

PRÁCTICA SERVIDORES WEB



Alumno: Jesús Carlos Mora Mesa

Módulo: Servicios de Red e Internet

2º ASIR

Índice

1.- Dominios mediante el archivo hosts:.....	3
2.- Instalación de la pila LAMP.....	4
3.- Instalación y Configuración de WordPress.....	5
4.- Aplicación Python con WSGI.....	9
5.- Proteger Python con Autenticación.....	11
6.- Instalar y Configurar AWStats.....	12
7.- NGINX y PHPMyAdmin.....	13

Partiendo desde una instalación limpia de Ubuntu 25.01, lo primero que vamos a hacer es actualizar todos los paquetes:

```
jcm-srei@jcm-srei: ~ — sudo apt update
jcm-srei@jcm-srei:~$ sudo apt update && sudo apt upgrade -y
[sudo: authenticate] Password:
Obj:1 http://security.ubuntu.com/ubuntu questing-security InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu questing InRelease
Obj:3 http://es.archive.ubuntu.com/ubuntu questing-updates InRelease
Obj:4 http://es.archive.ubuntu.com/ubuntu questing-backports InRelease
Leyendo lista de paquetes... 9%
```

Esperamos a que se complete y vamos a por el primer punto.

1.- Dominios mediante el archivo hosts:

Según el enunciado, necesitamos simular dos dominios primero (centro.intranet y departamentos.centro.intranet) y un tercero para la parte final de la práctica con nginx (servidor2.centro.intranet). Para esto nos vamos al archivo hosts y lo editamos:

```
jcm-srei@jcm-srei: ~ — sudo nano /etc/hosts
GNU nano 8.4 /etc/hosts *
127.0.0.1 localhost
127.0.1.1 jcm-srei
127.0.0.1 centro.intranet
127.0.0.1 departamentos.centro.intranet
127.0.0.1 servidor2.centro.intranet
```

Para comprobar que esto se ha ejecutado correctamente, vamos a hacer ping a esos dominios que hemos añadido:

```
jcm-srei@jcm-srei:~$ ping centro.intranet
PING centro.intranet (127.0.0.1) 56(84) bytes of data:
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.135 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.040 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.043 ms
^C
--- centro.intranet ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2027ms
rtt min/avg/max/mdev = 0.040/0.072/0.135/0.044 ms
jcm-srei@jcm-srei:~$
```

```
jcm-srei@jcm-srei:~$ ping departamentos.centro.intranet
PING departamentos.centro.intranet (127.0.0.1) 56(84) bytes of data:
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.049 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.037 ms
^C
--- departamentos.centro.intranet ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1032ms
rtt min/avg/max/mdev = 0.037/0.043/0.049/0.006 ms
jcm-srei@jcm-srei:~$
```

```
jcm-srei@jcm-srei:~$ ping servidor2.centro.intranet
PING servidor2.centro.intranet (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.029 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.041 ms
^C
--- servidor2.centro.intranet ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1045ms
rtt min/avg/max/mdev = 0.029/0.035/0.041/0.006 ms
jcm-srei@jcm-srei:~$
```

2.- Instalación de la pila LAMP

Vamos a instalar Apache, MySQL y PHP todo a la misma vez mediante el repositorio apt:

```
jcmw-srei@jcmw-srei:~$ sudo apt install apache2 mysql-server php libapache2-mod-php php-mysql php-gd php-xml php-mbstring -y
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
linux-headers-6.17.0-5 linux-headers-6.17.0-5-generic linux-modules-6.17.0-5-generic linux-tools-6.17.0-5 linux-tools-6.17.0-5-generic
Utilice «sudo apt autoremove» para eliminarlos.

Installing:
  apache2 libapache2-mod-php mysql-server php php-gd php-mbstring php-mysql php-xml

Installing dependencies:
  apache2-bin libaprutil1-ldap libfcgi0t64 mecab-ipadic php-common php8.4-opcache
  apache2-data libaprutil1t64 libgoogle-perftools4t64 mecab-ipadic-utf8 php8.4 php8.4-readline
  apache2-utils libargon2-1 libhtml-template-perl mecab-utils php8.4-cli php8.4-xml
  libaio1t64 libbcbi-fast-perl libmecab2 mysql-client php8.4-common
  libapache2-mod-php8.4 libcgi-pm-perl libprotobuf-lite32t64 mysql-client-core php8.4-gd
  libapr1t64 libfcgi-bin libsodium23 mysql-common php8.4-mbstring
  libaprutil1-dbd-sqlite3 libfcgi-perl libtcmalloc-minimal4t64 mysql-server-core php8.4-mysql
```

