2º DAW

Alfonso Navarrete
Alex Dominguez
Leobardo Ríos
Gabriel Miralles

Proyecto intermodular

20 de febrero del 2021

Instalación configuración y administración del servidor web para el proyecto intermodular

Proyecto realizado por los alumnos del curso 2º Desarrollo de aplicaciones web 2020/2021.

Objetivos

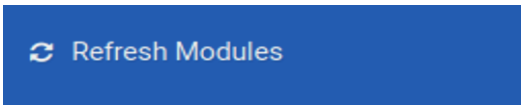
1. **Definir los nombres de Hosts necesarios (DNS) en una zona que debe estar correctamente replicada entre un servidor maestro y otro esclavo.**
2. **Instalar un sistema LAMP en una máquina local**
3. **Configurar las opciones necesarias en Apache, MariaDB y PhpMyAdmin**
4. **Configurar los VirtualHosts necesarios en Apache**
5. **Acceder remotamente a PhpMyAdmin**

1. Configuración del DNS

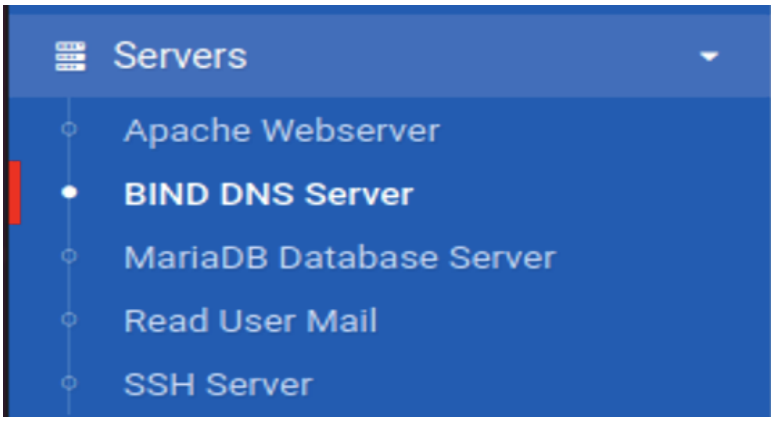
Para poder realizar la administración de los dominios del servidor hemos utilizado BIND9, con el fin de gestionar todos los dominios y zonas que posee el proyecto, para proceder a su instalación empleamos el comando:

```
sudo apt-get install bind9
```

Una vez instalado el servicio bind9, desde Webmin accedemos a la pestaña:



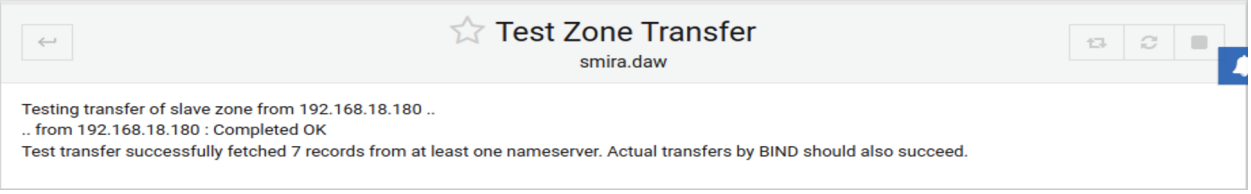
Una vez refrescada la página, aparecerá dentro de la pestaña “Servers”, la opción:



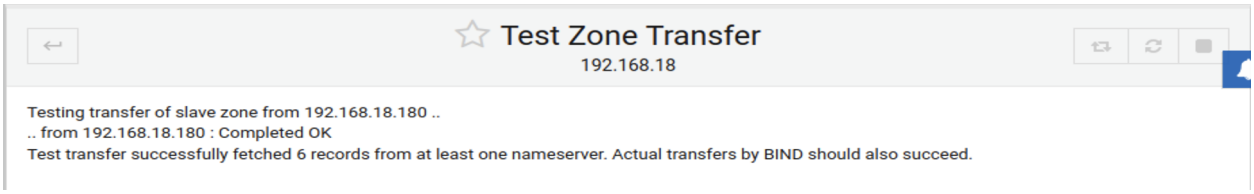
que nos da acceso a toda la configuración de BIND9, a través de la interfaz de Webmin. para el proyecto intermodular hemos creado una zona maestra denominada “smira.daw” con su correspondiente zona inversa, cuyo nombre atiende a la red interna que hemos generado para el proyecto de tipo D, la dirección de inicio de esta red es 192.168.18., la direccion de finalizacion es 192.168.18.254 y la máscara de subred quen se obtiene a partir de esta red es 255.255.255.0, nuestra zona maestra de resolución inversa tiene como dominio “192.168.18”.

Para la zona esclava, hemos clonado la máquina virtual con Debian 11 que teníamos ya configurada, además de cambiarle la dirección IP fija, que en nuestro caso hemos asignado la

IP 192.168.18.182. Después de generar ambas zonas esclavas, tanto la resolución directa, como la resolución inversa, utilizando la opción de “Test Zone Transfer”, en ambas resoluciones, se duplican todos los registros generados en la zona maestra a la zona esclava.



Zona de resolución directa



Zona de resolución inversa

Después de generar los registros necesarios, que en nuestro caso han sido tres registros A cuyo nombre son smira.daw, ns1.smira.daw. y adminsdb.smira.daw., un CNAME que corresponde a www. cuyo valor es smira.daw. y un registro de tipo NS smira.daw cuyo valor es ns1.smira.daw; hemos podido transferirlas a la zona esclava sin ningún problema.

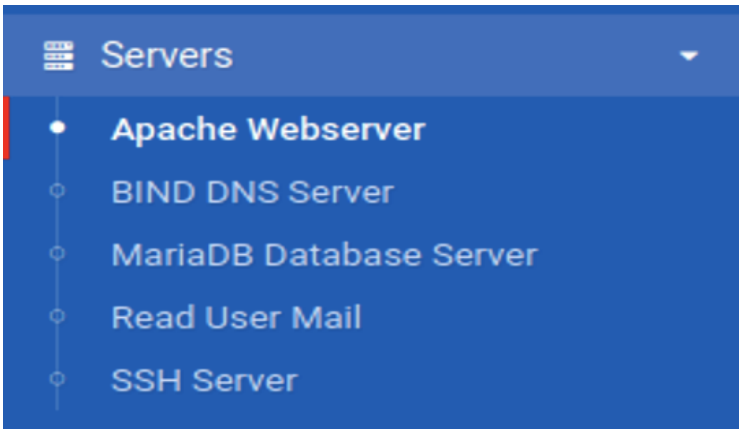
2. Instalación del servidor web Apache

En distribuciones como Debian resulta más sencillo realizar la instalación de Apache desde un repositorio aunque también podríamos hacerlo desde su código fuente.

Por esta razón decidimos instalarlo mediante:

```
sudo apt install apache2
```

Una vez instalado, para integrarlo con Webmin pulsamos sobre Refresh Modules y tras actualizar los módulos disponibles, las herramientas de administración de Apache2 aparecerán en el menú Servers.



Todas las configuraciones para los módulos se encuentran en:

```
/etc/apache2/mods-available
```

Y se pueden activar y desactivar según nos convenga con:

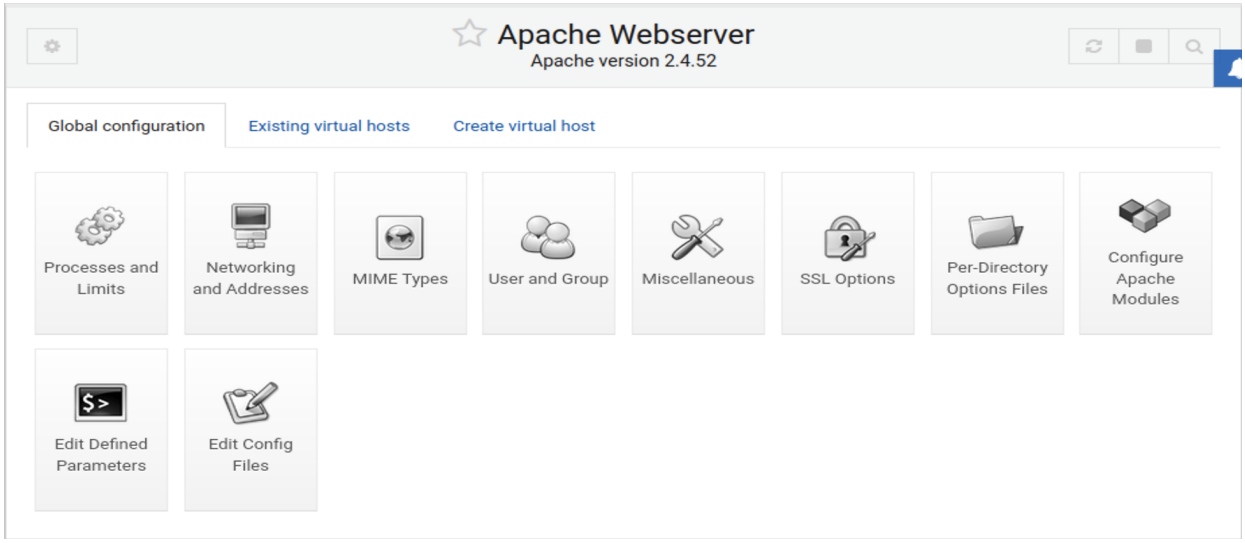
```
sudo a2enmod nombre-modulo
```

```
sudo a2dismod nombre-modulo
```

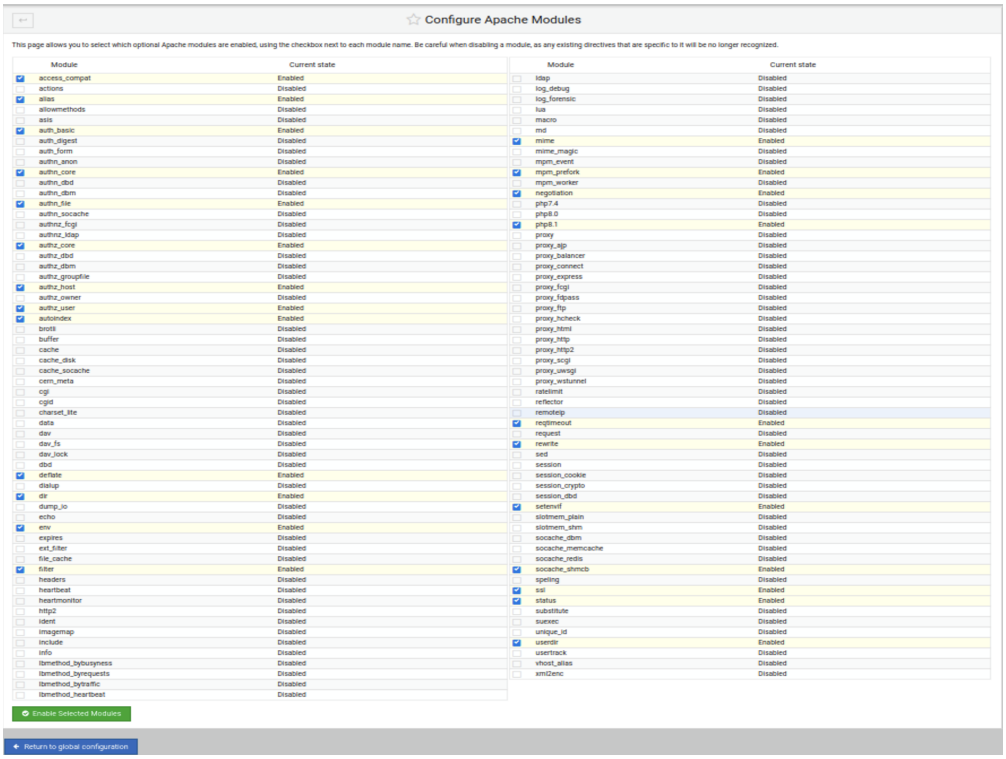
Aunque también es posible gestionarlos desde Webmin en:

Servers → Apache Webserver

Pestaña: Global configuration > Configure Apache Modules



Configuración global de Apache desde Webmin



Módulos de Apache activados para nuestro proyecto

3. Configurar Apache, MariaDB y PhpMyAdmin

Para poder gestionar nuestra base de datos creadas con MariaDB desde PhpMyAdmin de forma segura, gracias a nuestro servicio de Apache, antes de nada, es necesario crear un Certificado SSL para poder tener nuestras conexiones seguras, gracias al cifrado criptográfico que realiza.

El primer paso a realizar es habilitar el módulo SSL dentro de Apache2, o bien desde Webmin o a través de la terminal de comandos gracias al comando:

```
sudo a2enmod ssl
```

Una vez activado es necesario recargar la configuración de Apache reiniciando:

```
sudo systemctl restart apache2
```

Tras estos pasos, ya podemos habilitar Virtual Host con conexión segura HTTPS, pero antes debemos crear un certificado, que en nuestro caso es autofirmado. Para poder crearlos debemos instalar los paquetes openssl y ca-certificates con el comando:

```
sudo apt install openssl ca-certificates
```

En nuestro caso los certificados los almacenamos en /etc/apache2/ssl. Una vez situados sobre esta carpeta para poder crearlos, primeramente generamos la clave privada con el comando:

```
sudo openssl genrsa -out server.key 2048
```

```
sudo chmod 600 server.key
```

A continuación generamos el archivo CSR (Certificate Signing Request):

```
sudo openssl req -new -key server.key -out server.csr
```

Nos pedira rellenar la informacion necesaria para identificar nuestro certificado y una vez generado, apartir de este archivo .csr podemos generar el certificado autofirmado con el comando:

```
sudo openssl x509 -req -days 365 -in server.csr -signkey server.key
-out server.crt
```

Para la instalación de MariaDB y PHP hemos utilizado los siguiente comandos:

```
sudo apt-get install mariadb
```

```
sudo apt-get install php
```

Una vez instalados, para comprobar que MariaDB está instalado utilizamos el comando:

```
mariadb
```

```
root@mrclusterman-vm:/etc/apache2/sites-available# mariadb
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 32
Server version: 10.6.5-MariaDB-1:10.6.5+maria~bullseye mariadb.org binary distribution

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)]>
```

Muestra por pantalla la consola de comando de MariaDB

En caso de tenerlo bien instalado, la respuesta por pantalla debería ser parecida a la captura que se muestra anteriormente. Por otro lado, para comprobar que PHP está instalado, con el comando siguiente muestra la versión instalada:

```
php -v
```

```
root@mrclusterman-vm:/etc/apache2/sites-available# php -v
PHP 8.1.2 (cli) (built: Jan 27 2022 12:22:31) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2, Copyright (c), by Zend Technologies
root@mrclusterman-vm:/etc/apache2/sites-available#
```

En nuestro caso hemos instalado la versión de PHP 8.

La configuración de PhpMyAdmin, será realizada a través de un proyecto de Laravel así que únicamente precisamos de un Virtual Host que apunte a la carpeta que contiene el proyecto.

4. Configurar los Virtual Hosts necesarios en Apache

Una de las configuraciones básicas tras la instalación de Apache es la creación del virtual host. Para ello nos dirigimos a:

`/etc/apache2/sites-available`

desde donde creamos y configuramos nuestro host virtual.

