Contenido

[ENRUTAR 1](#_Toc97144924)

[DHCP 1](#_Toc97144925)

[DNS 1](#_Toc97144926)

[FTP 4](#_Toc97144927)

[HTTP 4](#_Toc97144928)

[LAMP 7](#_Toc97144929)

# ENRUTAR

**apt install iptables**

**echo "1" > /proc/sys/net/ipv4/ip\_forward**

**iptables -A FORWARD -j ACCEPT**

**iptables -t nat -A POSTROUTING -o enp0s? -j MASQUERADE**

# DHCP

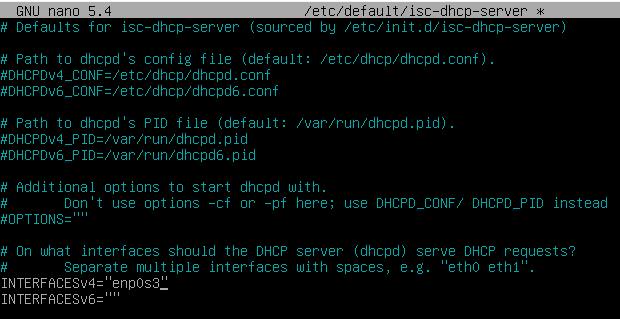
**PRIMERO TENER IP’S**

**apt install isc-dhcp-server**

Dirigirse al primer archivo de configuración

**nano /etc/default/isc-dhcp-server**

Cambiar la directiva poniendo el nombre de la tarjeta de red (generalmente enp0s3)

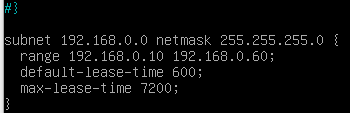


Dirigirse al otro archivo de configuración de DHCP.

**nano /etc/dhcp/dhcpd.conf**

**Ámbito estándar**

SUBNET TIENE QUE ACABAR EN 0

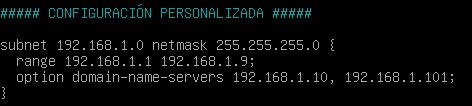


**range:** el rango de direcciones que repartirá el ámbito, en este caso desde la 0.10 hasta la 0.60.

**default-lease-time:** el tiempo de concesión de una IP por defecto, en segundos, en este caso 600.

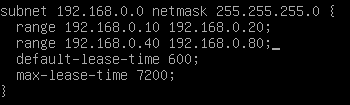
**max-lease-time**: el tiempo máximo de concesión de una IP, en segundos, en este caso 7200.

**Ámbito con servidor DNS**



**option domain-name-servers**

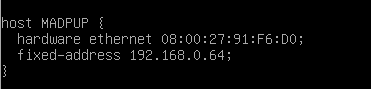
**Ámbito con exclusión**



**Reserva**

(SE PONE FUERA DEL AMBITO)

**host (CUALQUIER NOMBRE)**



**hardware ethernet:** este campo especifica la dirección MAC de la máquina que recibirá la reserva.

**fixed-address:** este campo especifica la dirección IP que se reservará para dicha máquina, en este caso he elegido la 192.168.0.64.

**REINICIAR SERVICIO**

**service isc-dhcp-server restart**

# DNS

Instalamos el paquete.

**apt install bind9**

Cambiamos el resolv para añadir los servidores DNS primario y secundario.



Nos dirigimos al directorio del servicio */etc/bind.*



Editamos el archivo named.conf.local donde establecemos las zonas con la ruta del archivo de zona y si es maestro (primario) o esclavo (secundario).

**NAMED.CONF.LOCAL**

**nano */etc/bind/named.conf.local***

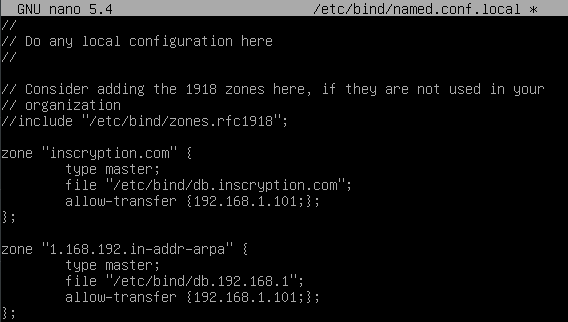


Ilustración 1 Ejemplo de zona directa e inversa

Como se puede ver, he dado de alta la zona “***inscryption.com***” y su equivalente inversa con la dirección de red de la zona, “***1.168.192.in-addr-arpa***”.