Una vez se ha instalado, vamos a activar el servicio de apache:

```
jcm-srei@jcm-srei:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
jcm-srei@jcm-srei:~$ sudo systemctl start apache2
jcm-srei@jcm-srei:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-11-25 23:38:22 CET; 1min 3s ago
 Invocation: 385f313a9c2b431094489d0deb5b7bc8
    Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 20084 (apache2)
    Tasks: 6 (limit: 8855)
  Memory: 13.4M (peak: 13.6M)
     CPU: 110ms
  CGroup: /system.slice/apache2.service
          └─20084 /usr/sbin/apache2 -k start
            └─20087 /usr/sbin/apache2 -k start
              └─20088 /usr/sbin/apache2 -k start
                └─20089 /usr/sbin/apache2 -k start
                  └─20090 /usr/sbin/apache2 -k start
                    └─20091 /usr/sbin/apache2 -k start

nov 25 23:38:22 jcm-srei systemd[1]: Starting apache2.service - The Apache HTTP Server...
nov 25 23:38:22 jcm-srei apachectl[20083]: AH00558: apache2: Could not reliably determine the server's full
nov 25 23:38:22 jcm-srei systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-20/20 (END)
```

Con esto ya quedaría instalada la pila LAMP, más adelante configuraremos los elementos por separado.

3.- Instalación y Configuración de WordPress

El dominio centro.intranet será el que servirá el wordpress. Lo primero que vamos a hacer es crear la base de datos, sin ella el wordpress no funciona:

```
jcm-srei@jcm-srei: ~ — sudo mysql -u root

jcm-srei@jcm-srei:~$ sudo mysql -u root
[sudo: authenticate] Password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.4.7-0ubuntu0.25.10.2 (Ubuntu)

Copyright (c) 2000, 2025, 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>
```

Dentro de sql, creamos la base de datos con estos parámetros:

```
CREATE DATABASE wordpressdb;
```

```
CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'user';
```

```
GRANT ALL PRIVILEGES ON wordpressdb.* TO 'wpuser'@'localhost';
```

```
FLUSH PRIVILEGES;
```

```
EXIT;
```

Según he visto, es la base de datos más sencilla para que Wordpress funcione.

```
mysql> CREATE DATABASE wordpressdb;
Query OK, 1 row affected (0,01 sec)

mysql> CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'user';
Query OK, 0 rows affected (0,04 sec)

mysql> GRANT ALL PRIVILEGES ON wordpressdb.* TO 'wpuser'@'localhost';
Query OK, 0 rows affected (0,00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,00 sec)

mysql>
```

Después de esto, vamos a descargar Wordpress:

```
jcm-srei@jcm-srei:~$ cd /tmp
jcm-srei@jcm-srei:/tmp$ wget https://wordpress.org/latest.tar.gz
--2025-11-26 00:04:30-- https://wordpress.org/latest.tar.gz
Resolviendo wordpress.org (wordpress.org)... 198.143.164.252, 2607:f978:5:8002::c68f:a4f
Conectando con wordpress.org (wordpress.org)[198.143.164.252]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 26928488 (26M) [application/octet-stream]
Guardando como: 'latest.tar.gz'

latest.tar.gz                               100%[=====]
2025-11-26 00:04:32 (12,5 MB/s) - 'latest.tar.gz' guardado [26928488/26928488]
```

Lo descomprimos, movemos y le cambiamos los permisos:

```
jcm-srei@jcm-srei:/tmp$ tar -xvzf latest.tar.gz
wordpress/
wordpress/index.php
```

```
jcm-srei@jcm-srei:/tmp$ sudo mv wordpress /var/www/centro
jcm-srei@jcm-srei:/tmp$ sudo chown -R www-data:www-data /var/www/centro
jcm-srei@jcm-srei:/tmp$ sudo chmod -R 755 /var/www/centro
jcm-srei@jcm-srei:/tmp$
```

