por ahora. Antes de hacerlo, podemos añadir un archivo ficticio .htaccess de modo que esté disponible para que WordPress lo use más adelante.

Cree el archivo escribiendo lo siguiente:

```
touch /tmp/wordpress/.htaccess
```

```
usuario@usuario-VirtualBox:/tmp/wordpress$ touch /tmp/wordpress/.htaccess
usuario@usuario-VirtualBox:/tmp/wordpress$ ll
total 220
drwxr-xr-x  5 usuario usuario  4096 nov 22 21:30 .
drwxrwxrwt 20 root   root    4096 nov 22 21:28 ../
-rw-rw-r--  1 usuario usuario   0 nov 22 21:30 .htaccess
-rw-r--r--  1 usuario usuario  405 feb  6 2020 index.php
-rw-r--r--  1 usuario usuario 19915 ene  1 2021 license.txt
-rw-r--r--  1 usuario usuario 7346 jul  6 14:23 readme.html
-rw-r--r--  1 usuario usuario 7165 ene 21 2021 wp-activate.php
drwxr-xr-x  9 usuario usuario  4096 nov 10 18:24 wp-admin/
-rw-r--r--  1 usuario usuario  351 feb  6 2020 wp-blog-header.php
-rw-r--r--  1 usuario usuario 2328 feb 17 2021 wp-comments-post.php
-rw-r--r--  1 usuario usuario 3004 may 21 2021 wp-config-sample.php
drwxr-xr-x  4 usuario usuario  4096 nov 10 18:24 wp-content/
-rw-r--r--  1 usuario usuario 3939 jul 30 2020 wp-cron.php
drwxr-xr-x 25 usuario usuario 12288 nov 10 18:24 wp-includes/
-rw-r--r--  1 usuario usuario 2496 feb  6 2020 wp-links-opml.php
-rw-r--r--  1 usuario usuario 3900 may 15 2021 wp-load.php
-rw-r--r--  1 usuario usuario 45463 abr  6 2021 wp-login.php
-rw-r--r--  1 usuario usuario 8509 abr 14 2020 wp-mail.php
-rw-r--r--  1 usuario usuario 22297 jun  2 01:09 wp-settings.php
-rw-r--r--  1 usuario usuario 31693 may  7 2021 wp-signup.php
-rw-r--r--  1 usuario usuario 4747 oct  8 2020 wp-trackback.php
-rw-r--r--  1 usuario usuario 3236 jun  8 2020 xmlrpc.php
usuario@usuario-VirtualBox:/tmp/wordpress$ S
```

También copiaremos sobre el archivo de configuración de muestra al nombre de archivo que lee WordPress:

```
cp /tmp/wordpress/wp-config-sample.php /tmp/wordpress/wp-config.php
```

También podemos crear el directorio de actualización, de modo que WordPress no tenga problemas de permisos al intentar hacerlo por su cuenta siguiendo una actualización a su software:

```
mkdir /tmp/wordpress/wp-content/upgrade
```

Ahora podemos **copiar todo el contenido del directorio en nuestro root de documentos**. Usaremos un punto al final de nuestro directorio de origen para indicar que todo lo que está dentro del directorio debe copiarse, incluyendo archivos ocultos (como el archivo .htaccess que hemos creado):

```
sudo cp -a /tmp/wordpress/. /var/www/centro.intranet
```

(nota: renombré la carpeta luego después de efectuar la operación a centro.intranet según conveniencia)

```
usuario@usuario-VirtualBox:/tmp/wordpress$ cp /tmp/wordpress/wp-config-sample.php /tmp/wordpress/wp-config.php
usuario@usuario-VirtualBox:/tmp/wordpress$ mkdir /tmp/wordpress/wp-content/upgrade
usuario@usuario-VirtualBox:/tmp/wordpress$ sudo cp -a /tmp/wordpress/. /var/www/centro
[sudo] contraseña para usuario:
usuario@usuario-VirtualBox:/tmp/wordpress$
```

```
usuario@usuario-VirtualBox:/var/www/centro.intranet$ cd ..
usuario@usuario-VirtualBox:/var/www$ ll
total 16
drwxr-xr-x 4 root      root      4096 nov 23 09:50 .
drwxr-xr-x 15 root      root      4096 nov 22 19:38 ../
drwxr-xr-x  5 usuario   usuario   4096 nov 23 09:26 centro.intranet/
drwxr-xr-x  2 root      root      4096 nov 22 19:38 html/
usuario@usuario-VirtualBox:/var/www$ cd centro.intranet
usuario@usuario-VirtualBox:/var/www/centro.intranet$ ll
total 224
drwxr-xr-x  5 usuario   usuario   4096 nov 23 09:26 .
drwxr-xr-x  4 root      root      4096 nov 23 09:50 ../
-rw-rw-r--  1 usuario   usuario    0 nov 22 21:30 .htaccess
-rw-r--r--  1 usuario   usuario   405 feb  6 2020 index.php
-rw-r--r--  1 usuario   usuario 19915 ene  1 2021 license.txt
-rw-r--r--  1 usuario   usuario  7346 jul  6 14:23 readme.html
-rw-r--r--  1 usuario   usuario  7165 ene 21 2021 wp-activate.php
drwxr-xr-x  9 usuario   usuario  4096 nov 16 18:24 wp-admin/
-rw-r--r--  1 usuario   usuario  351 feb  6 2020 wp-blog-header.php
-rw-r--r--  1 usuario   usuario 2328 feb  1 2021 wp-comments-post.php
-rw-r--r--  1 usuario   usuario 3004 nov 22 21:33 wp-config.php
-rw-r--r--  1 usuario   usuario 3004 may 21 2021 wp-config-sample.php
drwxr-xr-x  5 usuario   usuario  4096 nov 22 21:33 wp-content/
-rw-r--r--  1 usuario   usuario 3939 jul 30 2020 wp-cron.php
drwxr-xr-x 25 usuario   usuario 12288 nov 10 18:24 wp-includes/
-rw-r--r--  1 usuario   usuario 2496 feb  6 2020 wp-links-opml.php
-rw-r--r--  1 usuario   usuario 3900 may 15 2021 wp-load.php
-rw-r--r--  1 usuario   usuario 45463 abr  6 2021 wp-login.php
-rw-r--r--  1 usuario   usuario 8509 abr 14 2020 wp-mail.php
-rw-r--r--  1 usuario   usuario 22297 jun  2 01:09 wp-settings.php
-rw-r--r--  1 usuario   usuario 31693 may  7 2021 wp-signup.php
-rw-r--r--  1 usuario   usuario 4747 oct  8 2020 wp-trackback.php
-rw-r--r--  1 usuario   usuario 3236 jun  8 2020 xmlrpc.php
usuario@usuario-VirtualBox:/var/www/centro.intranet$
```

Ahora podremos **configurar el directorio de WordPress**, aunque antes de realizar la configuración, debemos ajustar algunos elementos en nuestro directorio de WordPress.

Empecemos realizando ajustes de propiedad y permisos, primero a todos los archivos, al usuario y al grupo www-data. Este es el usuario como el que ejecuta el servidor web Apache, y este último deberá poder leer y escribir archivos de WordPress para presentar el sitio web y realizar actualizaciones automáticas.

Actualice la propiedad con el comando chown que le permite modificar la propiedad del archivo.

```
sudo chown -R www-data:www-data /var/www/centro.intranet
```

```
usuario@usuario-VirtualBox:/var/www$ sudo chown -R www-data:www-data /var/www/centro.intranet
```

A continuación, ejecutaremos dos comandos find para establecer los permisos correctos de los directorios y archivos de WordPress:

```
sudo find /var/www/centro.intranet/ -type d -exec chmod 750 {} \;
sudo find /var/www/centro.intranet/ -type f -exec chmod 640 {} \;
```

```
usuario@usuario-VirtualBox:/var/www$ sudo find /var/www/centro.intranet/ -type d -exec chmod 750 {} \;
usuario@usuario-VirtualBox:/var/www$ sudo find /var/www/centro.intranet/ -type f -exec chmod 640 {} \;
```

Estos permisos deberían hacer que pueda trabajar de forma efectiva con WordPress.

El siguiente paso será **configurar el archivo de configuración de WordPress**, debiendo realizar algunos cambios en el archivo de configuración principal de WordPress.

Cuando abramos el archivo, nuestra primera tarea será ajustar algunas claves secretas para proporcionar un nivel de seguridad a nuestra instalación. WordPress proporciona un generador seguro para estos valores, para que no tenga que crear valores correctos por su cuenta.

Para obtener valores seguros del generador de claves secretas de WordPress, escriba lo siguiente:

```
curl -s https://api.wordpress.org/secret-key/1.1/salt/
```

```
usuario@usuario-VirtualBox:/var/www$ curl -s https://api.wordpress.org/secret-key/1.1/salt/
define('AUTH_KEY', 'CQpcxEiVOmfLSKlAhY~eqW>?-h:tw_r`q;6+D|V)VS;iVO`_Lbe1+$0HiHk5k-so');
define('SECURE_AUTH_KEY', 'V[~BzX){Y8aZKQ%6B-1r=f;#^FlWnrW5k*&gC@BB,Q?go,Wy4SyyjF<)@+G*Tk,C');
define('LOGGED_IN_KEY', '3`C@v-+)CAFGi#Oa/<c{?S]ay/ug.%>3[>Ha2#:|&+f$1p)d7k|N2[KPlBr>z!');
define('NONCE_KEY', 'P,Bxs-,7y_`^G*+]2z$pVrrB0q>~{}2%JDNEfwjF/+&`{bo$!0,-I;#)HL2D2`Pk');
define('AUTH_SALT', 'kZx1Z= (+l1Ked#q-k uA*$J:K++-&%W3^.M&&GsVFcgPAbV&;j>8~z)u0+<a:E!');
define('SECURE_AUTH_SALT', '_3a%7bn:o2af5V|:+91V<%p2ov6yqd)EE/$Wr+UD{L<xD|3GBlrbs${&|,$K6@?');
define('LOGGED_IN_SALT', 'Tl|j0r)pzjG,~E1!LLaqdQifZJ-3Q=${+W9`M~lJyf!+KZTd Ai-}i#WtR1KhF?');
define('NONCE_SALT', 'w5EdtJ*?+^e-mFy>s=yY>exSld845~tr+<8*D-.45|g[7a_Ho<I^i3/c?fu$0}.');
```

Son líneas de configuración que podemos pegar directamente en nuestro archivo de configuración para establecer claves seguras. Copie el resultado que obtuvo ahora.

```
define('AUTH_KEY', 'CQpcxEiVOmfLSKlAhY~eqW>?-h:tw_r`q;6+D|V)VS;iVO`_Lbe1+$0HiHk5k-so');
define('SECURE_AUTH_KEY', 'V[~BzX){Y8aZKQ%6B-1r=f;#^FlWnrW5k*&gC@BB,Q?go,Wy4SyyjF<)@+G*Tk,C');
define('LOGGED_IN_KEY', '3`C@v-+)CAFGi#Oa/<c{?S]ay/ug.%>3[>Ha2#:|&+f$1p)d7k|N2[KPlBr>z!');
define('NONCE_KEY', 'P,Bxs-,7y_`^G*+]2z$pVrrB0q>~{}2%JDNEfwjF/+&`{bo$!0,-I;#)HL2D2`Pk');
define('AUTH_SALT', 'kZx1Z= (+l1Ked#q-k uA*$J:K++-&%W3^.M&&GsVFcgPAbV&;j>8~z)u0+<a:E!');
define('SECURE_AUTH_SALT', '_3a%7bn:o2af5V|:+91V<%p2ov6yqd)EE/$Wr+UD{L<xD|3GBlrbs${&|,$K6@?');
define('LOGGED_IN_SALT', 'Tl|j0r)pzjG,~E1!LLaqdQifZJ-3Q=${+W9`M~lJyf!+KZTd Ai-}i#WtR1KhF?');
define('NONCE_SALT', 'w5EdtJ*?+^e-mFy>s=yY>exSld845~tr+<8*D-.45|g[7a_Ho<I^i3/c?fu$0}.');
```

```

define('LOGGED_IN_SALT', 'Tl|jOr)pz?jG,~E1!LLaqdQifZJ-3Q=${+W9`M~lJyf!+KZTd
Ai~]i#WtR1KhF?');
define('NONCE_SALT',
'w5EdtJ*?+^e-mFy>s=yY>exSID845~tr+<8^D-.45|g[7a_Ho<l^i3/c?fu$O}. ');