* En el campo “*type”* he establecido *master* en ambas zonas, puesto que este se trata del servidor DNS primario.
* En el campo *“file”* he establecido la ruta de la base de datos de cada zona, siendo */etc/bind/db.inscryption.com* en la zona de búsqueda directa y */etc/bind/db.192.168.1* en la zona de búsqueda inversa.
* También he añadido el campo *“allow-transfer”* para especificar la máquina que podrá recibir una transferencia de zona, en este caso la **192.168.1.101** que será el servidor DNS secundario.

Se añadirían mas zonas con sus respectivas inversas si se necesitase.

ARCHIVOS DE ZONA

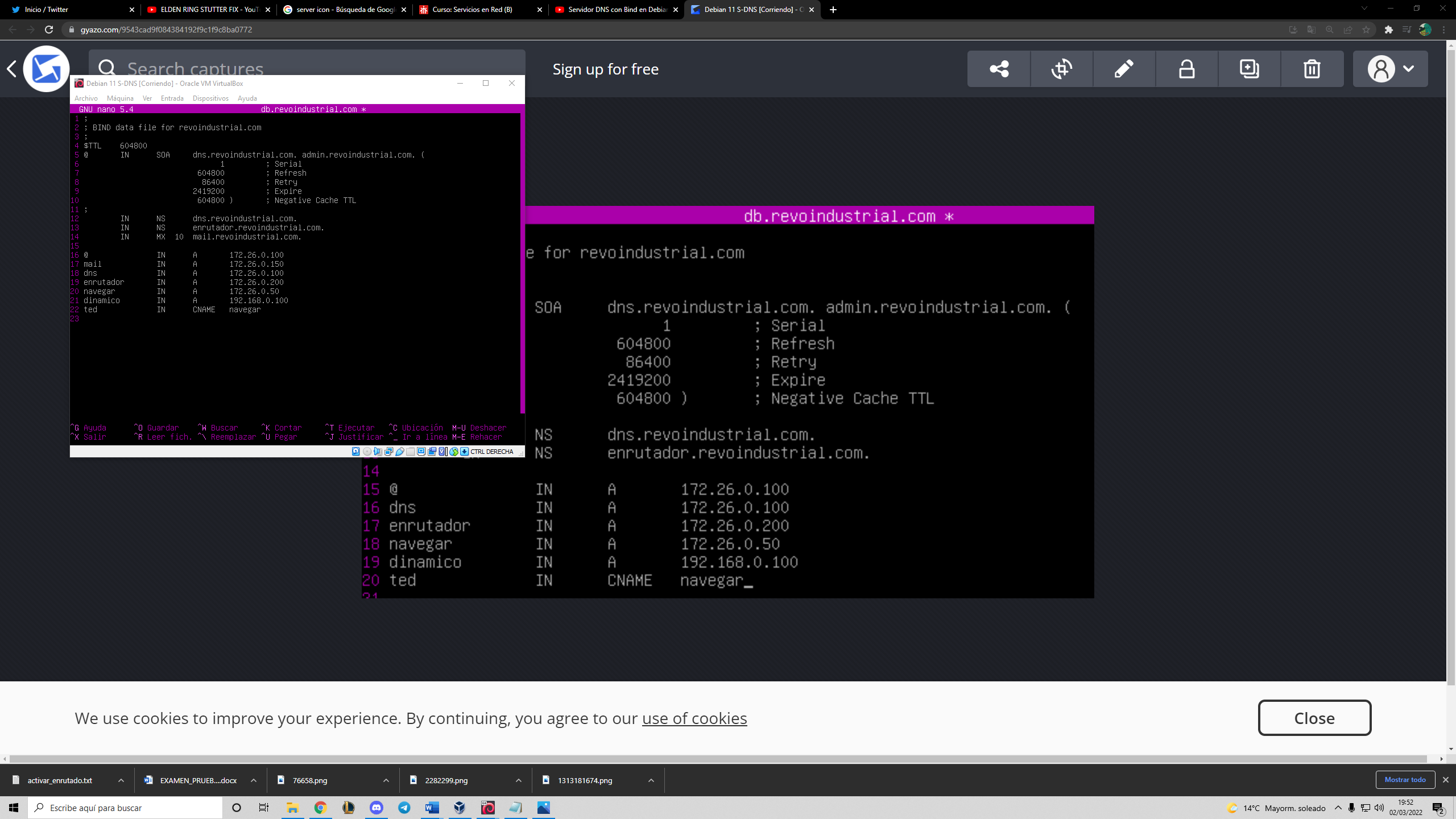
Para crear un archivo de resolución directa se copa el archivo db.local.