Por último, vamos a configurar el VirtualHost, creando el archivo de configuración para el sitio:

```
jcm-srei@jcm-srei:/$ sudo nano /etc/apache2/sites-available/centro.conf
jcm-srei@jcm-srei:/$
```

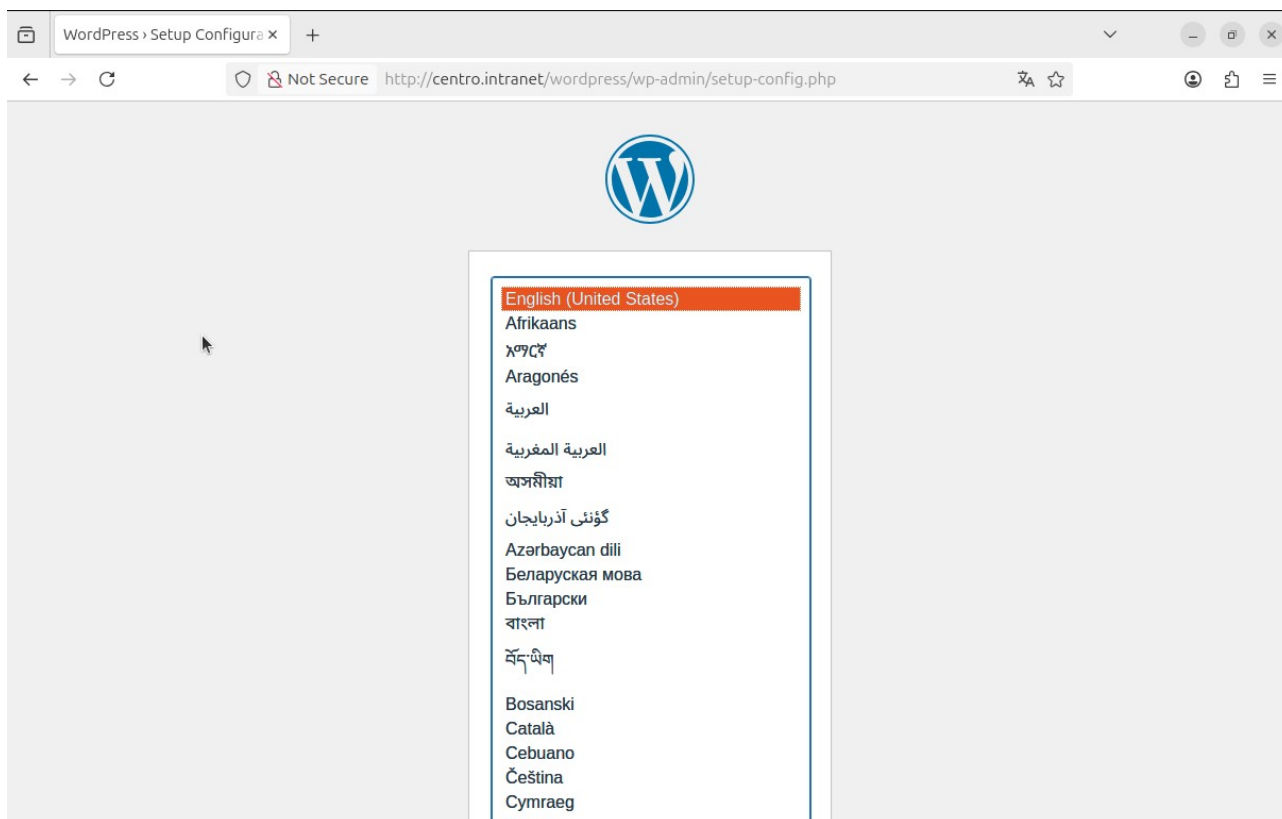
Lo creamos con esta configuración:

```
GNU nano 8.4 /etc/apache2/sites-available/centro.conf *
<VirtualHost *:80>
    ServerName centro.intranet
    DocumentRoot /var/www/centro
    <Directory /var/www/centro>
        AllowOverride All
        Require all granted
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/centro_error.log
    CustomLog ${APACHE_LOG_DIR}/centro_access.log combined
</VirtualHost>
```