```

A continuación, abra el archivo de configuración de WordPress:

```
sudo nano /var/www/wordpress/wp-config.php
```

Busque la sección que contiene los valores de ejemplo para esos ajustes:

```

GNU nano 4.8                               /var/www/centro.intranet/wp-config.php
* You can change these at any point in time to invalidate all existing c
* This will force all users to have to log in again.
*
* @since 2.6.0
*/
efine( 'AUTH_KEY',      'put your unique phrase here' );
efine( 'SECURE_AUTH_KEY', 'put your unique phrase here' );
efine( 'LOGGED_IN_KEY',   'put your unique phrase here' );
efine( 'NONCE_KEY',      'put your unique phrase here' );
efine( 'AUTH_SALT',       'put your unique phrase here' );
efine( 'SECURE_AUTH_SALT', 'put your unique phrase here' );
efine( 'LOGGED_IN_SALT',   'put your unique phrase here' );
efine( 'NONCE_SALT',      'put your unique phrase here' );

***#@-*/
**
* WordPress database table prefix.
```

Elimine esas líneas y pegue los valores que copió de la línea de comandos:

```

*/ 
define('AUTH_KEY',      'CQpcxEiV0mfLSKLAhY~eqW>?-h:tw_r`q;6+D|V)VS;iV0`_Lbe1+$0HiHk5k-so');
define('SECURE_AUTH_KEY', 'V[~BzX){`8aZKQ%6B-1r=f;#^FLWnrW5k&*gC@BB,Q?go,Wy4SyyjF<@+G*Tk,C');
define('LOGGED_IN_KEY',   '3`C@v-+)CAFGiou#0a/<c{?S]ay/ug.%>3[>Ha2#:|&+$1p)d7k|N2[KPlBr>z!');
define('NONCE_KEY',      'P,Bxs-,7y_~G*+>]2z$pVrrB0q->[]2%JDNEfwjF/+&`{bo$!0,-I:#)HL2D2`Pk');
define('AUTH_SALT',       'kZx1Z= (+l1Ked#q-k uA*$J:K+-&%W3^.M&&GsVFcgPAbV&;j>8~z)u0+<a:E!');
define('SECURE_AUTH_SALT', '_3a%7bn:o2af5V|:~+91V<%p2ov6yqD)EE/$Wr+UD{L<xD|3GBlrbs${&|,$K6@?');
define('LOGGED_IN_SALT',   'Tl|jOr)pz?jG,~E1!LLaqdQifZJ-3Q=${+W9`M~lJyf!+KZTd Ai~]i#WtR1KhF?');
define('NONCE_SALT',      'w5EdtJ*?+^e-mFy>s=yY>exSID845~tr+<8^D-.45|g[7a_Ho<l^i3/c?fu$O}. ');

***#@_*
```

A continuación, vamos a **modificar algunos de los ajustes de conexión de la base de datos** al principio del archivo. Debe ajustar el nombre de la base de datos, su usuario y la contraseña asociada que configuramos dentro de MySQL.

El otro cambio que debemos realizar es configurar el método que debe emplear WordPress para escribir el sistema de archivos. Debido a que hemos dado permiso al servidor web para escribir donde debe hacerlo, podemos fijar de forma explícita el método del sistema de

archivos a “direct”. Si no lo configuramos con nuestros ajustes actuales, WordPress solicitará las credenciales de FTP cuando realicemos algunas acciones.

Este ajuste se puede agregar debajo de los ajustes de conexión de la base de datos o en cualquier otra parte del archivo:

```
sudo nano /var/www/centro.intranet/wp-config.php
```

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'centro' );
```

```
/** MySQL database username */
define( 'DB_USER', 'centrousuario' );
```

```
/** MySQL database password */
define( 'DB_PASSWORD', 'Centrousuario123_' );
```

```
/** MySQL hostname */
define( 'DB_HOST', 'localhost' );
```

```
/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );
```

```
/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

```
...
```

```
define('FS_METHOD', 'direct');
```

```
GNU nano 4.8                               /var/www/centro.intranet/wp-config.php
*/
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'centro' );

/** MySQL database username */
define( 'DB_USER', 'centrousuario' );

/** MySQL database password */
define( 'DB_PASSWORD', 'Centrousuario123_' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

```
/* Add any custom values between this line and the one below */

define( 'FS_METHOD', 'direct' );
```

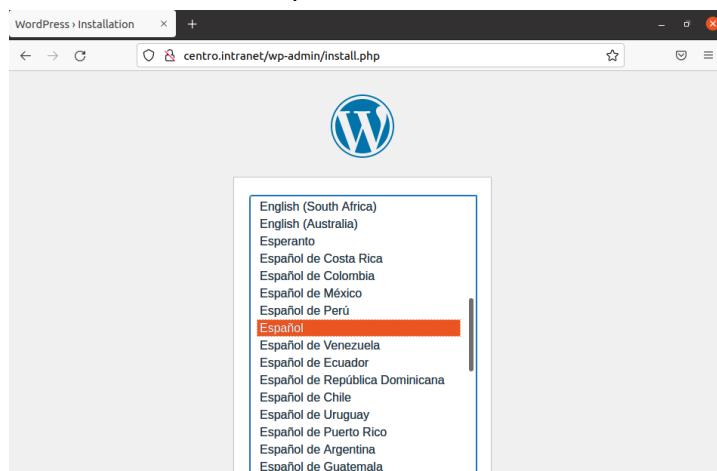
Guarda y cierra el archivo cuando termines.

Una vez completado, seguiremos con la instalación a través de la interfaz web

En el navegador web, introducimos:

<http://127.0.0.1> Ó <http://centro.intranet>

Seleccione el idioma que desee usar:



A continuación, accederá a la **página principal de configuración**.

Selecciona un nombre para su sitio WordPress y selecciona un nombre de usuario. De forma automática, se generará una contraseña segura o escribir una contraseña alternativa.

Introduce tu dirección de correo electrónico y define si quieras que los motores de búsqueda no indexen su sitio. Por último, damos click a Instalar Wordpress en la parte inferior:

Título del sitio	centro.intranet
Nombre de usuario	centrousuario
	Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.
Contraseña	<input type="password" value="Centrousuario123_"/>  Ocultar Débil
	Importante: Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.
Confirma la contraseña	<input checked="" type="checkbox"/> Confirma el uso de una contraseña débil.
Tu correo electrónico	<input type="text" value="aafrivera@ieslamarisma.net"/>
	Comprueba bien tu dirección de correo electrónico antes de continuar.
Visibilidad en los motores de búsqueda	<input checked="" type="checkbox"/> Disuadir a los motores de búsqueda de indexar este sitio Depende de los motores de búsqueda atender esta petición o no.
Mostrar aplicaciones	Instalar WordPress

Cuando haga clic para seguir, irá a una página que pide que inicie sesión:

¡Lo lograste!

WordPress ya está instalado. ¡Gracias, y que lo disfrutes!

Nombre de usuario centrousuario

Contraseña *La contraseña que has elegido.*

[Acceder](#)

Tras iniciar sesión, accede al panel de administración de WordPress:

Tras iniciar sesión, accede al panel de administración de WordPress:

The screenshot shows the WordPress dashboard with the following content:

- Primeros pasos:**
 - Personaliza tu sitio
 - O [cambia tu tema por completo](#)
- Siguientes pasos:**
 - Escribe tu primera entrada en el blog
 - Añade una página «Acerca de»
 - Establece tu página de inicio
 - Ver tu sitio
- Más acciones:**
 - Gestionar widgets
 - Gestionar menús
 - Activa o desactiva los comentarios
 - Aprende más sobre cómo empezar
- Estado de salud del sitio:** Aún no hay pruebas de salud.
- Borrador rápido:** Título (empty input field).

En este momento, puedes comenzar a diseñar tu sitio web WordPress:

The screenshot shows the published WordPress site with the following content:

CENTRO.INTRANET

Otro sitio realizado con WordPress

¡Hola, mundo!

Bienvenido a WordPress. Esta es tu primera entrada. Edítala o bórrala, ¡luego empieza a escribir!

Publicada el 23 de noviembre de 2021 [Editar](#)

WordPress está ahora instalado y listo para usarse.

3. ACTIVAR EL MÓDULO “wsgi” PARA PERMITIR LA EJECUCIÓN DE APLICACIONES PYTHON.

En principio, necesitamos hacer que Apache pueda servir archivos Python. Para ello, necesitaremos habilitar un módulo que brinde este soporte.

Existen varios módulos de Apache que brindan soporte para correr archivos Python. Usaremos el módulo mod_wsgi.

Para habilitar mod_wsgi en Apache, basta con **instalar el paquete libapache2-mod-wsgi**:

```
sudo apt-get install libapache2-mod-wsgi
```

```
usuario@usuario-VirtualBox:/var/www$ sudo apt-get install libapache2-mod-wsgi
[sudo] contraseña para usuario:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libpython2-stdlib libpython2.7 libpython2.7-minimal libpython2.7-stdlib python2 python2-mi
  python2.7 python2.7-minimal
Paquetes sugeridos:
  python2-doc python-tk python2.7-doc binutils binfmt-support
Se instalarán los siguientes paquetes NUEVOS:
  libapache2-mod-wsgi libpython2-stdlib libpython2.7 libpython2.7-minimal libpython2.7-stdlib
  python2-minimal python2.7 python2.7-minimal
0 actualizados, 9 nuevos se instalarán, 0 para eliminar y 125 no actualizados.
```

Ahora toca **crear la estructura de directorios para nuestra aplicación**. Primero, es importante saber cómo va a funcionar nuestra aplicación y cómo va a interactuar vía Web por lo que debemos tener un directorio destinado a montar toda la aplicación:

```
mkdir /home/usuario/departamentos_py/trunk/departamentos-web
```

Dentro de este directorio, vamos a dividir su arquitectura en dos partes:

- Destinada al almacenaje de nuestra aplicación Python pura (será un directorio privado, no servido).
- Destinada a servir la aplicación (directorio público servido) en el cuál solo almacenaremos archivos estáticos.

Estas serán las rutas temporales para nuestra aplicación Python:

```
mkdir /home/usuario/departamentos_py/trunk/departamentos-web/departamentosapp
mkdir /home/usuario/departamentos_py/trunk/departamentos-web/public_html
```

```
usuario@usuario-VirtualBox:~$ mkdir departamentos_py
usuario@usuario-VirtualBox:~$ cd departamentos_py
usuario@usuario-VirtualBox:~/departamentos_py$ mkdir trunk
usuario@usuario-VirtualBox:~/departamentos_py$ cd trunk
usuario@usuario-VirtualBox:~/departamentos_py/trunk$ mkdir departamentos-web
usuario@usuario-VirtualBox:~/departamentos_py/trunk$ ll
total 12
drwxrwxr-x 3 usuario usuario 4096 nov 23 20:56 .
drwxrwxr-x 3 usuario usuario 4096 nov 23 20:56 ..
drwxrwxr-x 2 usuario usuario 4096 nov 23 20:56 departamentos-web/
usuario@usuario-VirtualBox:~/departamentos_py/trunk$ cd departamentos-web
usuario@usuario-VirtualBox:~/departamentos_py/trunk/departamentos-web$ mkdir /home/usuario/departamentos
_py/trunk/departamentos-web/departamentosapp
usuario@usuario-VirtualBox:~/departamentos_py/trunk/departamentos-web$ mkdir /home/usuario/departamentos
_py/trunk/departamentos-web/public_html
usuario@usuario-VirtualBox:~/departamentos_py/trunk/departamentos-web$
```

Dentro de nuestro directorio “departamentosapp”, almacenaremos entonces, todos los módulos y paquetes de nuestra aplicación Python, mientras que en public_html, estarán todos los archivos estáticos y será el único directorio al que se pueda acceder mediante el navegador Web.

Aprovecharemos este paso, para crear una carpeta, destinada a almacenar los logs de errores y accesos a nuestra Web App:

```
mkdir /home/usuario/departamentos_py/trunk/departamentos-web/logs
```

```
usuario@usuario-VirtualBox:~/departamentos_py/trunk/departamentos-web$ mkdir /home/usuario/departamentos
_py/trunk/departamentos-web/logs
```

El siguiente paso es **crear un controlador para la aplicación**.

Todas las peticiones realizadas por el usuario (es decir, las URI a las cuáles el usuario acceda por el navegador), serán manejadas por un único archivo, que estará almacenado en nuestro directorio departamentosapp.

