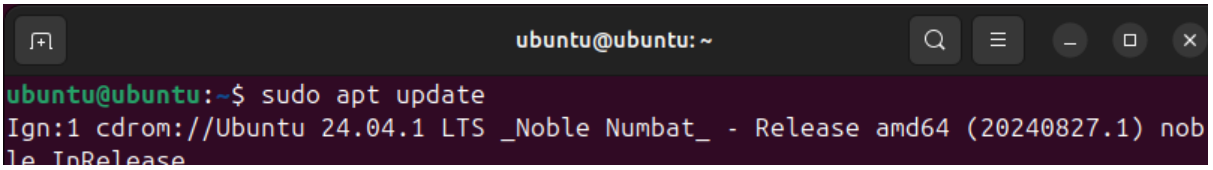
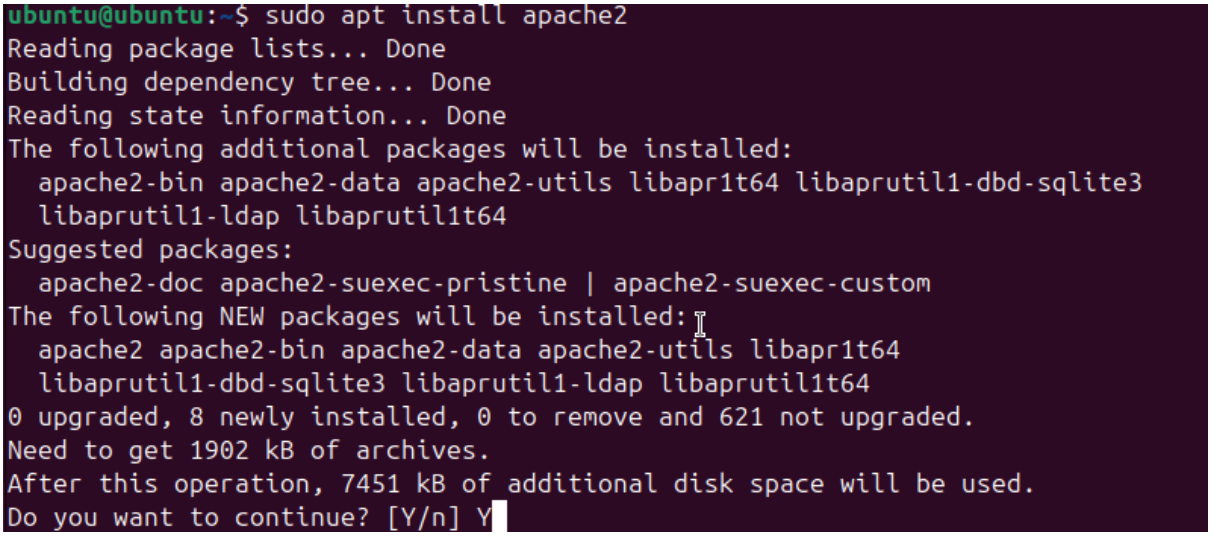
Instalación del servidor web Apache

Instalamos Apache de esta manera:

```
sudo apt update
sudo apt install apache2
```