Habilitamos el sitio con el modulo a2ensite:

```
jcm-srei@jcm-srei:/$ sudo a2ensite centro.conf
Enabling site centro.
To activate the new configuration, you need to run:
systemctl reload apache2
jcm-srei@jcm-srei:/$ sudo systemctl reload apache2
jcm-srei@jcm-srei:/$
```

Para comprobar que esto se ha realizado correctamente, vamos al navegador y ponemos en la barra de búsqueda “centro.intranet” y nos debería aparecer la página de configuración de wordpress:



Podemos completarla para finalizar la instalación. Ponemos los datos y en la siguiente ventana nos conectamos a la base de datos:

A continuación tendrás que introducir los detalles de tu conexión con la base de datos. Si no estás seguro de ellos, contacta con tu proveedor de alojamiento.

Nombre de la base de datos	<input type="text" value="wordpressdb"/>
	El nombre de la base de datos que quieres usar con WordPress.
Nombre de usuario	<input type="text" value="wpuser"/>
	El nombre de usuario de tu base de datos.
Contraseña	<input type="password" value="...."/> Mostrar
	La contraseña de tu base de datos.
Servidor de la base de datos	<input type="text" value="localhost"/>
	Si localhost no funciona, deberías poder obtener esta información de tu proveedor de alojamiento web.
Prefijo de tabla	<input type="text" value="wp_"/>
	Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.

[Enviar](#)

Rellenamos el nombre, nombre de usuario...

Información necesaria

Por favor, proporciona la siguiente información. No te preocupes, siempre podrás cambiar estos ajustes más tarde.

Título del sitio	<input type="text" value="Wordpress_prueba"/>
Nombre de usuario	<input type="text" value="jcmm"/>
	Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.
Contraseña	<input type="password" value="^y9H02vdRKpLf351Z0"/> Ocultar
	Fuerte
	Importante: Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.
Tu correo electrónico	<input type="text" value="jesuscmoramesa@ieslamarisma.net"/>
	Comprueba bien tu dirección de correo electrónico antes de continuar.
Visibilidad en los motores de búsqueda	<input type="checkbox"/> Pedir a los motores de búsqueda que no indexen este sitio
	Depende de los motores de búsqueda atender esta petición o no.

[Instalar WordPress](#)

¡Lo lograste!

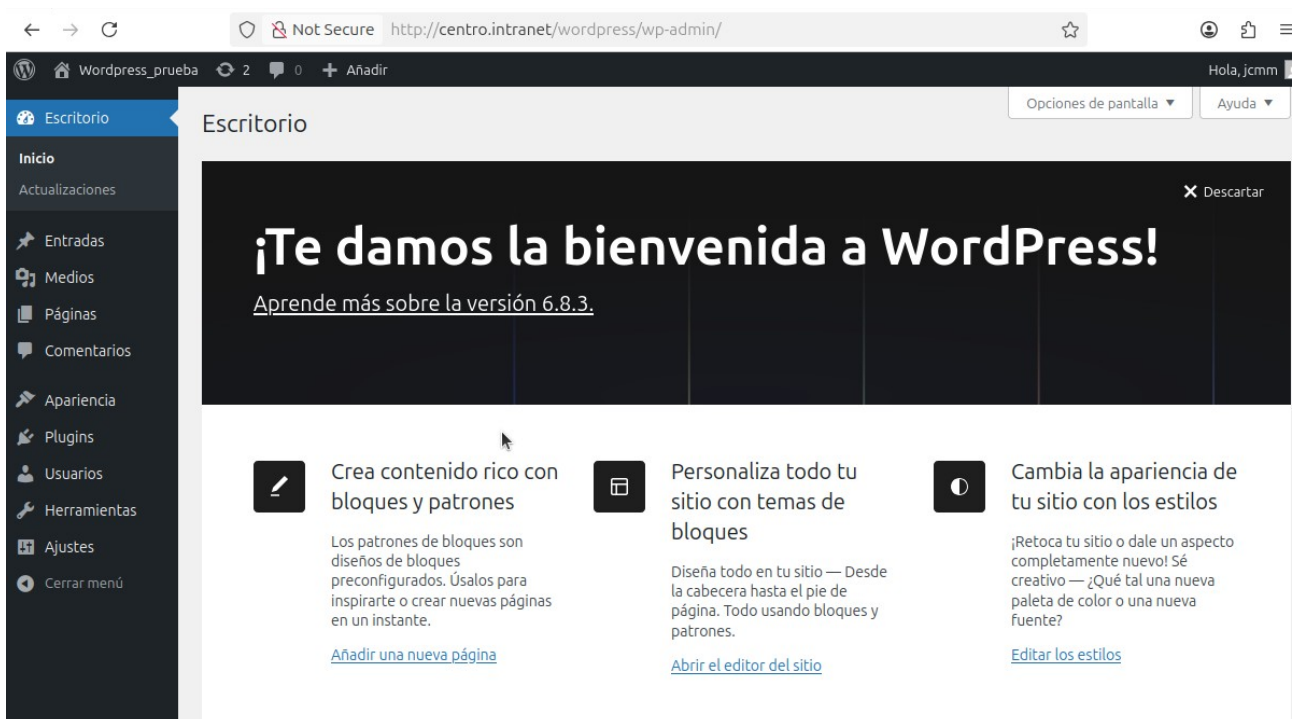
WordPress ya está instalado. ¡Gracias, y que lo disfrutes!