```
echo '# -*- coding: utf-8 -*-' > departamentosapp/controller.py
```

```
usuario@usuario-VirtualBox:~/departamentos_py/trunk/departamentos-web/departamentosapp$ sudo nano contro
ller.py
[sudo] contraseña para usuario:
```

Una vez generado, lo editamos con con nano introduciendo lo siguiente:

```
def application(environ, start_response):
    # Genero la salida HTML a mostrar al usuario
    output = "<p>Bienvenido a <b>Departamentos de IES La Marisma</b>!!!</p>"
    # Inicio una respuesta al navegador
    start_response('200 OK', [('Content-Type', 'text/html; charset=utf-8')])
    # Retorno el contenido HTML
    return output
```

```

GNU nano 4.8                               controller.py
-*- coding: utf-8 -*-

def application(environ, start_response):
    # Genero la salida HTML a mostrar al usuario
    output = "<p>Bienvenido a <b>Departamentos de IES La Marisma</b>!!!</p>"
    # Inicio una respuesta al navegador
    start_response('200 OK', [('Content-Type', 'text/html; charset=utf-8')])
    # Retorno el contenido HTML
    return output

```

Por último, muevo mis directorios hacia /var/www/ para poder configurar y servir bajo la configuración de Apache. El acceso a la ruta da problemas en el directorio raíz.

```
sudo mv departamentos_py /var/www/
```

```

usuario@usuario-VirtualBox:/var/www$ ll
total 20
drwxr-xr-x  5 root      root      4096 nov 24 18:01 .
drwxr-xr-x 15 root      root      4096 nov 22 19:38 ../
drwxr-x---  5 www-data  www-data  4096 nov 23 20:23 centro.intranet/
drwxrwxr-x  4 usuario   usuario   4096 nov 24 18:14 departamentos_py/
drwxr-xr-x  3 root      root      4096 nov 24 19:08 html/
usuario@usuario-VirtualBox:/var/www$
```

Ahora toca **configurar el VirtualHost**:

Mientras que el DocumentRoot de nuestro sitio Web, será la carpeta pública, public_html, una variable del VirtualHost, será la encargada de redirigir todas las peticiones públicas del usuario, hacia nuestro front controller. Y la variable que se encargue de esto, será el alias WSGIScriptAlias:

```
sudo nano /etc/apache2/sites-available/departamentos.centro.intranet.conf
```

Una vez allí, escribimos el contenido del nuevo virtual host:

```

<VirtualHost *:80>
    ServerName departamentos.centro.intranet
    DocumentRoot /var/www/departamentos_py/trunk/departamentos-web/public_html
    WSGIScriptAlias /var/www/departamentos_py/trunk/departamentos-web/departamentosapp/controller.py
        ErrorLog /var/www/departamentos_py/logs/errors.log
        CustomLog /var/www/departamentos_py/logs/access.log combined

    <Directory />
        Options FollowSymLinks
        AllowOverride All
    </Directory>
</VirtualHost>
```

```
GNU nano 4.8      /etc/apache2/sites-available/departamentos.centro.intranet.conf
<VirtualHost *:80>
    ServerName departamentos.centro.intranet
    DocumentRoot /var/www/departamentos_py/trunk/departamentos-web/public_html
    WSGIScriptAlias / /var/www/departamentos_py/trunk/departamentos-web/departamentosapp/controller.py
    ErrorLog /var/www/departamentos_py/logs/errors.log
    CustomLog /var/www/departamentos_py/logs/access.log combined

    <Directory />
        Options FollowSymLinks
        AllowOverride All
    </Directory>
</VirtualHost>
```

Una vez configurado nuestro VirtualHost:

- Habilitamos el sitio web: sudo a2ensite departamentos.centro.intranet

```
usuario@usuario-VirtualBox:/etc/apache2/sites-available$ sudo a2ensite departamentos.centro.intranet
Enabling site departamentos.centro.intranet.
To activate the new configuration, you need to run:
    systemctl reload apache2
usuario@usuario-VirtualBox:/etc/apache2/sites-available$ sudo systemctl reload apache2
u Mostrar aplicaciones rtualBox:/etc/apache2/sites-available$ sudo nano /etc/hosts
u _ rtualBox:/etc/apache2/sites-available$
```

- Recargamos Apache: `sudo systemctl reload apache2`
- Comprobamos el sitio en nuestro host: `sudo nano /etc/hosts` y si no existiera allí, agregamos la siguiente línea:

```
127.0.0.1      departamentos.centro.intranet
```

```
GNU nano 4.8
127.0.0.1      localhost
127.0.1.1      usuario-VirtualBox
127.0.0.1      centro.intranet
127.0.0.1      departamentos.centro.intranet

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

- A partir de ahora, si abrimos nuestro navegador Web e ingresamos la url <http://departamentos.centro.intranet>, veremos la frase: "Bienvenido a Departamentos de IES La Marisma!!!".



4. PROTEGER LA APLICACIÓN PYTHON MEDIANTE AUTENTICACIÓN.

La autenticación es cualquier proceso mediante el cual se verifica que alguien es quien dice ser. La autorización es cualquier proceso por el cual a alguien se le permite estar donde quiere ir, o tener la información que quiere tener.

Necesitará crear un archivo de contraseñas. Éste archivo debería colocarlo en algún sitio no accesible mediante la Web. Por ejemplo, si sus documentos son servidos desde /usr/local/apache/htdocs usted podría querer colocar el(los) archivo(s) de contraseñas en /usr/local/apache/passwd.

Para **crear un archivo de contraseñas**, use la utilidad htpasswd que viene con Apache. Esta utilidad puede encontrarla en el directorio bin de cualquier sitio en que haya instalado Apache. Para crear el archivo, escriba:

```
sudo htpasswd -c /usr/local/apache/passwd/passwords andres
```

```
usuario@usuario-VirtualBox:/usr/local/apache/passwd$ sudo htpasswd -c /usr/local/apache/passwd/passwords andres
New password:
Re-type new password:
Adding password for user andres
usuario@usuario-VirtualBox:/usr/local/apache/passwd$
```

El siguiente paso es **configurar el servidor para que solicite una contraseña y decirle al servidor a qué usuarios se les permite el acceso**. Puede hacer esto editando el archivo apache2.conf. Para proteger el directorio /var/www/departamentos_py, puede usar las siguientes directivas en apache2.conf dentro de una sección <Directory /var/www/html/departamentos.centro.intranet>

```
sudo nano /etc/apache2/apache2.conf
```

```
<Directory "/var/www/departamentos_py">
    AuthType Basic
    AuthName "Restricted Files"
    # (Following line optional)
    AuthBasicProvider file
    AuthUserFile "/usr/local/apache/passwd/passwords"
    Require user andres
</Directory>
```

```
GNU nano 4.8                               /etc/apache2/apache2.conf
    AllowOverride None
        Require all denied
</Directory>

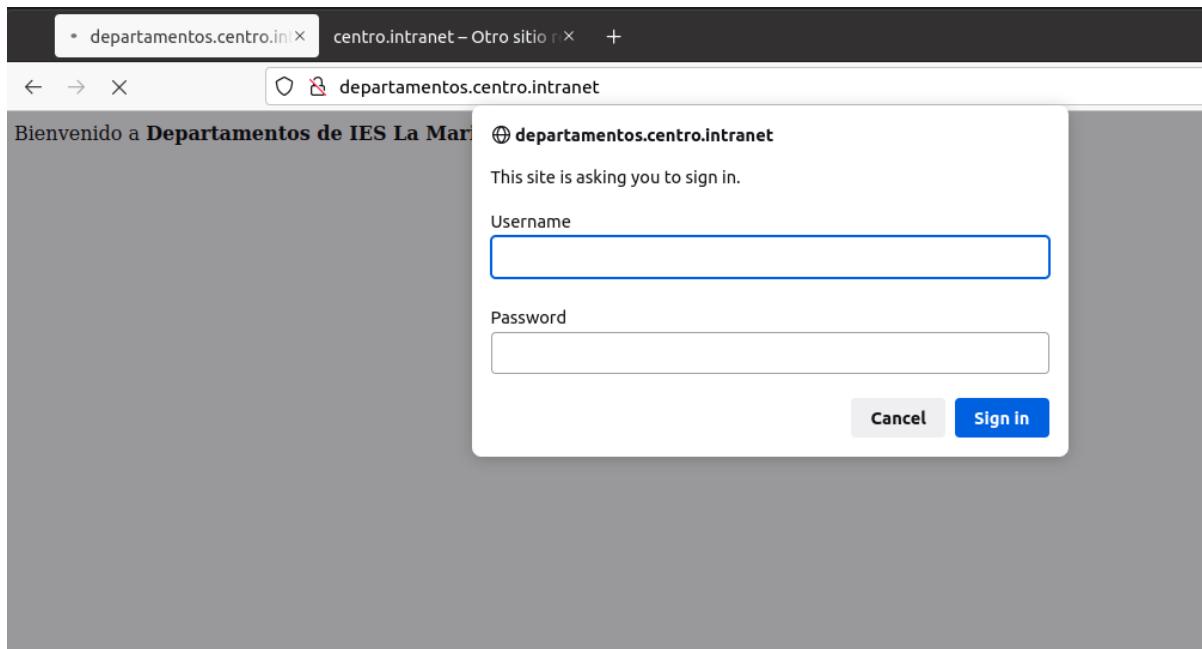
<Directory /usr/share>
    AllowOverride None
        Require all granted
</Directory>

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

<Directory "/var/www/departamentos_py">
    AuthType Basic
    AuthName "Restricted Files"
    # (Following line optional)
    AuthBasicProvider file
    AuthUserFile "/usr/local/apache/passwd/passwords"
    Require user andres
</Directory>

#<Directory /srv/>
#    Options Indexes FollowSymLinks
```

Nuestra aplicación python estaría protegida con autenticación:



5. INSTALA Y CONFIGURA AWSTATS.

AWStats es una herramienta de generación de informes de analítica web de código abierto que genera gráficamente estadísticas avanzadas de web, streaming, FTP o servidor de correo. Este analizador de registros funciona como un CGI o desde la línea de comandos y le muestra toda la información posible que su registro contiene en unas cuantas páginas web gráficas. Utiliza un archivo de información parcial para poder procesar archivos de registro grandes con frecuencia y rapidez. Soporta la mayoría de los formatos de archivo de registro del servidor web, incluyendo Apache, IIS y muchos otros formatos de registro del servidor web. **Se instala ejecutando:**

```
sudo apt-get install awstats
```

```
usuario@usuario-VirtualBox:/etc/apache2$ sudo apt-get install awstats
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libnet-xwhois-perl
Paquetes sugeridos:
  libnet-dns-perl libnet-ip-perl libgeo-ipfree-perl
Se instalarán los siguientes paquetes NUEVOS:
  awstats libnet-xwhois-perl
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 125 no actualizados.
Se necesita descargar 1.861 kB de archivos.
Se utilizarán 7.057 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
```

A continuación, deberás **habilitar el módulo CGI en Apache**. Puedes hacerlo corriendo:

```
sudo a2enmod cgi
```

Y reinicia Apache para reflejar los cambios.

```
sudo systemctl restart apache2
```

```
usuario@usuario-VirtualBox:/etc/apache2$ sudo a2enmod cgi
Enabling module cgi.
To activate the new configuration, you need to run:
  systemctl restart apache2
usuario@usuario-VirtualBox:/etc/apache2$ sudo systemctl restart apache2
usuario@usuario-VirtualBox:/etc/apache2$ █
```

Para **configurar AWStats**, se debe crear un archivo de configuración para cada dominio o sitio web del que deseé ver estadísticas. En este ejemplo crearemos un archivo de configuración para » centro.intranet « y para » departamentos.centro.intranet «.