**cp *db.local db.revoindustrial.com***

Ejemplo de archivo de zona:

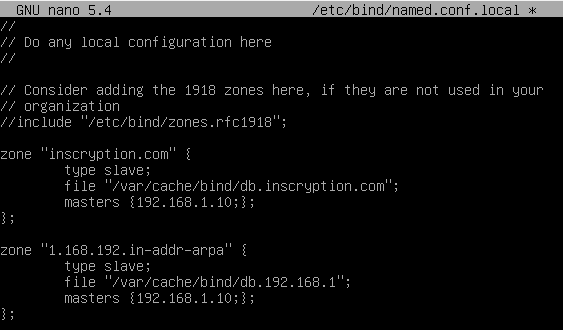


Ejemplo con registro MX



**REALIZAR LA TRANSFERENCIA DE ZONA EN EL SERVIDOR SECUNDARIO**

Después de instalar el paquete Bind9 en el servidor secundario se dan de alta las zonas como en el servidor primario solo que añadiendo de tipo **slave** y cambiando la ruta a **/var/cache/bind/(nombredearchivo)**



Después se solicita la transferencia de la zona con el comando

**dig inscription.com @192.168.1.10 axfr**

# FTP

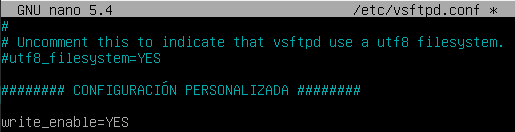
Instalamos el paquete

**apt install vsftpd**

Para habilitar la subida de archivos, nos dirigimos al archivo /etc/vsftpd.conf

**nano /etc/vsftpd.conf**

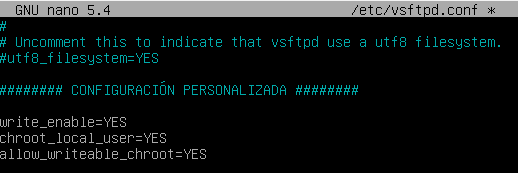
Añadimos a mano la siguiente directiva para permitir la subida de archivos. **write\_enable=YES.**



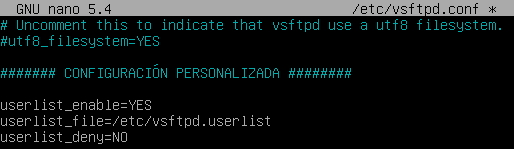
Para enjaular a los usuarios se añaden las directivas:

**chroot\_local\_user=YES**

**allow\_writeable\_chroot=YES**



Para solo dar acceso a algunos usuarios se añaden las siguientes directivas:



Y se modifica el archivo con la lista de usuarios.

**nano /etc/vsftpd.userlist**

Y se añaden los usuarios.



Para cambiar el directorio de conexión de un usuario se cambia su directorio *home*.

**usermod -d /nuevo/directorio -m nombredeusuario**

Se reinicia el servicio

**service vsftpd restart**

# HTTP

Instalamos el paquete.