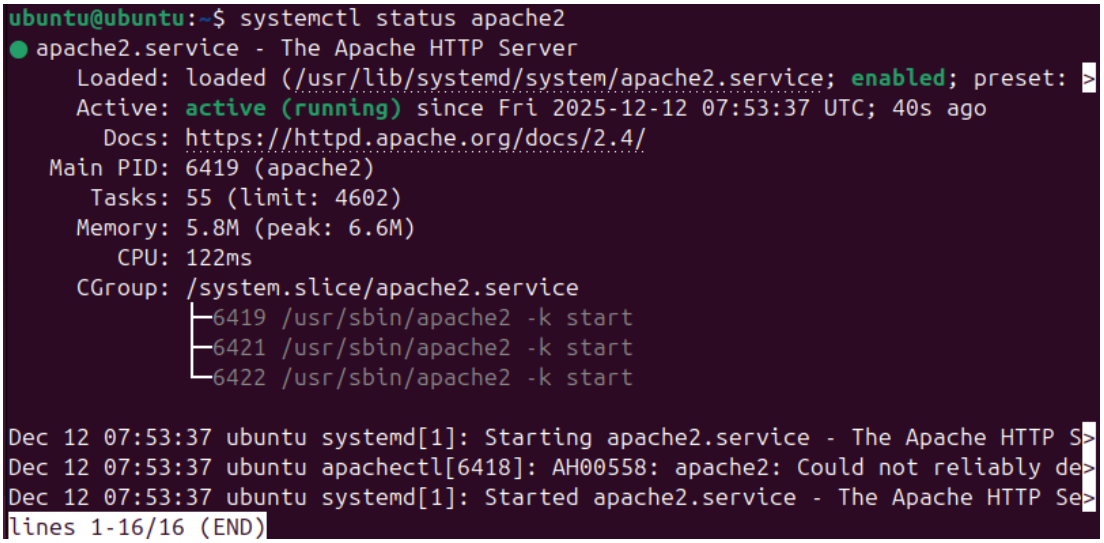




Y en donde pone Do you want to continue? pondremos Y de yes para que lo haga

Comprobamos estado

```
systemctl status apache2
```



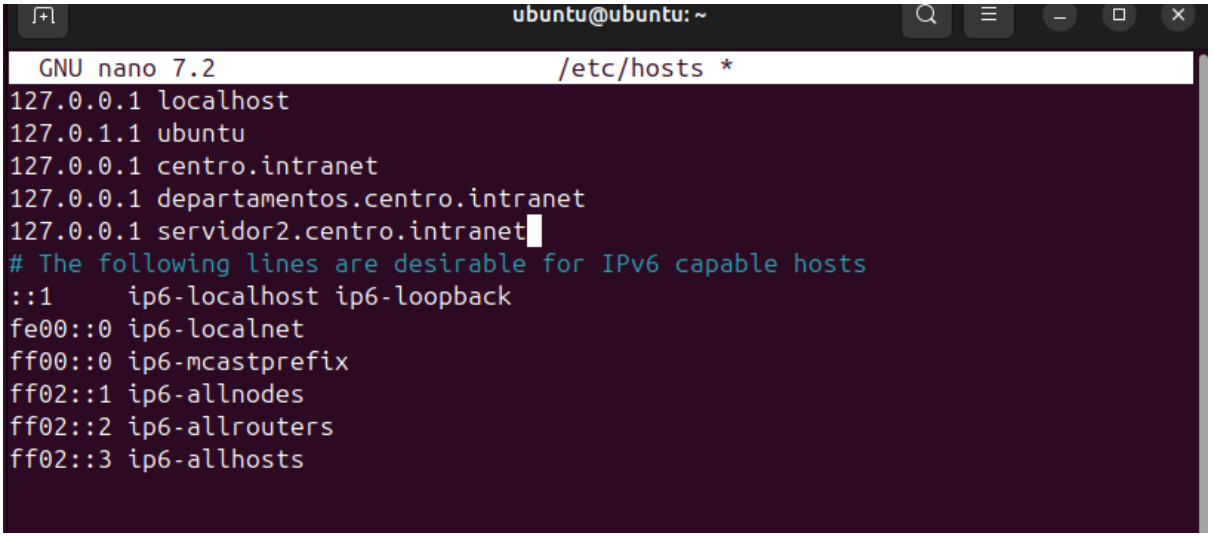
Configuración de dominios locales

Debemos modificar el archivo `/etc/hosts` con:

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

Añadimos:

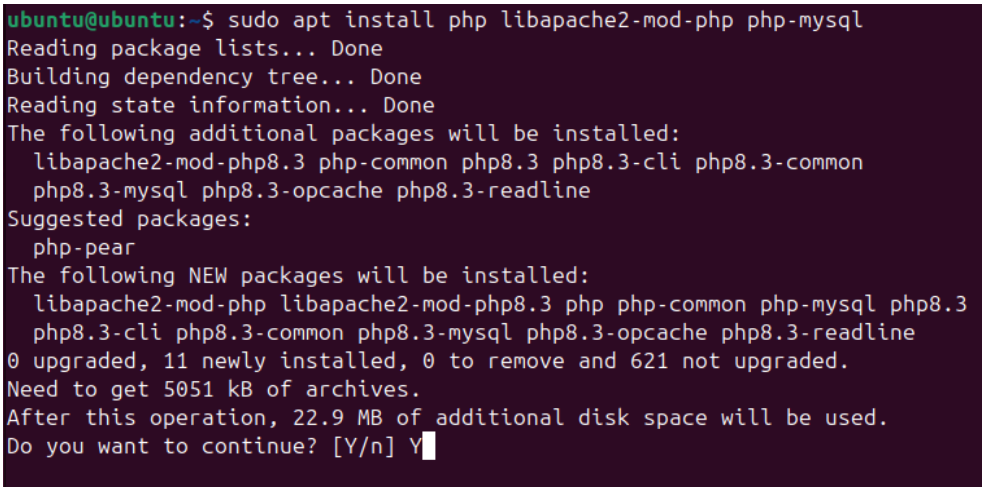
```
127.0.0.1   centro.intranet
127.0.0.1   departamentos.centro.intranet
127.0.0.1   servidor2.centro.intranet
```



Activamos módulos necesarios: PHP y MySQL

Instalamos PHP + módulos

`sudo apt install php libapache2-mod-php php-mysql`, y en donde pone Do you want to continue? pondremos Y de yes para que lo haga



Y reiniciamos Apache de esta manera:

```
sudo systemctl restart apache2
```

Instalar MariaDB/MySQL

```
sudo apt install mariadb-server
```

En donde pone Do you want to continue? pondremos de nuevo Y de yes para que lo haga

```
ubuntu@ubuntu:~$ sudo apt install mariadb-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  galera-4 gawk libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl
  libdbd-mysql-perl libdbi-perl libfcgi-bin libfcgi-perl libfcgi0t64
  libhtml-template-perl libmariadb3 libmysqlclient21 libsigsegv2 libsnappy1v5
  libterm-readkey-perl liburing2 mariadb-client mariadb-client-core
Need to get 19.4 MB/19.8 MB of archives.
After this operation, 202 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