Puede hacer esto duplicando el archivo de configuración por defecto de AWStats a uno con su nombre de dominio:

```
sudo cp /etc/awstats/awstats.conf /etc/awstats/awstats.centro.intranet.conf
```

y

```
sudo cp /etc/awstats/awstats.conf /etc/awstats/awstats.departamentos.centro.intranet.conf
```

```
usuario@usuario-VirtualBox:/etc/apache2$ sudo cp /etc/awstats/awstats.conf  
/etc/awstats.awstats.centro.intranet.conf  
usuario@usuario-VirtualBox:/etc/apache2$ sudo cp /etc/awstats/awstats.conf  
/etc/awstats.awstats.departamentos.centro.intranet.conf  
usuario@usuario-VirtualBox:/etc/apache2$
```

Ahora, necesitas hacer algunos **cambios en el archivo de configuración**:

```
sudo nano /etc/awstats/awstats.centro.intranet.conf
```

y

```
sudo nano /etc/awstats/awstats.departamentos.centro.intranet.conf
```

Para centro.intranet:

```
# Cambiar al archivo de registro de Apache, por defecto es  
/var/log/apache2/access.log  
LogFile="/var/log/apache2/access.log"  
# Cambiar el nombre de dominio del sitio web  
SiteDomain="centro.intranet"  
HostAliases="centros.intranet localhost 127.0.0.0.1"  
# Cuando este parámetro se establece en 1, AWStats añade un botón en la página  
del informe para permitir «actualizar» las estadísticas desde un navegador web.  
AllowToUpdateStatsFromBrowser=1
```

```
, if there are several log files from logs  
# Example: "/path/to/tools/logresolvemerge.p  
#  
LogFile="/var/log/apache2/access.log"  
  
# Enter the log file type you want to anal  
# Possible values:
```

```

# Example: "ftp.domain.com"
# Example: "domain.com"
#
SiteDomain="centro.intranet"

#
# Enter here all other possible
# aliases someone can use to acc
# number of possible names/addr

#
# Note: You can also use @/mypath/myfile if list of .
# Example: "www.myserver.com localhost 127.0.0.1 REG"
#
HostAliases="centros.intranet localhost 127.0.0.1"

#
# If you want to have hosts reported by name instead
# of IP address, set this to 1
# Possible values: 0 or 1
# Default: 0
#
AllowToUpdateStatsFromBrowser=1

#
# AWStats saves and sorts its data

```

Para departamentos.centro.intranet:

```

LogFile="/var/log/apache2/access.log"
SiteDomain="departamentos.centro.intranet"
HostAliases="departamentos.centros.intranet localhost 127.0.0.1"
AllowToUpdateStatsFromBrowser=1

```

Guarda y cierre el archivo.

Después de estos cambios,necesitas **construir tus estadísticas iniciales** que se generarán a partir de los registros actuales en tu servidor. Puedes hacerlo utilizando:

```
sudo /usr/lib/cgi-bin/awstats.pl -config=centro.intranet -update
```

y

```
sudo /usr/lib/cgi-bin/awstats.pl -config=centro.intranet -update
```

```
usuario@usuario-VirtualBox:/usr/lib/cgi-bin$ sudo /usr/lib/cgi-bin/awstats.pl -config=centro.intranet -update
Create/Update database for config "/etc/awstats/awstats.centro.intranet.conf" by AWStats version 7.6 (build 20161204)
From data in log file "/var/log/apache2/access.log"...
Phase 1 : First bypass old records, searching new record...
Searching new records from beginning of log file...
Phase 2 : Now process new records (Flush history on disk after 20000 hosts)...
Reverse DNS lookup for ::1 not available without ipv6 plugin enabled.
Jumped lines in file: 0
Parsed lines in file: 244
Found 0 dropped records,
Found 0 comments,
Found 0 blank records,
Found 0 corrupted records,
Found 0 old records,
Found 244 new qualified records.
usuario@usuario-VirtualBox:/usr/lib/cgi-bin$
```

```
usuario@usuario-VirtualBox:/usr/lib/cgi-bin$ sudo /usr/lib/cgi-bin/awstats.pl -config=departamentos.centro.intranet -update
Create/Update database for config "/etc/awstats/awstats.departamentos.centro.intranet.conf" by AWStats version 7.6 (build 20161204)
From data in log file "/var/log/apache2/access.log"...
Phase 1 : First bypass old records, searching new record...
Searching new records from beginning of log file...
Phase 2 : Now process new records (Flush history on disk after 20000 hosts)...
Reverse DNS lookup for ::1 not available without ipv6 plugin enabled.
Jumped lines in file: 0
Parsed lines in file: 244
Found 0 dropped records,
Found 0 comments,
Found 0 blank records,
Found 0 corrupted records,
Found 0 old records,
Found 244 new qualified records.
```

El siguiente paso a realizar será **configurar Apache para AWStats**:

A continuación, debe configurar Apache2 para que muestre estas estadísticas. Ahora copie el contenido de la carpeta «cgi-bin» en el directorio raíz del documento por defecto de su instalación de Apache. Por defecto se encuentra en la carpeta «/usr/lib/cgi-bin».

Puede hacerlo corriendo:

```
sudo cp -r /usr/lib/cgi-bin /var/www/html/
sudo chown www-data:www-data /var/www/html/cgi-bin/
sudo chmod -R 755 /var/www/html/cgi-bin/
```

```
usuario@usuario-VirtualBox:/usr/lib/cgi-bin$ sudo cp -r /usr/lib/cgi-bin /var/www/html/
usuario@usuario-VirtualBox:/usr/lib/cgi-bin$ sudo chown www-data:www-data /var/www/html/cgi-bin/
usuario@usuario-VirtualBox:/usr/lib/cgi-bin$ sudo chmod -R 755 /var/www/html/cgi-bin/
usuario@usuario-VirtualBox:/usr/lib/cgi-bin$
```

Prueba AWStats:

Ahora puedes acceder a tus AWStats visitando la url:

<http://centro.intranet/cgi-bin/awstats.pl?config=centro.intranet>

y

<http://departamentos.centro.intranet/cgi-bin/awstats.pl?config=departamentos.centro.intranet>

Le mostrará una página de resultados como esta:

Para centro.intranet:

The screenshot shows the AWStats summary page for the domain `centro.intranet`. The top navigation bar includes links for `Statistics for: centro.intranet`, `Last Update: 24 Nov 2021 - 19:06`, `Update now`, `Reported period: Nov 2021`, and `OK`. The main content area has a **Summary** tab selected, showing traffic statistics for November 2021. The table includes columns for Viewed traffic and Not viewed traffic, with detailed metrics like Unique visitors, Number of visits, Pages, Hits, and Bandwidth.

	Unique visitors	Number of visits	Pages	Hits	Bandwidth
Viewed traffic *	2	(1.5 visits/visitor)	(32.33 Pages/Visit)	(75 Hits/Visit)	1.62 MB (552.42 KB/Visit)
Not viewed traffic *		8	19		4.30 KB

* Not viewed traffic includes traffic generated by robots, worms, or replies with special HTTP status codes.

Below the summary, there is a **Monthly history** section showing traffic data for each month from January to October 2021. The table has columns for Month, Unique visitors, Number of visits, Pages, Hits, and Bandwidth, all showing zero values.

Month	Unique visitors	Number of visits	Pages	Hits	Bandwidth
Jan 2021	0	0	0	0	0
Feb 2021	0	0	0	0	0
Mar 2021	0	0	0	0	0
Apr 2021	0	0	0	0	0
May 2021	0	0	0	0	0
Jun 2021	0	0	0	0	0
Jul 2021	0	0	0	0	0
Aug 2021	0	0	0	0	0
Sep 2021	0	0	0	0	0
Oct 2021	0	0	0	0	0

Y para departamentos.centro.intranet:

The screenshot shows the AWStats summary page for the domain `departamentos.centro.intranet`. The top navigation bar includes links for `Statistics for: departamentos.centro.intranet`, `Last Update: 24 Nov 2021 - 19:07`, `Update now`, `Reported period: Nov 2021`, and `OK`. The main content area has a **Summary** tab selected, showing traffic statistics for November 2021. The table includes columns for Viewed traffic and Not viewed traffic, with detailed metrics like Unique visitors, Number of visits, Pages, Hits, and Bandwidth.

	Unique visitors	Number of visits	Pages	Hits	Bandwidth
Viewed traffic *	2	(1.5 visits/visitor)	(32.33 Pages/Visit)	(75 Hits/Visit)	1.62 MB (552.42 KB/Visit)
Not viewed traffic *		8	19		4.30 KB

* Not viewed traffic includes traffic generated by robots, worms, or replies with special HTTP status codes.

Below the summary, there is a **Monthly history** section showing traffic data for each month from January to April 2021. The table has columns for Month, Unique visitors, Number of visits, Pages, Hits, and Bandwidth, all showing zero values.

Month	Unique visitors	Number of visits	Pages	Hits	Bandwidth
Jan 2021	0	0	0	0	0
Feb 2021	0	0	0	0	0
Mar 2021	0	0	0	0	0
Apr 2021	0	0	0	0	0

Por último y para dejar los Awstats lo mejor posible, **configuramos “Cron” para actualizar los registros:**

Se recomienda programar un trabajo Cron para actualizar regularmente la base de datos de AWStats usando entradas de registro recién creadas, para que las estadísticas se actualicen regularmente. Esto también te ahorrará tiempo.

Para ello es necesario editar el fichero «/etc/crontab»:

```
sudo nano /etc/crontab
```

Añade la siguiente línea que le dice a AWStats que actualice cada diez minutos:

```
*/10 * * * * root /usr/lib/cgi-bin/awstats.pl -config=centro.intranet -update
*/10 * * * * root /usr/lib/cgi-bin/awstats.pl -config=departamentos.centro.intranet
-updated
```

Guarde y cierre el archivo.

```
GNU nano 4.8                               /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the 'crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# ----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .---- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .--- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | |
# * * * * * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
*/10 * * * * root /usr/lib/cgi-bin/awstats.pl -config=centro.intranet -update
*/10 * * * * root /usr/lib/cgi-bin/awstats.pl -config=departamentos.centro.intranet -update
```

6. INSTALA UN SEGUNDO SERVIDOR DE TU ELECCIÓN (NGINX, LIGHTTPD) BAJO EL DOMINIO “servidor2.centro.intranet”. DEBES CONFIGURARLO PARA QUE SIRVA EN EL PUERTO 8080 Y HAZ LOS CAMBIOS NECESARIOS PARA EJECUTAR PHP. INSTALA PHPMYADMIN.

a) Instalar Nginx en Ubuntu 20.04:

Nginx es uno de los servidores web más populares del mundo y aloja algunos de los sitios más grandes y con mayor tráfico en Internet. Es una opción ligera que se puede utilizar como servidor web o proxy inverso. Debemos tener un non-root user normal con privilegios sudo configurado en su servidor.

Paso 1: Instalar Nginx

Debido a que Nginx está disponible en los repositorios predeterminados de Ubuntu, es posible instalarlo desde estos repositorios usando el sistema de paquetes apt.

Ya que esta es nuestra primera interacción con el sistema de paquetes apt en esta sesión, actualizaremos nuestro índice local de paquetes de modo que tengamos acceso a los listados de paquetes más recientes. A continuación, podremos instalar nginx:

```
sudo apt update
```

```
usuario@usuario-VirtualBox:/var/www$ sudo apt update
Des:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Obj:2 http://es.archive.ubuntu.com/ubuntu focal InRelease
Des:3 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Des:5 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1 246
```

Para instalar Nginx, ejecutamos:

```
sudo apt install nginx
```

```
se pueden actualizar los paquetes. Ejecute «apt list --upgradable»
usuario@usuario-VirtualBox:/var/www$ sudo apt install nginx
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter l
  nginx-common nginx-core
```

Tras aceptar el procedimiento, apt instalará Nginx y cualquier dependencia necesaria en su servidor.

Paso 2: Aplicar ajustes al firewall

Antes de probar Nginx, se deben aplicar ajustes al software del firewall para permitir el acceso al servicio. Nginx se registra de forma automática como un servicio con ufw tras la instalación, lo que hace que sea sencillo permitir el acceso de Nginx.