**apt install apache2**

Instalamos también SSH para pasar los archivos por SCP.

**apt install ssh**

Configuramos que root pueda iniciar sesión en SSH.

**nano */etc/ssh/sshd\_config***

Añadimos/descomentamos la línea **PermitRootLogin yes**



Creamos el/los directorios que alojarán el/los sitios.

**mkdir /var/www/html1**

Nos dirigimos a la ruta **/etc/apache2/sites-available**

Copiamos el archivo 000-default.conf para cada sitio que queramos crear.

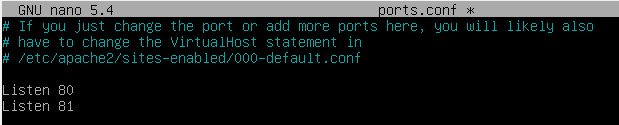
**cp *000-default.conf sitio.conf***

Editamos el archivo .conf nuevo

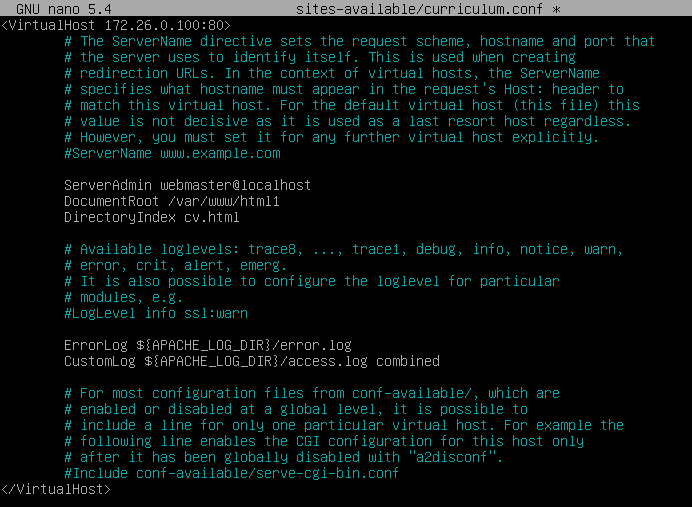
**VirtualHost** escribimos como se accede a este sitio (IP/nombre y puerto)



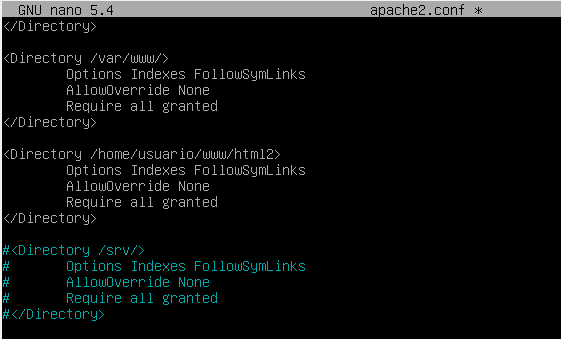
En caso de cambiar el puerto se tiene que especificar en el archivo **ports.conf** donde se añade una directiva **Listen** con el puerto que sea.



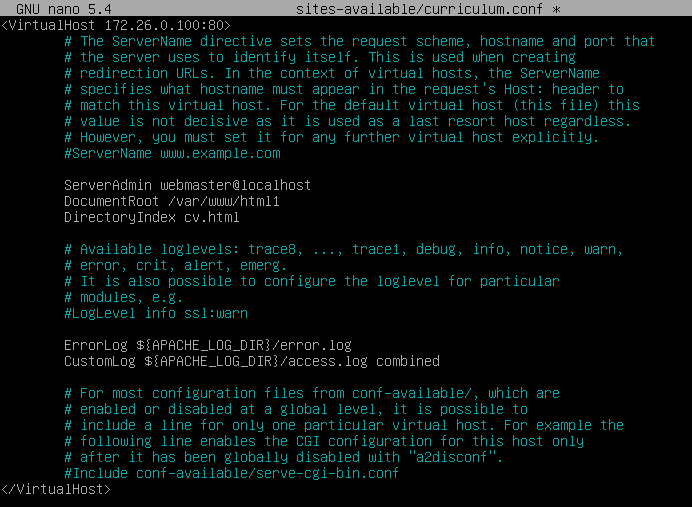
Cambiamos la directiva **DocumentRoot** para que apunte a la ruta donde esta el sitio (archivo html)



En caso de especificar cualquier ruta que NO sea **/var/www/** hay que dirigirse al archivo **apache2.conf** y añadir una directiva **<Directory *(ruta)*>** copiando una ya hecha.