```
sudo mysql_secure_installation
```

```
ubuntu@ubuntu:~$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] Y
```

Y en esta última línea pondremos Y, y a continuación también en las demás preguntas

Instalar y configurar WordPress

Creamos VirtualHost para centro.intranet:

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

A continuación añadimos este contenido:

```
<VirtualHost *:80>
    ServerName centro.intranet
    DocumentRoot /var/www/centro.intranet

    <Directory /var/www/centro.intranet>
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/centro-error.log
    CustomLog ${APACHE_LOG_DIR}/centro-access.log combined
</VirtualHost>
```

Y activamos el sitio con:

```
sudo systemctl reload apache2
sudo a2ensite centro.intranet.conf
```

```
ubuntu@ubuntu:~$ sudo systemctl reload apache2
ubuntu@ubuntu:~$ sudo a2ensite centro.intranet.conf
Site centro.intranet already enabled
```

Para instalar WordPress:

```
cd /var/www
sudo wget https://wordpress.org/latest.tar.gz
sudo tar -xvzf latest.tar.gz
sudo mv wordpress centro.intranet
sudo chown -R www-data:www-data centro.intranet
```

```
ubuntu@ubuntu:~$ cd /var/www
ubuntu@ubuntu:/var/www$ sudo wget https://wordpress.org/latest.tar.gz
--2025-12-12 08:27:44--  https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252, 2607:f978:5:8002::c6
8f:a4fc
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 27058021 (26M) [application/octet-stream]
Saving to: ‘latest.tar.gz’

latest.tar.gz       100%[===================>]  25.80M  15.6MB/s    in 1.7s

2025-12-12 08:27:47 (15.6 MB/s) - ‘latest.tar.gz’ saved [27058021/27058021]