Enumere las configuraciones de la aplicación con las que ufw sabe trabajar escribiendo lo siguiente:

```
sudo ufw app list
```

```
usuari@usuario-VirtualBox:/var/www$ sudo ufw app list
Aplicaciones disponibles:
 Apache
 Apache Full
 Apache Secure
 CUPS
 Nginx Full
 Nginx HTTP
 Nginx HTTPS
```

Como se muestra en el resultado, hay tres perfiles disponibles para Nginx:

- Nginx Full: este perfil abre el puerto 80 (tráfico web normal, no cifrado) y el puerto 443 (tráfico TLS/SSL cifrado)
- Nginx HTTP: este perfil abre solo el puerto 80 (tráfico web normal, no cifrado)
- Nginx HTTPS: este perfil abre solo el puerto 443 (tráfico TLS/SSL cifrado)

Se recomienda habilitar el perfil más restrictivo, que de todos modos permitirá el tráfico que se configuró. En este momento, solo tendremos que permitir el tráfico en el puerto 80.

Puedes habilitarlo escribiendo lo siguiente:

```
sudo ufw allow 'Nginx HTTP'
```

Puede verificar el cambio escribiendo lo siguiente:

```
sudo ufw status
```

El resultado indicará el tráfico de HTTP que se permite:

```
usuario@usuario-VirtualBox:/var/www$ sudo ufw allow 'Nginx HTTP'
Regla añadida
Regla añadida (v6)
usuario@usuario-VirtualBox:/var/www$ sudo ufw status
Estado: activo

Hasta           Acción      Desde
----           -----      -----
Apache          ALLOW       Anywhere
Nginx HTTP     ALLOW       Anywhere
Apache (v6)    ALLOW       Anywhere (v6)
Nginx HTTP (v6) ALLOW       Anywhere (v6)
```

Paso 3: Comprobar su servidor web

Al final del proceso de instalación, Ubuntu 20.04 inicia Nginx. El servidor web ya debería estar activo.

Realice una verificación con systemctl init para asegurarse de que el servicio esté en ejecución escribiendo lo siguiente:

```
sudo systemctl status nginx
```

```
usuario@usuario-VirtualBox:/var/www$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
  Active: inactive (dead)
    Docs: man:nginx(8)
```

El servicio, por lo que se ve, no ha iniciado. Vamos a parar Apache, ya que nos ha creado conflicto:

```
sudo systemctl stop apache2
sudo systemctl start nginx
```

```
usuario@usuario-VirtualBox:/etc/apache2/sites-enabled$ sudo systemctl start nginx
usuario@usuario-VirtualBox:/etc/apache2/sites-enabled$ systemctl status nginx.service
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2021-11-24 20:09:20 CET; 2min 19s ago
    Docs: man:nginx(8)
   Process: 11641 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, st>
   Process: 11642 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SU>
 Main PID: 11643 (nginx)
   Tasks: 2 (limit: 2299)
  Memory: 2.3M
 CGroup: /system.slice/nginx.service
         └─11643 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             ├─11644 nginx: worker process

nov 24 20:09:20 usuario-VirtualBox systemd[1]: Starting A high performance web server and a reverse pro>
[Mostrar aplicaciones]
```

Como lo confirma este resultado, el servicio se inició correctamente. Sin embargo, la mejor forma de comprobarlo es solicitar una página de Nginx.

Puede acceder a la página de aterrizaje predeterminada de Nginx para confirmar que el software funcione correctamente dirigiéndose a la dirección IP de su servidor. Si no conoce la dirección IP de su servidor, puede buscarla con la herramienta icanhazip.com, que le proporcionará su dirección IP pública tal como la recibió de otra ubicación en Internet:

```
curl -4 ianhazip.com
```

```
usuario@usuario-VirtualBox:/etc/apache2/sites-enabled$ curl -4 ianhazip.com
193.178.46.91
```

Cuando tenga la dirección IP de su servidor, introducuela en la barra de direcciones de su navegador:

<http://193.178.46.91>

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Paso 4: Configurar bloques de servidor (recomendado)

Al emplear el servidor web Nginx, se pueden utilizar bloques de servidor (similares a hosts virtuales de Apache) para encapsular los detalles de la configuración y alojar más de un dominio desde un único servidor. Configuraremos un dominio llamado `servidor2.centro.intranet`. Para obtener más información sobre cómo configurar un nombre de dominio con DigitalOcean, consulte nuestra Introducción al DNS de DigitalOcean.

Nginx en Ubuntu 20.04 tiene habilitado un bloque de servidor por defecto, que está configurado para suministrar documentos desde un directorio en `/var/www/html`. Si bien esto funciona bien para un solo sitio, puede ser difícil de manejar si aloja varios. En vez de modificar `/var/www/html`, vamos a crear una estructura de directorios dentro de `/var/www`

para nuestro sitio servidor2.centro.intranet y dejaremos /var/www/html como directorio predeterminado que se suministrará si una solicitud de cliente no coincide con otros sitios.

Cree el directorio para servidor2.centro.intranet como se muestra a continuación, usando el indicador -p para crear cualquier directorio principal necesario:

```
sudo mkdir -p /var/www/servidor2.centro.intranet/html
```

```
usuario@usuario-VirtualBox:/var/www/html$ sudo mkdir -p /var/www/servidor2.centro.intranet/html
usuario@usuario-VirtualBox:/var/www/html$ cd /var/www
usuario@usuario-VirtualBox:/var/www$ ll
total 24
drwxr-xr-x  6 root      root      4096 nov 24 21:51 .
drwxr-xr-x 15 root      root      4096 nov 22 19:38 ..
drwxr-x---  5 www-data  www-data  4096 nov 23 20:23 centro.intranet/
drwxrwxr-x  4 usuario   usuario   4096 nov 24 18:14 departamentos_py/
drwxr-xr-x  3 root      root      4096 nov 24 21:49 html/
drwxr-xr-x  3 root      root      4096 nov 24 21:51 servidor2.centro.intranet/
usuario@usuario-VirtualBox:/var/www$
```

A continuación, asigne la propiedad del directorio con la variable de entorno \$USER:

```
sudo chown -R $USER:$USER /var/www/servidor2.centro.intranet/html
```

```
usuario@usuario-VirtualBox:/var/www$ sudo chown -R $USER:$USER /var/www/servidor2.centro.intranet/html
```

Los permisos de los roots web deberían ser correctos si no se modificó el valor umask, que establece permisos de archivos predeterminados. Para asegurarse de que sus permisos sean correctos y permitir al propietario leer, escribir y ejecutar los archivos, y a la vez conceder solo permisos de lectura y ejecución a los grupos y terceros, puede ingresar el siguiente comando:

```
sudo chmod -R 755 /var/www/servidor2.centro.intranet
```

```
usuario@usuario-VirtualBox:/var/www$ sudo chmod -R 755 /var/www/servidor2.centro.intranet
```

A continuación, cree una página de ejemplo index.html utilizando nano o su editor favorito:

```
sudo nano /var/www/servidor2.centro.intranet/html/index.html
```

```
<html>
  <head>
    <title>Bienvenido al Servidor 2 de IES La Marisma!</title>
  </head>
  <body>
    <h1>Si ves esto, es que el bloque de dominio ha funcionado correctamente</h1>
  </body>
</html>
```

```
/var/www/servidor2.centro.intranet/html/index.html
<html>
  <head>
    <title>Bienvenido al Servidor 2 de IES La Marisma!<
  </head>
  <body>
    <h1>Si ves esto, es que el bloque de dominio ha fun
  </body>
</html>

[ 8 líneas escritas ]
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Text^J
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar ^T
```

Cuando termine, escriba CTRL y X, y luego, Y y ENTER, para guardar y cerrar el archivo.

Para que Nginx presente este contenido, es necesario crear un bloque de servidor con las directivas correctas. En vez de modificar el archivo de configuración predeterminado directamente, crearemos uno nuevo en /etc/nginx/sites-available/servidor2.centro.intranet:

```
sudo nano /etc/nginx/sites-available/servidor2.centro.intranet
```

E introducimos:

```
server {
  listen 8080;
  listen [::]:8080;

  root /var/www/servidor2.centro.intranet/html;
  index index.html index.htm index.nginx-debian.html;

  server_name servidor2.centro.intranet www.servidor2.centro.intranet;

  location / {
    try_files $uri $uri/ =404;
  }
}
```

```
GNU nano 4.8                               /etc/nginx/sites-available/servidor2.centro.intranet
server {
    listen 8080;
    listen [::]:8080;

    root /var/www/servidor2.centro.intranet/html;
    index index.html index.htm index.nginx-debian.html;

    server_name servidor2.centro.intranet www.servidor2.centro.intranet;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

Observe que actualizamos la configuración root en nuestro nuevo directorio y el server_name para nuestro nombre de dominio.

A continuación, habilitaremos el archivo creando un enlace entre él y el directorio sites-enabled, en el cual Nginx obtiene lecturas durante el inicio:

```
sudo ln -s /etc/nginx/sites-available/servidor2.centro.intranet /etc/nginx/sites-enabled/
```

```
usuario@usuario-VirtualBox:/var/www$ sudo ln -s /etc/nginx/sites-available/servidor2.centro.intranet /etc/nginx/sites-enabled/
```

Ahora, contamos con dos bloques de servidor habilitados y configurados para responder a las solicitudes conforme a las directivas listen y servidor2.centro.intranet.

Para evitar un problema de memoria de depósito de hash que pueda surgir al agregar nombres de servidor, es necesario aplicar ajustes a un valor en el archivo /etc/nginx/nginx.conf. Abra el archivo:

```
sudo nano /etc/nginx/nginx.conf
```

Encuentre la directiva server_names_hash_bucket_size y borre el símbolo # para eliminar el comentario de la línea.

```
GNU nano 4.8                               /etc/nginx/nginx.conf
http {

    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    # server_tokens off;

    server_names_hash_bucket_size 64;
# server_name_in_redirect off;
```

Guarde y cierre el archivo cuando termine.

A continuación, compruebe que no haya errores de sintaxis en ninguno de sus archivos de Nginx:

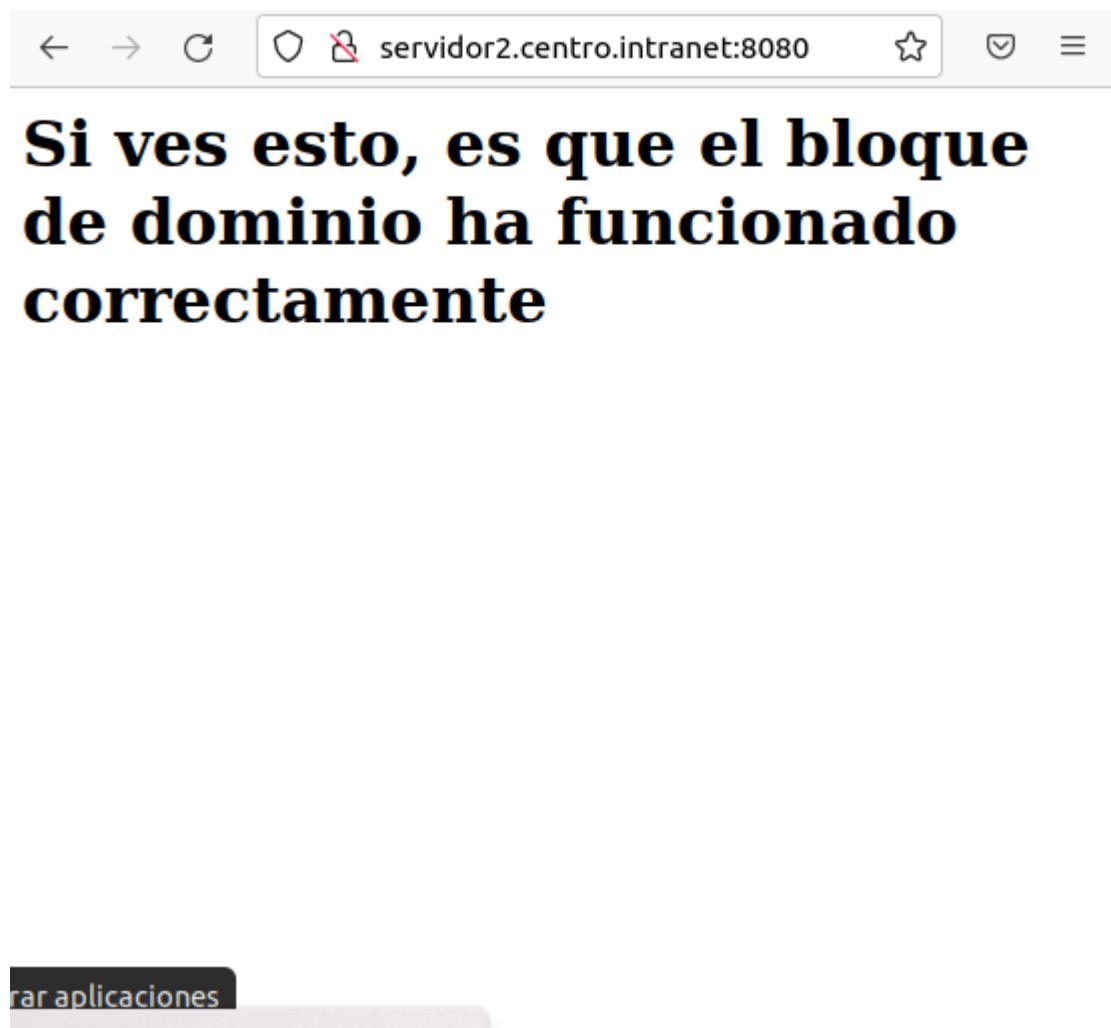
```
sudo nginx -t
```