En caso de que el archivo html NO se llame **index.html** hay que cambiar la directiva **DirectoryIndex** seguido de como se llame el archivo html.

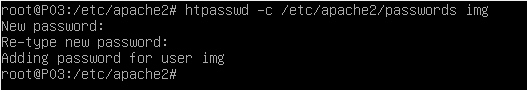


**PARA REQUERIR USUARIO Y CONTRASEÑA PARA VER UN SITIO**

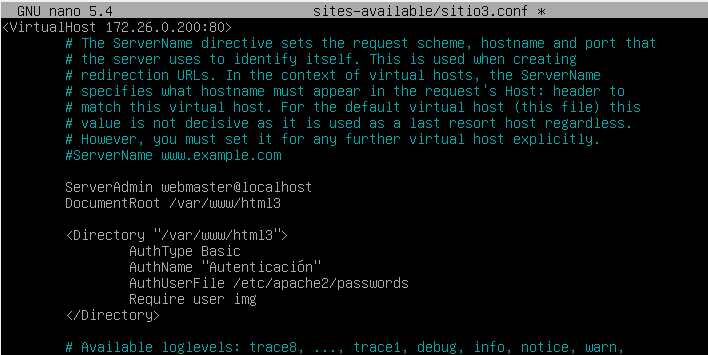
Creamos un archivo de contraseña con el siguiente comando, especificando también su primer usuario. (para añadir mas se omite la **-c**)

**htpasswd -c */etc/apache2/passwords* img**

Nos pedirá una contraseña



En el archivo .conf del sitio añadimos la directiva **<Directory>** con lo siguiente:



Una vez creados los archivos de sitio deshabilitamos el sitio por defecto.

**a2dissite *000-default.conf***



Y habilitamos los sitios que queramos.

**a2ensite sitio*.conf***

Reiniciando después el servicio.

**service apache2 restart**

# LAMP

Descargamos los paquetes

**apt install apache2 mariadb-server php**

Paquetes adicionales de WordPress:

**apt install php-mysql php-xml**

Paquetes adicionales de Moodle:

**apt install php-curl php-zip**

Descargamos el CMS, ubicándonos en la carpeta donde queramos tenerlo, ejemplo /var/www/wordpress.

**wget** [**https://es.wordpress.org/latest-es\_ES.tar.gz**](https://es.wordpress.org/latest-es_ES.tar.gz)

**CREAR BASE DE DATOS**

Para acceder a los comandos de la base de datos usamos el comando **mysql**

Creamos la base de datos con el nombre que sea.

**CREATE DATABASE *dbwordpress*;**

Creamos el usuario, especificando la máquina (localhost) y la contraseña.

**CREATE USER ‘*monke*’@‘*localhost*’ IDENTIFIED BY ‘banana’;**

Le damos privilegios de administrador y refrescamos privilegios.

**GRANT ALL PRIVILEGES ON \*.\* TO ‘*monke*’@‘*localhost’;***

**FLUSH PRIVILEGES;**

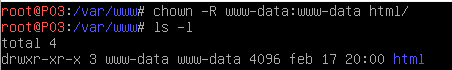
Salimos de maria.db con **QUIT**

Descomprimimos el archivo con el WordPress.