ubuntu@ubuntu:/var/www$ sudo tar -xvf latest.tar.gz
wordpress/
wordpress/index.php
wordpress/license.txt
wordpress/readme.html
wordpress/wp-activate.php
```

Crear base de datos

Entramos en:

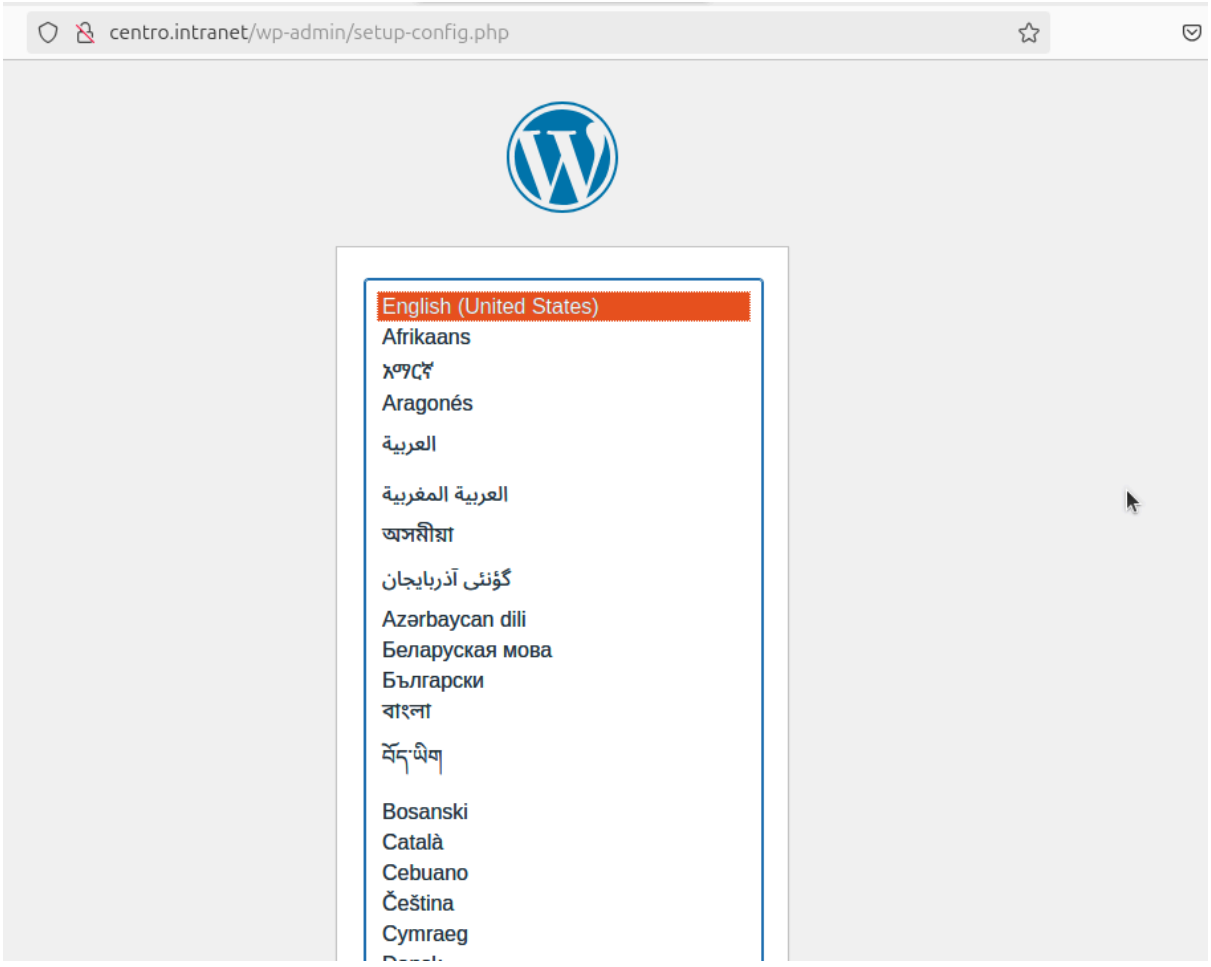
```
sudo mysql -u root -p
```

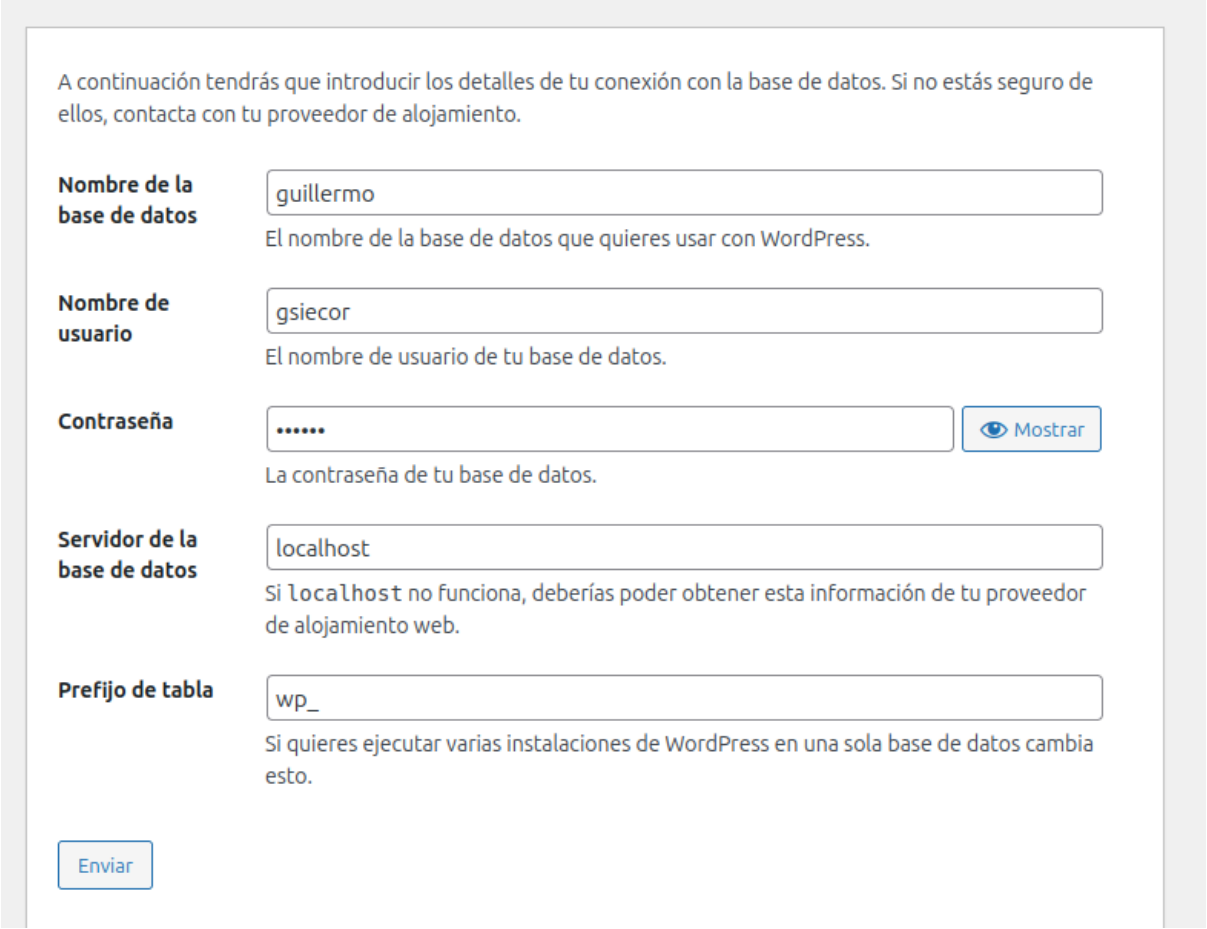
Y ponemos:

```
CREATE DATABASE wpdb;
CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'clave';
GRANT ALL PRIVILEGES ON wpdb.* TO 'wpuser'@'localhost';
FLUSH PRIVILEGES;
```

```
ubuntu@ubuntu:~$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 39
Server version: 10.11.13-MariaDB-0ubuntu0.24.04.1 Ubuntu 24.04

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

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

MariaDB [(none)]> CREATE DATABASE wpdb;
Query OK, 1 row affected (0.003 sec)