```
usuario@usuario-VirtualBox:/var/www$ sudo nano /etc/nginx/nginx.conf
usuario@usuario-VirtualBox:/var/www$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
usuario@usuario-VirtualBox:/var/www$
```

Si no hay problemas, reinicie Nginx para habilitar los cambios:

```
sudo systemctl restart nginx
```

Con esto, Nginx debería proporcionar su nombre de dominio. Puede probarlo visitando <http://servidor2.centro.intranet:8080>, donde debería ver algo como esto:



Paso 5: Configurar Nginx para utilizar el procesador PHP:

Al emplear el servidor web Nginx, podemos crear bloques de servidor (similares a los hosts virtuales de Apache) para encapsular los detalles de configuración y alojar más de un dominio en un único servidor. En esta guía, utilizaremos servidor2.centro.intranet

Primero, vamos a modificar el archivo de configuración en el directorio sites-available de Nginx:

```
sudo nano /etc/nginx/sites-available/servidor2.centro.intranet
```

Y lo ajustamos así:

```
server {
    listen 8080;
    server_name servidor2.centro.intranet www.servidor2.centro.intranet;
    root /var/www/servidor2.centro.intranet;

    index index.html index.htm index.php;

    location / {
        try_files $uri $uri/ =404;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
    }

    location ~ /\.ht {
        deny all;
    }
}

GNU nano 4.8          /etc/nginx/sites-available/servidor2.centro.intranet
```

Guardamos y comprobamos que no tenemos errores de sintaxis con el siguiente comando:

```
sudo nginx -t
```

```
ya existe
usuario@usuario-VirtualBox:/etc/nginx/sites-available$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Cuando esté listo, vuelva a cargar Nginx para aplicar los cambios:

```
sudo systemctl reload nginx
```

Ahora, su nuevo sitio web está activo, pero el directorio root web /var/www/servidor2.centro.intranet todavía está vacío. Cree un archivo index.html en esa ubicación para poder probar que el nuevo bloque del servidor funcione según lo previsto:

```
sudo nano /var/www/servidor2.centro.intranet/index.html
```

Y escribimos el contenido:

```
<html>
<head>
    <title>Dominio del Servidor 2 de IES La Marisma</title>
</head>
<body>
    <h1>Bienvenido a IES La Marisma!</h1>
```

```
<p>Estamos aprendiendo a configurar PHP en <strong>Nginx</strong>.</p>
</body>
</html>
```

```
GNU nano 4.8                               /var/www/servidor2.centro.intranet/index.html
<html>
    <head>
        <title>Dominio del Servidor 2 de IES La Marisma</title>
    </head>
    <body>
        <h1>Bienvenido a IES La Marisma!</h1>

        <p>Estamos aprendiendo a configurar PHP en <strong>Nginx</strong>.</p>
    </body>
</html>
```

Ahora, diríjase al navegador y acceda a <http://servidor2.centro.intranet:8080>



Bienvenido a IES La Marisma!

Estamos aprendiendo a configurar PHP en **Nginx**.

Si ve esta página, su bloque de servidor de Nginx está funcionando según lo previsto. Lo siguiente que tenemos que hacer es crear una secuencia de comandos PHP para probar que Nginx, de hecho, puede gestionar los archivos .php en el sitio web que recién configuramos.

Para hacerlo, cree un archivo PHP de prueba en el root de tu documento. En el editor de texto, abra un archivo nuevo denominado info.php en el root de su documento:

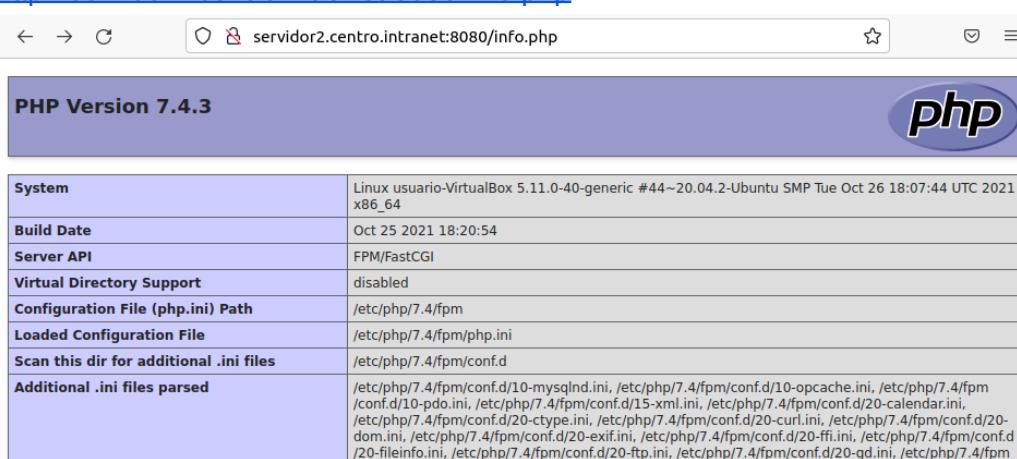
```
nano /var/www/servidor2.centro.intranet/info.php
```

```
<?php  
phpinfo();
```

```
GNU nano 4.8 /var/www/servidor2.centro.intranet/info.php
<?php
phpinfo();
```

Cuando termine, guarde y cierre el archivo escribiendo CTRL+X, luego y, y ENTER para confirmar.

Ahora, puede acceder a esta página en el navegador web al visitar <http://servidor2.centro.intranet:8080/info.php>



Ahora, por último, toca instalar PHPMyAdmin:

Actualizamos el índice de paquetes del servidor:

```
sudo apt update
```

```
usuario@usuario-VirtualBox:/etc/nginx/sites-available$ sudo apt update
[sudo] contraseña para usuario:
Obj:1 http://es.archive.ubuntu.com/ubuntu focal InRelease
Des:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Des:5 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [35,7 kB]
Des:6 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [564 kB]
Des:7 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1.346 kB]
Des:8 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [64,4 kB]
Des:9 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [2.464 B]
Des:10 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [278 kB]
Des:11 http://es.archive.ubuntu.com/ubuntu focal-updates/universe i386 Packages [648 kB]
Des:12 http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [876 kB]
Des:13 http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [357 kB]
Des:14 http://es.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [944 B]
Des:15 http://es.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [10,4 kB]
Descargados 4.510 kB en 2s (1.973 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se pueden actualizar 130 paquetes. Ejecute «apt list --upgradable» para verlos.
```

Ejecuta el siguiente comando para instalar los paquetes necesarios para instalar PHPMyAdmin:

```
sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl
```

```
Se pueden actualizar 130 paquetes. Ejecute «apt list --upgradable» para verlos.
usuario@usuario-VirtualBox:/etc/nginx/sites-available$ sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
php-curl ya está en su versión más reciente (2:7.4+75).
php-gd ya está en su versión más reciente (2:7.4+75).
php-mbstring ya está en su versión más reciente (2:7.4+75).
```

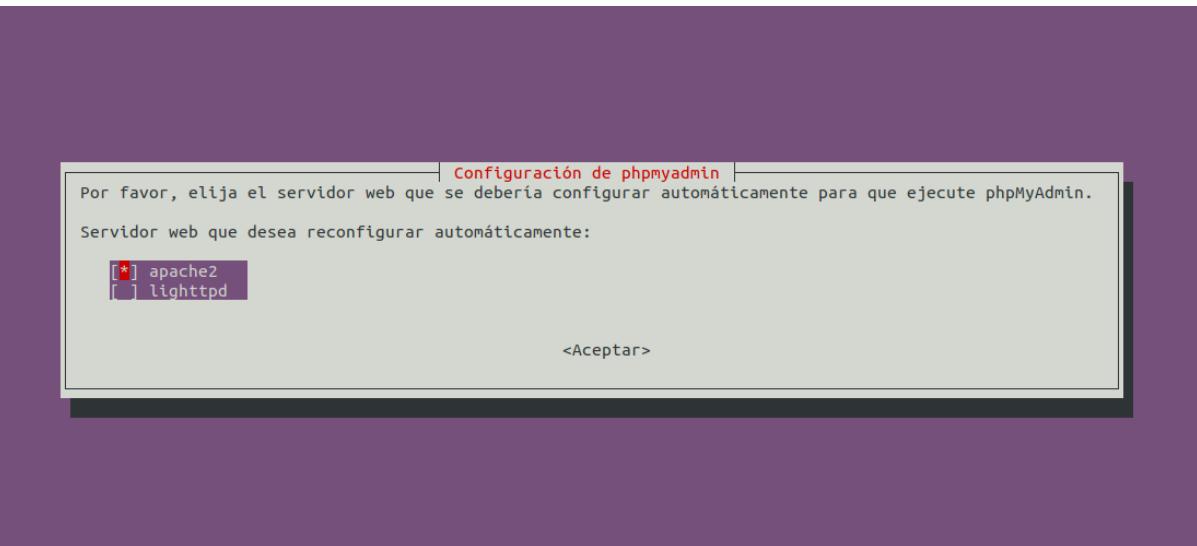
Estas son las opciones que debe elegir cuando se le solicite para configurar correctamente su instalación:

Para la selección del servidor, elija apache2

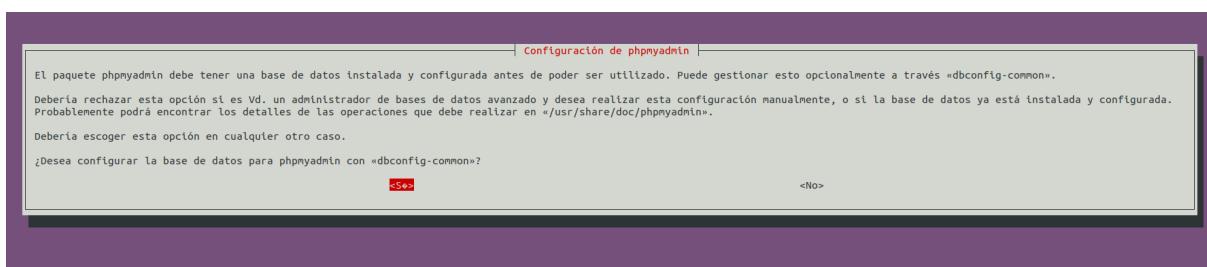
(Advertencia: Cuando la línea de comandos aparece, “apache2” está resaltado, pero no está seleccionado. Si no pulsa SPACE para seleccionar Apache, el instalador no moverá los archivos necesarios durante la instalación. Pulse ESPACIO, TAB y luego ENTER para seleccionar Apache.)

Cuando se le pregunte si utiliza dbconfig-common para configurar la base de datos, seleccione “Yes”.

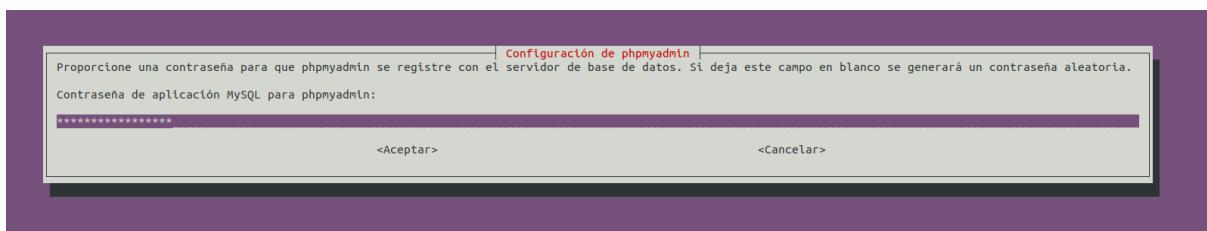
Luego, se le solicitará elegir y confirmar una contraseña para la aplicación de MySQL para phpMyAdmin.