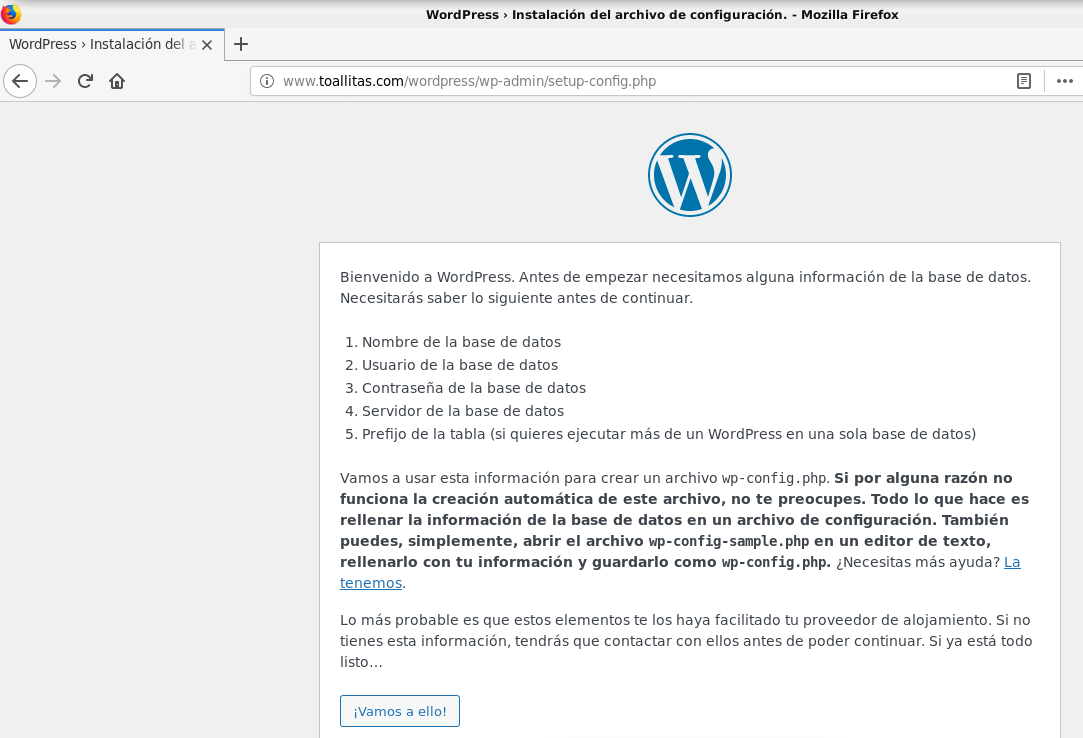
**tar -xzvf *latest-es\_ES.tar.gz***

Una vez extraído he cambiado el propietario de la carpeta donde se encuentrede manera recursiva para que no hubiera problema durante la instalación.

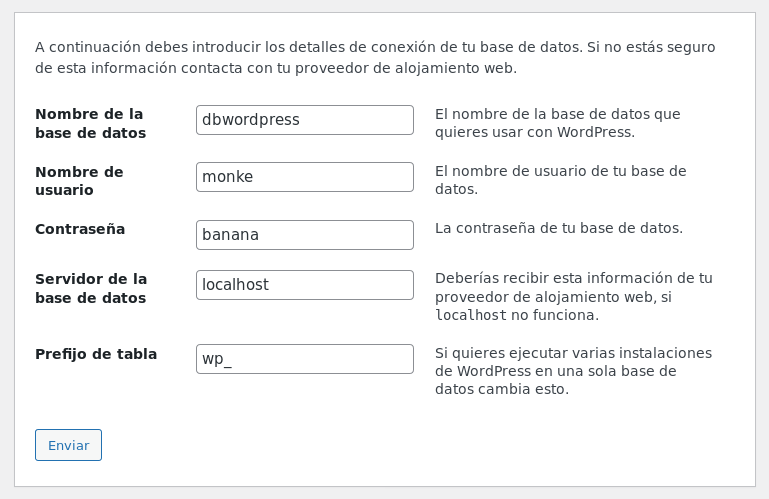
**chown -R www-data:www-data html/**



Abrimos la página del wordpress desde un navegador (tras haber creado el .conf del sitio etc etc)



Escribimos la información de la base de datos.



Creamos una cuenta y accedemos al WordPress.