MariaDB [(none)]> CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'clave';
Query OK, 0 rows affected (0.015 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON wpdb.* TO 'wpuser'@'localhost';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]>
```

Después entramos a `http://centro.intranet` y completamos la instalación.





Activar módulo WSGI para aplicaciones Python

Para instalar mod_wsgi:

```
sudo apt install libapache2-mod-wsgi-py3
sudo a2enmod wsgi
```

```
ubuntu@ubuntu:~$ sudo apt install libapache2-mod-wsgi-py3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  libapache2-mod-wsgi-py3
0 upgraded, 1 newly installed, 0 to remove and 621 not upgraded.
Need to get 103 kB of archives.
After this operation, 300 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble/main amd64 libapache2-mod-wsgi-py3
amd64 5.0.0-1build2 [103 kB]
Fetched 103 kB in 0s (272 kB/s)
Selecting previously unselected package libapache2-mod-wsgi-py3.
(Reading database ... 212620 files and directories currently installed.)
Preparing to unpack .../libapache2-mod-wsgi-py3_5.0.0-1build2_amd64.deb ...
Unpacking libapache2-mod-wsgi-py3 (5.0.0-1build2) ...
Setting up libapache2-mod-wsgi-py3 (5.0.0-1build2) ...
apache2_invoke: Enable module wsgi
ubuntu@ubuntu:~$ sudo a2enmod wsgi
Module wsgi already enabled
```

Creamos la aplicación Python

Creamos la carpeta de la aplicación Python:

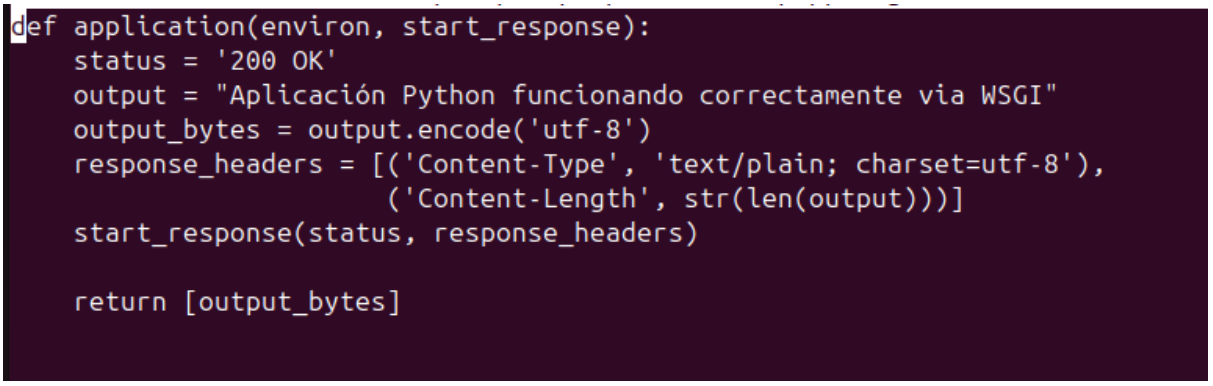
```
sudo mkdir -p /var/www/departamentos
```

Creamos el archivo WSGI dentro de esa carpeta:

sudo nano /var/www/departamentos/app.wsgi

Y ponemos esto:

```
def application(environ, start_response):
    status = '200 OK'
    output = "Aplicación Python funcionando correctamente via WSGI"
    output_bytes = output.encode('utf-8')
    response_headers = [
        ('Content-Type', 'text/plain; charset=utf-8'),
        ('Content-Length', str(len(output_bytes)))
    ]
    start_response(status, response_headers)
    return [output_bytes]
```



Creamos el VirtualHost de departamentos.centro.intranet

sudo nano /etc/apache2/sites-available/departamentos.conf

Y ponemos este contenido:

```
<VirtualHost *:80>
```

```
    ServerName departamentos.centro.intranet

    WSGIScriptAlias / /var/www/departamentos/app.wsgi

    <Directory /var/www/departamentos>
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/departamentos-error.log
    CustomLog ${APACHE_LOG_DIR}/departamentos-access.log combined
</VirtualHost>
```

```
<VirtualHost *:80>
    ServerName departamentos.centro.intranet

    WSGIScriptAlias / /var/www/departamentos/app.wsgi

    <Directory /var/www/departamentos>
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/departamentos-error.log
    CustomLog ${APACHE_LOG_DIR}/departamentos-access.log combined
</VirtualHost>
```

Activamos:

```
sudo a2ensite departamentos.centro.intranet.conf
sudo systemctl reload apache2
```

Y entramos a http://departamentos.centro.intranet

```
ubuntu@ubuntu:~$ sudo systemctl reload apache2
ubuntu@ubuntu:~$ sudo a2ensite departamentos.conf
Site departamentos already enabled
```

Aplicación Python funcionando correctamente via WSG

Proteger la aplicación Python con autenticación

Crear archivo de usuarios

```
sudo apt install apache2-utils
sudo htpasswd -c /etc/apache2/.htpasswd admin
```