Nombre de usuario jcomm

Contraseña *La contraseña que has elegido.*

[Acceder](#)

Y con esto ya quedaría instalado.



4.- Aplicación Python con WSGI

El dominio departamentos.centro.intranet es el que servirá la aplicación de python. Primero de todo necesitamos instalar el módulo WSGI:

```
jcomm-srei@jcomm-srei:/$ sudo apt install libapache2-mod-wsgi-py3 -y
```

Una vez instalado, vamos a crear la carpeta donde estará el script, así como el propio script:

```
jcm-srei@jcm-srei:/$ sudo mkdir /var/www/departamentos
jcm-srei@jcm-srei:/$ sudo nano /var/www/departamentos/app.py
```

En cuanto al script, voy a usar uno de mi repositorio de github, así que copiaré el código directamente.

Vamos a configurar el VirtualHost para Python:

```
jcm-srei@jcm-srei:/$ sudo nano /etc/apache2/sites-available/departamentos.conf
jcm-srei@jcm-srei:/$
```

```
jcm-srei@jcm-srei: / — sudo nano /etc/apache2/sites-available/departamentos.conf
GNU nano 8.4 /etc/apache2/sites-available/departamentos.conf *
<VirtualHost *:80>
    ServerName departamentos.centro.intranet
    DocumentRoot /var/www/departamentos


    # Config WSGI
    WSGIScriptAlias / /var/www/departamentos/app.py

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

Para terminar, vamos a habilitar el sitio:

```
jcm-srei@jcm-srei:/$ sudo a2ensite departamentos.conf
Enabling site departamentos.
To activate the new configuration, you need to run:
  systemctl reload apache2
jcm-srei@jcm-srei:/$ sudo systemctl reload apache2
jcm-srei@jcm-srei:/$
```

Para demostrar que funciona vamos a hacer lo mismo que para demostrar el wordpress, pero con el dominio “departamentos.centro.intranet”:

← → ↻  No seguro http://departamentos.centro.intranet

Agenda de Contactos (Python WSGI)

- **Juan:** 666111222
- **Maria:** 655444333
- **Pepe:** 611222333

Nota: Esta agenda es de solo lectura porque corre en servidor web.

5.- Proteger Python con Autenticación

Lo primero que hacemos es crear el archivo de contraseñas:

```
jcm-srei@jcm-srei:/$ sudo htpasswd -c /etc/apache2/.htpasswd UsuJesus
New password:
Re-type new password:
Adding password for user UsuJesus
jcm-srei@jcm-srei:/$
```

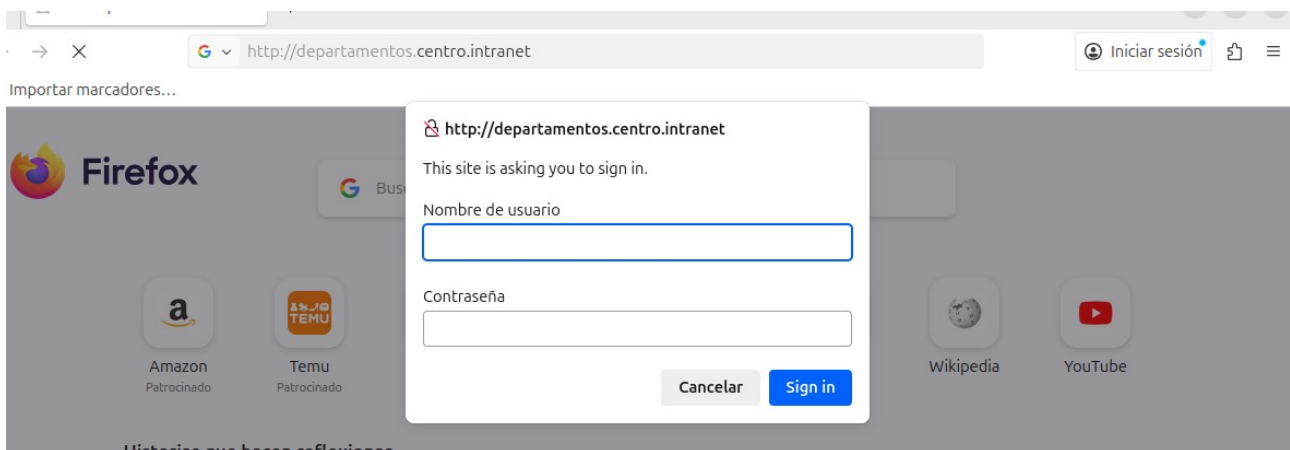
Y ahora vamos a modificar el archivo de configuración del VirtualHost, dentro de directory vamos a añadir el método de autenticación:

```
jcm-srei@jcm-srei: / — sudo nano /etc/apache2/si...
GNU nano 8.4 /etc/apache2/sites-available/departamentos
<VirtualHost *:80>
    ServerName departamentos.centro.intranet
    DocumentRoot /var/www/departamentos

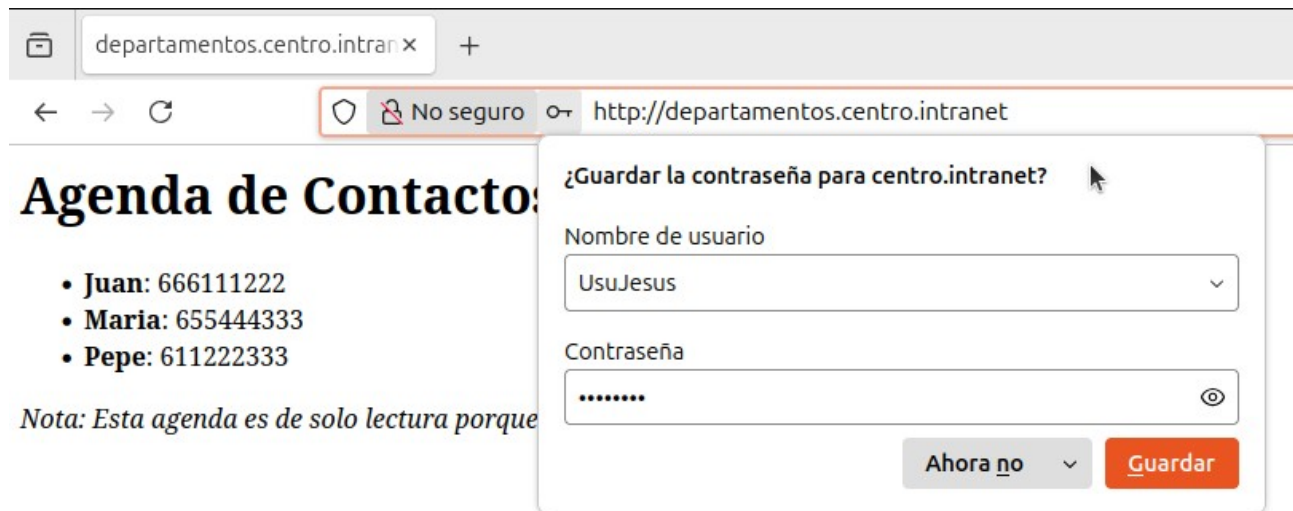
    # Config WSGI
    WSGIScriptAlias / /var/www/departamentos/app.py