Cuando se le pregunte si utiliza dbconfig-common para configurar la base de datos, seleccione Yes:

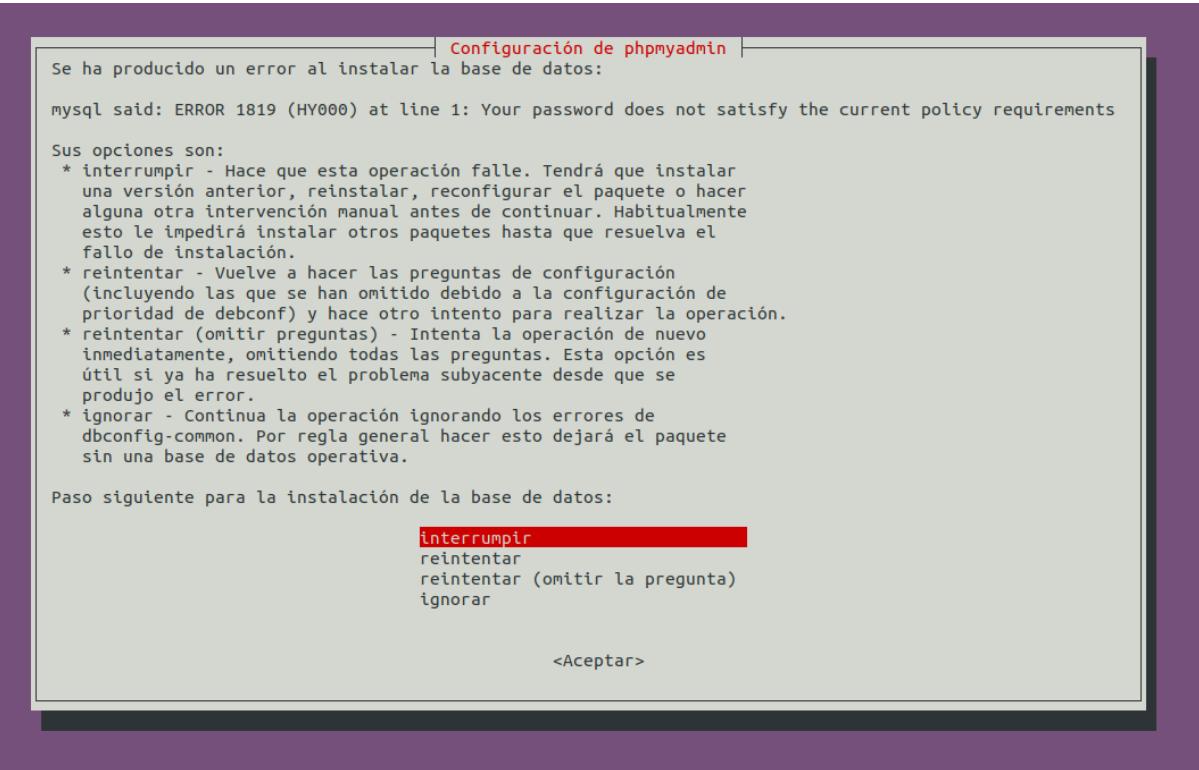


Luego, se le solicitará elegir y confirmar una contraseña para la aplicación de MySQL para phpMyAdmin:



Generará un error cuando intente establecer una contraseña para el usuario phpmyadmin:

Para resolver esto, seleccione la opción interrumpir a fin de detener el proceso de instalación. Luego, abra su línea de comandos de MySQL:



Si habilitó la autenticación de contraseña para el root user de MySQL, ejecute este comando y luego ingrese su contraseña cuando se le solicite:

sudo mysql -u root -p

Desde la línea de comandos, ejecute el siguiente comando para deshabilitar el componente Validate Password. Tenga en cuenta que con esto en realidad no se desinstalará, sino solo se evitará que el componente sea cargado en su servidor MySQL:

UNINSTALL COMPONENT "file:///component_validate_password";

Después de esto, puede cerrar el cliente MySQL:

exit;

```
ERROR 1698 (28000): Access denied for user 'root'@'localhost'
usuario@usuario-VirtualBox:/etc/nginx/sites-available$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 154
Server version: 8.0.27-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> UNINSTALL COMPONENT "file://component_validate_password";
Query OK, 0 rows affected (0,21 sec)

mysql> exit
Bye
```

Luego, vuelva a instalar el paquete phpmyadmin, que funcionará según lo previsto:

```
sudo apt install phpmyadmin
usuario@usuario-VirtualBox:/etc/nginx/sites-available$ sudo apt install phpmyadmin
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
phpmyadmin ya está en su versión más reciente (4:4.9.5+dfsg1-2).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 130 no actualizados.
1 no instalados del todo o eliminados.
Se utilizarán 0 B de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
Configurando phpmyadmin (4:4.9.5+dfsg1-2) ...
Determining localhost credentials from /etc/mysql/debian.cnf: succeeded.
dbconfig-common: writing config to /etc/dbconfig-common/phpmyadmin.conf
checking privileges on database phpmyadmin for phpmyadmin@localhost: user creation needed.
granting access to database phpmyadmin for phpmyadmin@localhost: success.
verifying access for phpmyadmin@localhost: success.
creating database phpmyadmin: success.
verifying database phpmyadmin exists: success.
populating database via sql... done.
dbconfig-common: flushing administrative password
```

Una vez que phpMyAdmin esté instalado, puede abrir la línea de comandos de MySQL una vez más con sudo mysql o mysql -u root -p y luego ejecutar el siguiente comando para volver a habilitar el componente “Validate Password”:

```
INSTALL COMPONENT "file://component_validate_password";
```

```
root@root: /etc/nginx/sites-available$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 166
Server version: 8.0.27-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> INSTALL COMPONENT "file:///component_validate_password";
Query OK, 0 rows affected (0,03 sec)

mysql> exit
Bye
```

El proceso de instalación añade el archivo de configuración de phpMyAdmin de Apache al directorio /etc/apache2/conhable/, donde se lee de forma automática. Para terminar de configurar Apache y PHP a fin de que funcionen con phpMyAdmin, la única tarea que queda a continuación en esta sección del tutorial es habilitar explícitamente la extensión PHP mbstring. Esto se puede hacer escribiendo lo siguiente:

sudo phpenmod mbstring

A continuación, reinicie Apache para que sus cambios surtan efecto:

```
sudo systemctl restart apache2
```

```
usuario@usuario-VirtualBox:/etc/nginx/sites-available$ sudo phpenmod mbstring  
usuario@usuario-VirtualBox:/etc/nginx/sites-available$ sudo systemctl restart apache2
```

phpMyAdmin ahora está instalado y configurado para funcionar con Apache. Sin embargo, para poder iniciar sesión y comenzar a interactuar con sus bases de datos de MySQL, deberá asegurarse de que sus usuarios de MySQL tengan los privilegios necesarios para interactuar con el programa.

Ajuste de la autenticación y los privilegios de usuario:

Cuando instaló phpMyAdmin en su servidor, automáticamente creó un usuario de base de datos llamado `phpmyadmin` que realiza ciertos procesos subyacentes para el programa. En vez de registrarse como este usuario con la contraseña administrativa que estableció durante la instalación, se le recomienda iniciar sesión como `root` user de MySQL o como usuario dedicado a administrar las bases de datos a través de la interfaz de phpMyAdmin.

En los sistemas Ubuntu con MySQL 5.7 (y versiones posteriores), el usuario root de MySQL se configura para la autenticación usando el complemento auth_socket de manera predeterminada en lugar de una contraseña. En muchos casos, esto proporciona mayor seguridad y utilidad, pero también puede generar complicaciones cuando sea necesario permitir que un programa externo (como phpMyAdmin) acceda al usuario.

Si aún no lo ha hecho, deberá cambiar el método de autenticación de auth_socket a uno que haga uso de una contraseña, para poder acceder a phpMyAdmin como su root user de MySQL. Para hacer esto, abra la consola de MySQL desde su terminal:

```
sudo mysql
```

A continuación, compruebe con el siguiente comando el método de autenticación utilizado por una de sus cuentas de usuario de MySQL:

```
SELECT user,authentication_string,plugin,host FROM mysql.user;
```

```
usuario@usuario-VirtualBox:/etc/nginx/sites-available$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 167
Server version: 8.0.27-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> SELECT user,authentication_string,plugin,host FROM mysql.user;
+-----+-----+-----+-----+
| user | authentication_string | plugin | host |
+-----+-----+-----+-----+
| centrousuario | *FC41F9856B13D1827D97042D2D57C0B489C8E5B8 | mysql_native_password | % |
| debian-sys-maint | $A$005$~'qjnaHzz8r"x:E7Goe0d3YNhLESBRB8/muxQtvQ.aDk0mzeb1Bmxj1a01 | caching_sha2_password | localhost |
| mysql.infoschema | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBREUSED | caching_sha2_password | localhost |
| mysql.session | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBREUSED | caching_sha2_password | localhost |
| mysql.sys | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBREUSED | caching_sha2_password | localhost |
N%&J.v.sg$YtDFqNb4Lccw5b4vR6uH4CST2YfWR.x1Mkpy2B | caching_sha2_password | localhost |
| root | | auth_socket | localhost |
+-----+-----+-----+-----+
7 rows in set (0,01 sec)
```

En este ejemplo, puede ver que, en efecto, el usuario root se autentica utilizando el complemento de auth_socket. Para configurar la cuenta de root de modo que la autenticación se realice con una contraseña, ejecute el siguiente comando ALTER USER. Asegúrese de cambiar password por una contraseña segura que elija:

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH caching_sha2_password BY
'Contraseña123_';
```

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH caching_sha2_password BY 'Contraseña123_';
Query OK, 0 rows affected (0,02 sec)
```

Nota: La instrucción ALTER USER previa establece el root user de MySQL para la autenticación con el complemento caching_sha2_password. Según la documentación oficial de MySQL, caching_sha2_password es el complemento de autenticación preferido por MySQL, ya que proporciona un cifrado de contraseña más seguro que el anterior, aunque aún usado ampliamente, mysql_native_password.

Sin embargo, algunas versiones de PHP no funcionan de forma confiable con caching_sha2_password. PHP notificó que este problema se corregió a partir de PHP 7.4, pero si encuentra un error cuando intente iniciar sesión en phpMyAdmin más adelante, es conveniente que en vez de esto establezca root para autenticar con mysql_native_password.

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY  
'Contraseña123_';
```

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'Contraseña123_';  
Query OK, 0 rows affected (0,01 sec)
```

Luego, compruebe nuevamente los métodos de autenticación empleados por cada uno de sus usuarios para confirmar que root ya no se autentique usando el complemento auth_socket:

```
SELECT user,authentication_string,plugin,host FROM mysql.user;
```

```
mysql> SELECT user,authentication_string,plugin,host FROM mysql.user;  
+-----+-----+-----+-----+  
| user | authentication_string | plugin | host |  
+-----+-----+-----+-----+  
| centrouserio | *FC41F9856B13D1827D97042D2D57C0B489CBE5B8 | mysql_native_password | % |  
| debian-sys-maint | $A$005$~qjnaHzz8r"x:E7G0e0d3YNhLESRB8/muxQtvQ.aDk0mzeb1Bmxj1a01 | caching_sha2_password | localhost |  
| mysql.infoschema | $A$005$THISISACOMBINATIONOFPASSWORDTHATMUSTNEVERBREUSED | caching_sha2_password | localhost |  
| mysql.session | $A$005$THISISACOMBINATIONOFPASSWORDTHATMUSTNEVERBREUSED | caching_sha2_password | localhost |  
| mysql.sys | $A$005$THISISACOMBINATIONOFPASSWORDTHATMUSTNEVERBREUSED | caching_sha2_password | localhost |  
| root | *517F9C87B9BB19F84152A0429AE4D1D1388F6F98 | mysql_native_password | localhost |  
+-----+-----+-----+-----+  
7 rows in set (0,00 sec)
```

En este resultado, podrá ver que el root user se autenticará usando una contraseña. Ahora podrá iniciar sesión en la interfaz phpMyAdmin como usuario root con la contraseña que estableció aquí.