```
ubuntu@ubuntu:~$ sudo apt install apache2-utils
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2-utils is already the newest version (2.4.58-1ubuntu8.8).
apache2-utils set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 621 not upgraded.
ubuntu@ubuntu:~$ sudo htpasswd -c /etc/apache2/.htpasswd admin
New password:
Re-type new password:
Adding password for user admin
```

Añadir autenticación al VirtualHost

Dentro del <Directory> ponemos esto para reemplazar lo anterior:

```
AuthType Basic
AuthName "Zona restringida"
AuthUserFile /etc/apache2/.htpasswd
Require valid-user
```

```
<VirtualHost *:80>
    ServerName departamentos.centro.intranet

    WSGIScriptAlias / /var/www/departamentos/app.wsgi

    <Directory /var/www/departamentos>
        AuthType Basic
        AuthName "Zona restringida"
        AuthUserFile /etc/apache2/.htpasswd
        Require valid-user
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/departamentos-error.log
    CustomLog ${APACHE_LOG_DIR}/departamentos-access.log combined
</VirtualHost>
```

Recargamos así:

```
sudo systemctl reload apache2
```

Instalar y configurar AWStats

Instalamos:

`sudo apt install awstats`

```
ubuntu@ubuntu:~$ sudo apt install awstats
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libnet-xwhois-perl
Suggested packages:
  libnet-dns-perl libnet-ip-perl libgeo-ipfree-perl
The following NEW packages will be installed:
  awstats libnet-xwhois-perl
0 upgraded, 2 newly installed, 0 to remove and 621 not upgraded.
Need to get 1889 kB of archives.
After this operation, 7154 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Y en Do you want to continue?, como siempre, volvemos a poner Y

Configuramos el dominio principal:

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

Y modificamos:

```
SiteDomain="centro.intranet"
LogFile="/var/log/apache2/access.log"
```

```
  GNU nano 7.2    /etc/awstats/awstats.centro.intranet.conf
# If analyzing mail log, enter here the domain name of mail server.
# Example: "myintranetserver"
# Example: "www.domain.com"
# Example: "ftp.domain.com"
# Example: "domain.com"
#
SiteDomain="centro.intranet"
LogFile="/var/log/apache2/access.log"


# Enter here all other possible domain names, addresses or virtual host
# aliases someone can use to access your site. Try to keep only the min
# number of possible names/addresses to have the best performances.
# You can repeat the "SiteDomain" value in this list.
# This parameter is used to analyze referer field in log file and to he
```

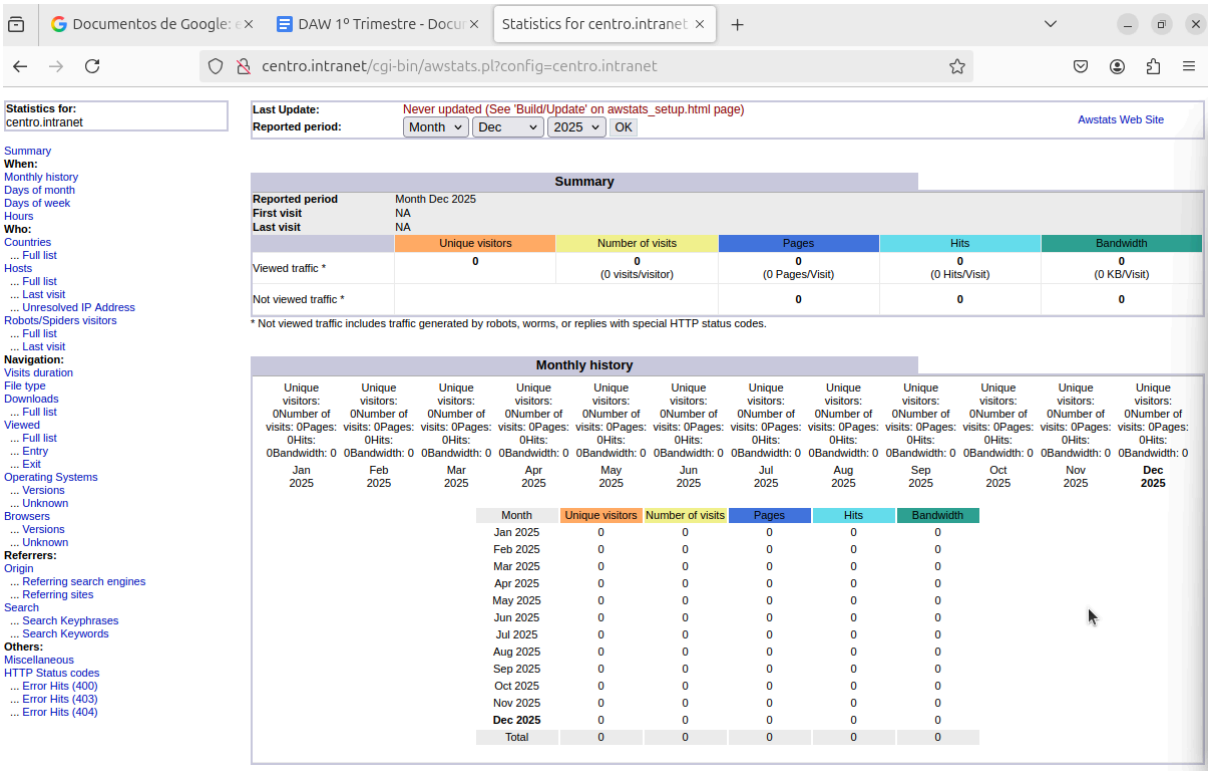
Activamos actualización:

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

```
ubuntu@ubuntu:~$ sudo /usr/lib/cgi-bin/awstats.pl -config=centro.intrane
t -update
Create/Update database for config "/etc/awstats/awstats.centro.intranet.
conf" by AWStats version 7.9 (build 20230108)
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...
Jumped lines in file: 0
Parsed lines in file: 0
 Found 0 dropped records,
 Found 0 comments,
 Found 0 blank records,
 Found 0 corrupted records,
 Found 0 old records,
 Found 0 new qualified records.
```

Y ponemos el acceso para comprobar:

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

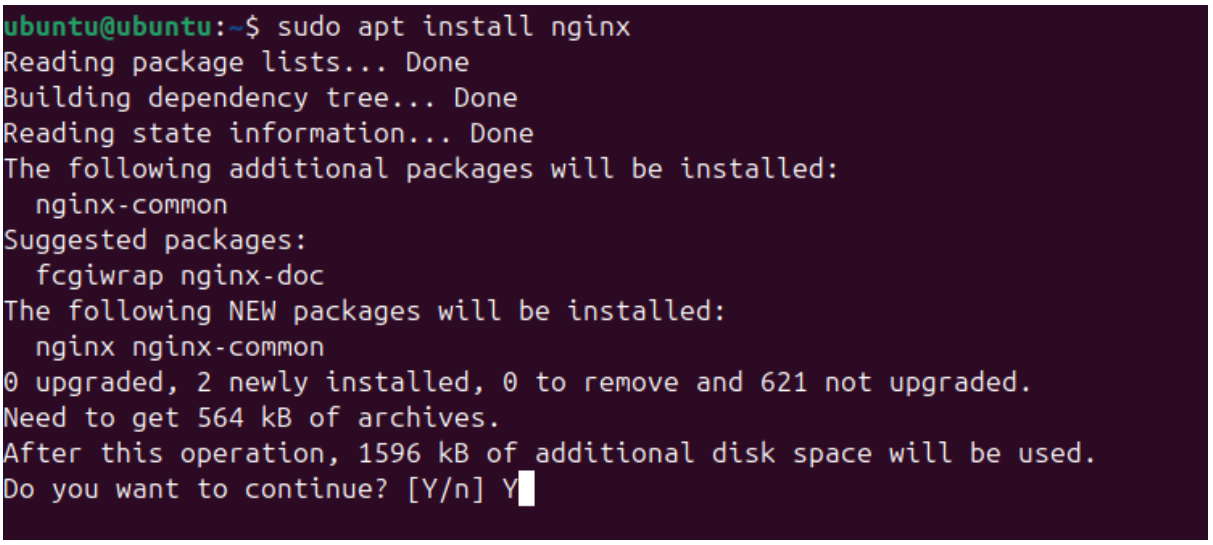


Instalar un segundo servidor web

En mi caso instalaré nginx.

Instalar nginx

```
sudo apt install nginx
```



Configuramos para puerto 8080

Abrimos con:

```
sudo nano /etc/nginx/sites-available/servidor2
```

Y ponemos este contenido:

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

    index index.php index.html;

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php8.2-fpm.sock;
    }
}
```

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

    index index.php index.html;

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php8.2-fpm.sock;
    }
}
```

Y activamos con:

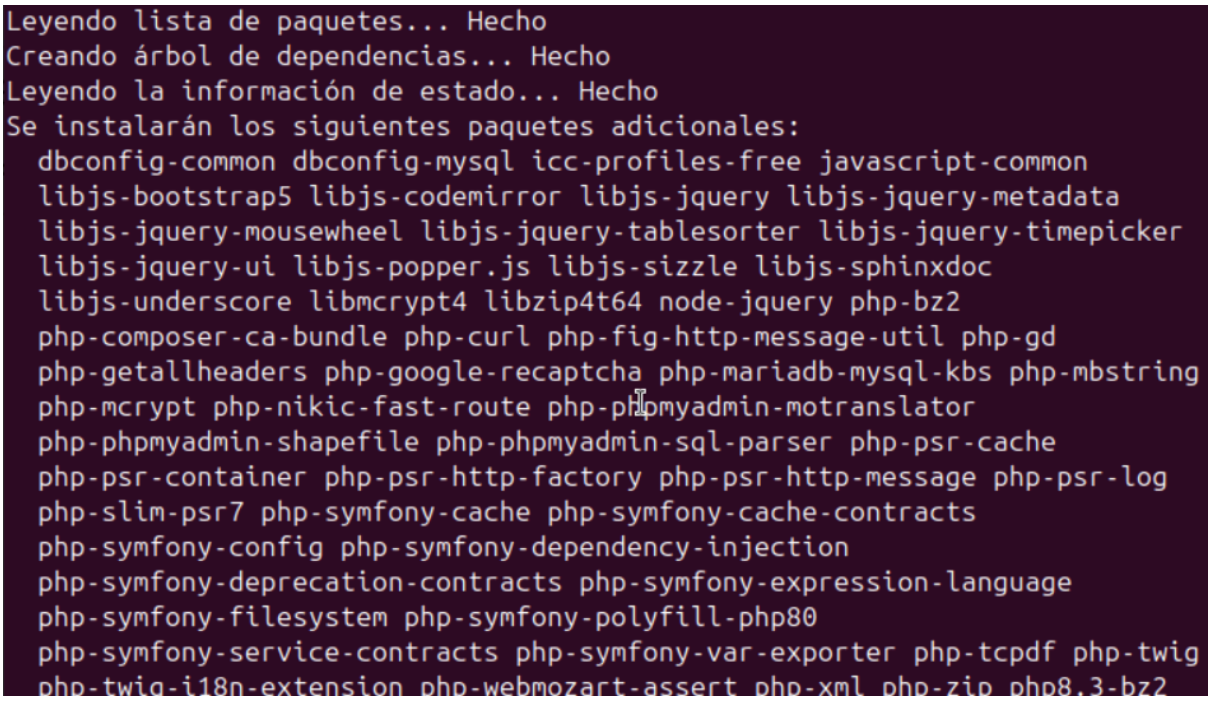
```
sudo ln -s /etc/nginx/sites-available/servidor2
/etc/nginx/sites-enabled/
sudo systemctl restart nginx
```

Instalar PHP para nginx (mediante PHP-FPM)

```
sudo apt install php-fpm php-mysql
sudo systemctl restart php8.2-fpm
```

Instalar phpMyAdmin en el servidor 2

```
sudo apt install phpmyadmin
```



Crear enlace dentro del root del servidor 2:

```
sudo ln -s /usr/share/phpmyadmin /var/www/servidor2/phpmyadmin
```

Entrar:

```
http://servidor2.centro.intranet:8080/phpmyadmin
```

servidor2.centro.intranet:8080

PHP Version 8.3.6

System	Linux ubuntu 6.8.0-41-generic #41-Ubuntu SMP PREEMPT_DYNAMIC Fri Aug 2 20:41:06 UTC 2024 x86_64
Build Date	Jul 14 2025 18:30:55
Build System	Linux
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.3/fpm
Loaded Configuration File	/etc/php/8.3/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/8.3/fpm/conf.d
Additional .ini files parsed	/etc/php/8.3/fpm/conf.d/10-mysqlnd.ini, /etc/php/8.3/fpm/conf.d/10-opcache.ini, /etc/php/8.3/fpm/conf.d/10-pdo.ini, /etc/php/8.3/fpm/conf.d/15-xml.ini, /etc/php/8.3/fpm/conf.d/20-bz2.ini, /etc/php/8.3/fpm/conf.d/20-calendar.ini, /etc/php/8.3/fpm/conf.d/20-ctype.ini, /etc/php/8.3/fpm/conf.d/20-curl.ini, /etc/php/8.3/fpm/conf.d/20-dom.ini, /etc/php/8.3/fpm/conf.d/20-exif.ini, /etc/php/8.3/fpm/conf.d/20-ffi.ini, /etc/php/8.3/fpm/conf.d/20-fileinfo.ini, /etc/php/8.3/fpm/conf.d/20-ftp.ini, /etc/php/8.3/fpm/conf.d/20-gd.ini, /etc/php/8.3/fpm/conf.d/20-gettext.ini, /etc/php/8.3/fpm/conf.d/20-iconv.ini, /etc/php/8.3/fpm/conf.d/20-mbstring.ini, /etc/php/8.3/fpm/conf.d/20-mcrypt.ini, /etc/php/8.3/fpm/conf.d/20-mysqli.ini, /etc/php/8.3/fpm/conf.d/20-pdo_mysql.ini, /etc/php/8.3/fpm/conf.d/20-phar.ini, /etc/php/8.3/fpm/conf.d/20-posix.ini, /etc/php/8.3/fpm/conf.d/20-readline.ini, /etc/php/8.3/fpm/conf.d/20-shmop.ini, /etc/php/8.3/fpm/conf.d/20-simplexml.ini, /etc/php/8.3/fpm/conf.d/20-sockets.ini, /etc/php/8.3/fpm/conf.d/20-sysvmsg.ini, /etc/php/8.3/fpm/conf.d/20-sysvsem.ini, /etc/php/8.3/fpm/conf.d/20-sysvshm.ini, /etc/php/8.3/fpm/conf.d/20-tokenizer.ini, /etc/php/8.3/fpm/conf.d/20-xmlreader.ini, /etc/php/8.3/fpm/conf.d/20-xmlwriter.ini, /etc/php/8.3/fpm/conf.d/20-xsl.ini, /etc/php/8.3/fpm/conf.d/20-zip.ini
PHP API	20230831
PHP Extension	20230831
Zend Extension	420230831
Zend Extension Build	API420230831,NTS
PHP Extension Build	API20230831,NTS