    <Directory /var/www/departamentos>
        AuthType Basic
        AuthName "Acceso restringido"
        AuthUserFile /etc/apache2/.htpasswd
        Require valid-user
    </Directory>
</VirtualHost>
```

Recargamos apache y comprobamos que nos pide usuario y contraseña en el navegador:



Ingresamos el nombre y usuario y ya podemos acceder:



The screenshot shows a web browser window with the address bar displaying 'http://departamentos.centro.intranet'. The page title is 'Agenda de Contacto'. Below the title, there is a list of contacts: Juan: 666111222, Maria: 655444333, and Pepe: 611222333. A note below the list states: 'Nota: Esta agenda es de solo lectura porque...'. A modal dialog box is open over the page, asking '¿Guardar la contraseña para centro.intranet?'. It contains a 'Nombre de usuario' field with 'Usu.Jesus' and a 'Contraseña' field with masked characters. At the bottom of the modal, there is a dropdown menu set to 'Ahora no' and a 'Guardar' button.

6.- Instalar y Configurar AWStats

Vamos a instalar awstats, que es una herramienta de estadísticas:

```
jcm-srei@jcm-srei:/$ sudo apt install awstats -y
Los paquetes indicados a continuación se instalaron de forma automática
son necesarios.
  linux-headers-6.17.0-5      linux-tools-6.17.0-5
  linux-headers-6.17.0-5-generic  linux-tools-6.17.0-5-generic
  linux-modules-6.17.0-5-generic
Utilice «sudo apt autoremove» para eliminarlos.

Installing:
  awstats
```

Vamos a configurarla para centro.intranet, copiando la configuración base:

```
jcm-srei@jcm-srei:/$ sudo cp /etc/awstats/awstats.conf /etc/awstats/awstats.centro.intranet.conf
jcm-srei@jcm-srei:/$ sudo nano /etc/awstats/awstats.centro.intranet.conf
```

Y modificamos estas líneas:

```
#
LogFile="/var/log/apache2/centro_access.log"
```

```
#
SiteDomain="centro.intranet"
```

Guardamos el archivo y vamos a habilitar CGI en Apache, puesto que lo necesitamos para ver las gráficas:

```
jcm-srei@jcm-srei:/$ sudo a2enmod cgi
Enabling module cgi.
To activate the new configuration, you need to run:
systemctl restart apache2
jcm-srei@jcm-srei:/$ sudo a2enconf serve-cgi-bin
Conf serve-cgi-bin already enabled
jcm-srei@jcm-srei:/$ sudo systemctl restart apache2
jcm-srei@jcm-srei:/$
```

Como último paso, generamos las estadísticas iniciales:

```
jcm-srei@jcm-srei:/$ 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 7.9 (build 20230108)
From data in log file "/var/log/apache2/centro_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)...
Jumped lines in file: 0
Parsed lines in file: 91
Found 0 dropped records,
Found 0 comments,
Found 0 blank records,
Found 0 corrupted records,
Found 0 old records,
Found 91 new qualified records.
```

Y para comprobarlo, nos vamos al navegador y ponemos lo siguiente:

← → ↻ ☆

Estadísticas de:
centro.intranet

Última actualización: 26 Nov 2025 - 00:57
Periodo mostrado: Mes Nov 2025 Aceptar

Resumen
Cuándo:
Historico Mensual
Días del mes
Días de la semana
Visitas por Horas
Quién:
Países
... Lista completa
Servidores
... Lista completa
... Última visita
... Dirección IP no identificada
Visitas de Robots/Spiders
... Lista completa
... Última visita
Navegación:
Distribución de las visitas

Resumen				
Periodo mostrado	Mes Nov 2025			
Primera visita	26 Nov 2025 - 00:16			
Última visita	26 Nov 2025 - 00:23			
	Visitantes distintos	Número de visitas	Páginas	Solicitudes
Tráfico visto *	1	1 (1 visitas/visitante)	25 (25 Páginas/Visita)	87 (87 Solicitudes)
Tráfico no visto *			3	4

* El tráfico "no visto" es tráfico generado por robots, gusanos o respuestas de código especial de estado HTTP.

7.- NGINX y PHPMyAdmin

Para que no haya interferencias entre apache y nginx, nginx y phpmyadmin lo vamos a poner en el puerto 8080. Primero, lo instalamos. Como Apache usa PHP como módulo, Nginx necesita php-fpm:

```
jcm-srei@jcm-srei:/$ sudo apt install nginx php-fpm -y
```

Si Nginx falla al iniciarse es porque el puerto 80 está ocupado por Apache. Eso es normal, ahora lo cambiamos. Editamos la configuración por defecto:

```
jcm-srei@jcm-srei:/$ sudo nano /etc/nginx/sites-available/default
```

Y editamos las líneas:

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

```
# Add index.php to the list if you are using PHP  
index index.php index.html index.htm index.nginx-debian.html;
```

```
#  
location ~ \.php$ {  
    include snippets/fastcgi-php.conf;  
#  
#    # With php-fpm (or other unix sockets):  
fastcgi_pass unix:/run/php/php7.4-fpm.sock;  
#    # With php-cgi (or other tcp sockets):  
fastcgi_pass 127.0.0.1:9000;  
#}
```

Y reiniciamos nginx con sudo systemctl restart nginx. Comprobamos su estado:

```
jcm-srei@jcm-srei:/$ sudo systemctl status nginx  
● nginx.service - A high performance web server and a reverse proxy server  
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)  
   Active: active (running) since Wed 2025-11-26 01:07:08 CET; 11s ago  
 Invocation: 225e31a630bb4918adb29a5bc3833f9e  
    Docs: man:nginx(8)  
  Process: 32197 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)  
  Process: 32199 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)  
 Main PID: 32200 (nginx)  
   Tasks: 5 (limit: 8855)  
  Memory: 4.5M (peak: 5M)  
     CPU: 56ms  
   CGroup: /system.slice/nginx.service  
           └─32200 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"  
             └─32201 "nginx: worker process"  
               └─32202 "nginx: worker process"  
                 └─32203 "nginx: worker process"  
                   └─32204 "nginx: worker process"  
  
nov 26 01:07:08 jcm-srei systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...  
nov 26 01:07:08 jcm-srei systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.  
jcm-srei@jcm-srei:/$
```

Lo siguiente va a ser instalar PHPMyAdmin, con sudo apt install phpmyadmin -y, y se nos va directamente a la configuración de phpmyadmin.

No marcamos ninguna de las dos opciones, nos vamos directamente a Aceptar:

Configuración de phpmyadmin

Por favor, elija el servidor web que se debería configurar automáticamente para que ejecute phpMyAdmin.

Servidor web que desea reconfigurar automáticamente:

☐ apache2
☐ lighttpd

<Aceptar>

Configuración de phpmyadmin

El paquete phpmyadmin debe tener una base de datos instalada y configurada antes de poder ser utilizado. Puede gestionar esto opcionalmente a través «dbconfig-common».

Debería rechazar esta opción si es Vd. un administrador de bases de datos avanzado y desea realizar esta configuración manualmente, o si la base de datos ya está instalada y configurada. Probablemente podrá encontrar los detalles de las operaciones que debe realizar en «/usr/share/doc/phpmyadmin».

Debería escoger esta opción en cualquier otro caso.

¿Desea configurar la base de datos para phpmyadmin con «dbconfig-common»?

<Si> **<No>**

Configuración de phpmyadmin

Proporcione una contraseña para que phpmyadmin se registre con el servidor de base de datos. Si deja este campo en blanco se generará un contraseña aleatoria.

Contraseña de aplicación MySQL para phpmyadmin:

<Aceptar> **<Cancelar>**

Una vez instalado, vamos a vincular a nginx mediante un enlace simbólico.

```
jcm-srei@jcm-srei:/$ sudo ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin
```

Y para terminar vamos a <http://servidor2.centro.intranet:8080/phpmyadmin> :

phpMyAdmin

No seguro http://servidor2.centro.intranet:8080/phpmyadmin/ Iniciar sesión

phpMyAdmin
Bienvenido a phpMyAdmin

Idioma (Language)
Español - Spanish

Iniciar sesión

Usuario:

Contraseña:

Iniciar sesión