Configuración del acceso con contraseña para un usuario dedicado de MySQL

Por otra parte, para el flujo de trabajo de algunos puede resultar más conveniente la conexión a phpMyAdmin con un usuario dedicado. Para hacer esto, abra una vez más el shell de MySQL:

```
sudo mysql
```

Si tiene habilitada la autenticación de contraseña para su root user, como se describe en la sección anterior, deberá ejecutar el siguiente comando e ingresar su contraseña cuando se le solicite para establecer conexión:

```
sudo mysql -u root -p
```

A partir de ahí, cree un usuario nuevo y asigne una contraseña segura:

```
CREATE USER 'andres'@'localhost' IDENTIFIED WITH caching_sha2_password BY  
'Andres123_';
```

```
mysql> CREATE USER 'andres'@'localhost' IDENTIFIED WITH caching_sha2_password BY 'Andres123_';  
Query OK, 0 rows affected (0,01 sec)
```

Nota: De nuevo, dependiendo de la versión de PHP que instaló, es conveniente establecer su nuevo usuario para la autenticación con mysql_native_password en lugar de caching_sha2_password:

```
ALTER USER 'andres'@'localhost' IDENTIFIED WITH mysql_native_password BY  
'Andres123_';
```

```
mysql> ALTER USER 'andres'@'localhost' IDENTIFIED WITH mysql_native_password BY 'Andres123_';  
Query OK, 0 rows affected (0,01 sec)
```

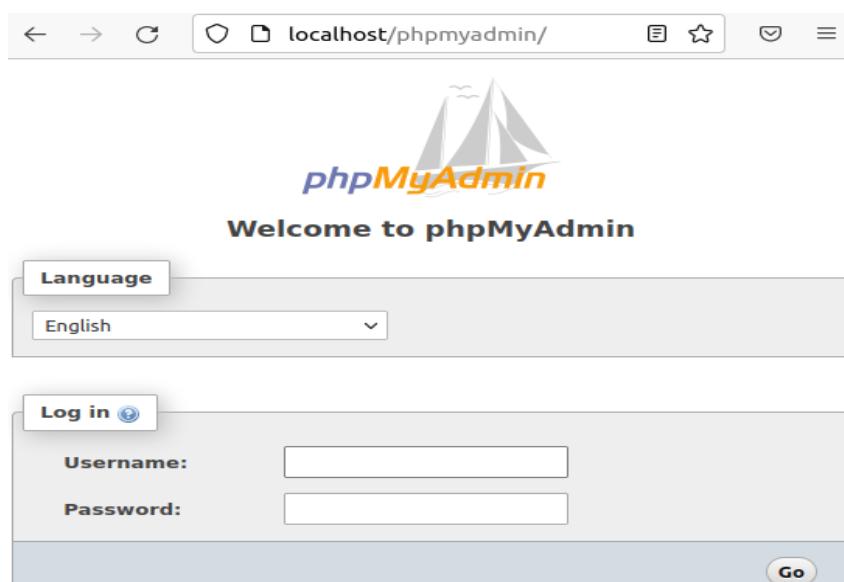
Luego, conceda a su nuevo usuario los privilegios apropiados. Por ejemplo, con el siguiente comando podría conceder privilegios de usuario a todas las tablas dentro de la base de datos, así como la facultad de añadir, cambiar y eliminar privilegios de usuario:

```
GRANT ALL PRIVILEGES ON *.* TO 'andres'@'localhost' WITH GRANT OPTION;
```

y cerramos mysql con un exit.

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'andres'@'localhost' WITH GRANT OPTION;  
Query OK, 0 rows affected (0,02 sec)  
  
mysql> exit  
Bye
```

Ahora puede acceder a la interfaz web visitando <http://localhost/phpmyadmin>



Inicie sesión en la interfaz como root o con el nombre de usuario y la contraseña que configuró.

Cuando inicie sesión, verá la interfaz de usuario. Tendrá el siguiente aspecto:

The screenshot shows the phpMyAdmin interface with the following details:

- General Configuration (Configuración general):**
 - Language: Cambio de contraseña
 - Character set for connection: utf8mb4_unicode_ci
- Appearance Configuration (Configuraciones de apariencia):**
 - Language: Español - Spanish
 - Theme: pmahomme
 - Font size: 82%
- Server Information (Servidor de base de datos):**
 - Server: localhost via UNIX socket
 - Type of server: MySQL
 - Connection to server: Not using SSL
 - Server version: 8.0.27-dbdubuntu.20.04.1 - (Ubuntu)
 - Protocol version: 10
 - User: root@localhost
 - Character set of the server: UTF-8 Unicode (utf8mb4)
- Web Server (Servidor web):**
 - Apache/2.4.41 (Ubuntu)
 - Version of the database client: libmysql - mysqld 7.4.3
 - PHP extension: mysqli curl mbstring
 - PHP version: 7.4.3
- phpMyAdmin Version (phpMyAdmin):**
 - About this version: 4.9.5deb2
 - Documentation
 - Official phpMyAdmin page
 - Contribute
 - Get support
 - List of changes
 - Licence

Ahora que puede establecer conexión e interactuar con phpMyAdmin, solo falta fortalecer la seguridad de sus sistemas para protegerlos de atacantes.

Protección de la instancia de phpMyAdmin

Debido a su presencia universal, phpMyAdmin es un objetivo popular para atacantes. Debe tomar medidas adicionales para evitar el acceso no autorizado. Una opción para hacer esto es disponer una puerta de enlace delante de toda la aplicación utilizando las funciones de autenticación y autorización de .htaccess integradas de Apache.

Para hacerlo, primero debe habilitar el uso de anulaciones del archivo .htaccess editando su archivo de configuración de Apache.

Utilice su editor de texto preferido para editar el archivo phpmyadmin.conf que se dispuso en su directorio de configuración de Apache. En este caso, utilizaremos nano:

```
sudo nano /etc/apache2/conf-available/phpmyadmin.conf
```

Agregue una directiva `AllowOverride All` dentro de la sección `<Directory /usr/share/phpmyadmin>` del archivo de configuración, como se muestra:

```
<Directory /usr/share/phpmyadmin>
    Options FollowSymLinks
    DirectoryIndex index.php
    AllowOverride All
```

```
GNU nano 4.8          /etc/apache2/conf-available/phpmyadmin.conf
# phpMyAdmin default Apache configuration

Alias /phpmyadmin /usr/share/phpmyadmin

<Directory /usr/share/phpmyadmin>
    Options FollowSymLinks
    Options SymLinksIfOwnerMatch
    DirectoryIndex index.php
    AllowOverride All

    # limit libapache2-mod-php to files and directories necessary by pr
    <IfModule mod_php7.c>
        php_admin_value upload_tmp_dir /var/lib/phpmyadmin/tmp
        php_admin_value open_basedir /usr/share/phpmyadmin/:/etc/phpmy
    </IfModule>
```

Una vez que agregue esta línea, guarde y cierre el archivo. Si utilizó nano para editar el archivo, hágalo presionando CTRL + X, Y y luego ENTER.

Para implementar los cambios que realizó, reinicie Apache:

```
sudo systemctl restart apache2
```

Ahora que habilitó el uso de .htaccess para su aplicación, deberá crear uno para implementar seguridad.

Para que pueda hacerlo de forma correcta, el archivo debe crearse dentro del directorio de la aplicación. Puede crear el archivo necesario y abrirlo en su editor de texto con privilegios root escribiendo lo siguiente:

```
sudo nano /usr/share/phpmyadmin/.htaccess
```

Cuando termine, guarde y cierre el archivo.

```
J+|                               usuario@usuario-VirtualBox: /etc/nginx/sites-available
GNU nano 4.8          /usr/share/phpmyadmin/.htaccess
AuthType Basic
AuthName "Restricted Files"
AuthUserFile /etc/phpmyadmin/.htpasswd
Require valid-user
```

La ubicación que seleccionó para su archivo de contraseña fue /etc/phpmyadmin/.htpasswd. Ahora puede crear este archivo y pasarle un usuario inicial con la utilidad htpasswd:

```
sudo htpasswd -c /etc/phpmyadmin/.htpasswd andres
```

```
usuario@usuario-VirtualBox:/etc/nginx/sites-available$ sudo htpasswd -c /etc/phpmyadmin/.htpasswd andres
New password:
Re-type new password:
Adding password for user andres
```

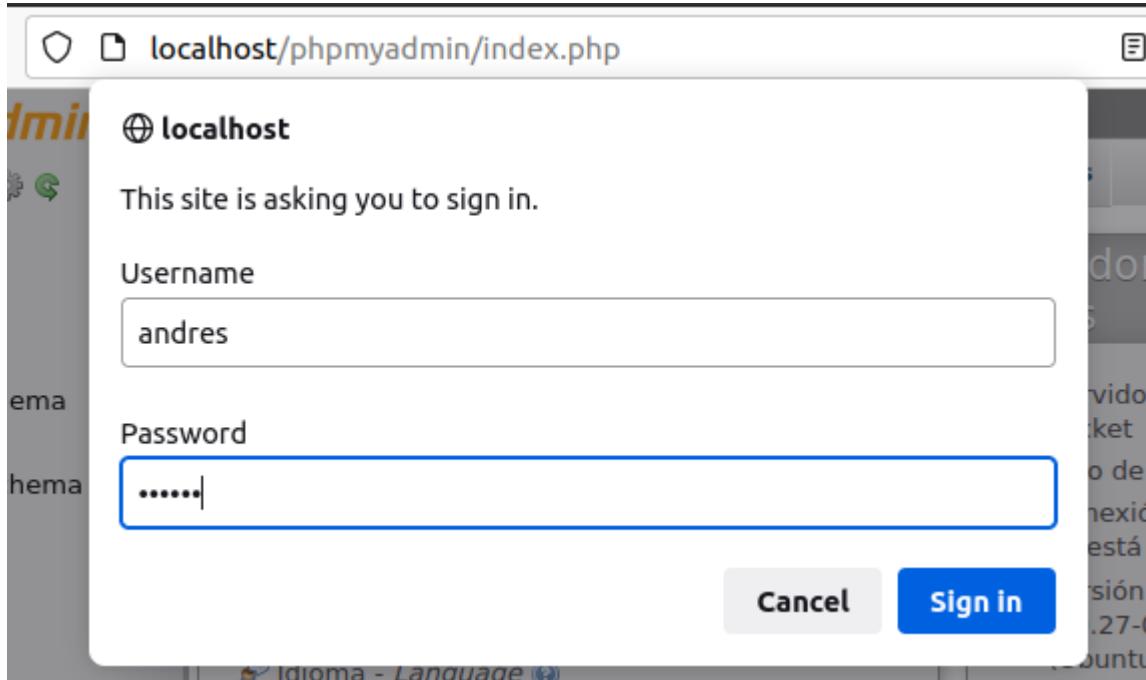
Para el usuario que creará, se le solicitará seleccionar y confirmar una contraseña. Después, el archivo se creará con la contraseña con hash que ingresó.

Si desea ingresar un usuario adicional, debe hacerlo sin el indicador -c, como se muestra:

```
sudo htpasswd /etc/phpmyadmin/.htpasswd additionaluser
```

```
Adding password for user usuario
usuario@usuario-VirtualBox:/etc/nginx/sites-available$ sudo htpasswd /etc/phpmyadmin/.htpasswd usuario
New password:
Re-type new password:
Adding password for user usuario
usuario@usuario-VirtualBox:/etc/nginx/sites-available$
```

Ahora, cuando acceda a su subdirectorio phpMyAdmin, se le solicitarán el nombre de cuenta y la contraseña adicionales que acaba de configurar:



Después de ingresar la autenticación de Apache, accederá a la página de autenticación normal de phpMyAdmin para ingresar sus credenciales de MySQL. Al añadir un conjunto adicional de credenciales que no son las de MySQL, proporciona a su base de datos una

capa de seguridad adicional. Esto es conveniente, ya que phpMyAdmin ha sido vulnerable a amenazas de seguridad en el pasado.

De esta manera, phpMyAdmin debería estar ya configurado y listo para usar en el servidor Ubuntu 20.04. Utilizando esta interfaz, puede crear fácilmente bases de datos, usuarios y tablas, y realizar operaciones habituales, como las de eliminar y modificar estructuras y datos.