En nuestro caso hemos asignado la dirección IP –> 192.168.18.180 con el puerto de escucha 4444. Además, tanto nuestro ServerName como nuestro ServerAlias los hemos llamado:

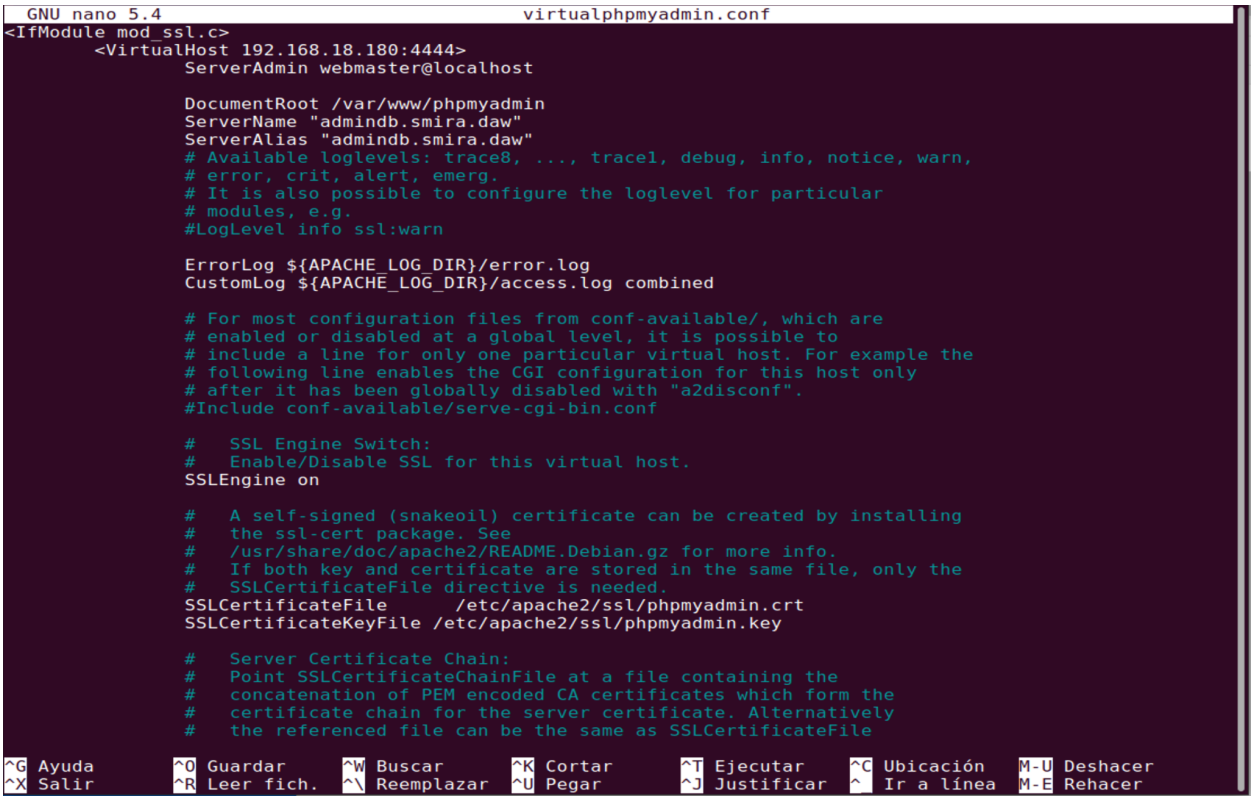
`“admindb.smira.daw”`

Hemos alojado nuestras páginas web del servidor en el directorio:

`/var/www/phpmyadmin`

Y los certificados autofirmados en:

`/etc/apache2/ssl`



Ejemplo VirtualHost para pagina 192.168.18.180:4444(PHPMYADMIN)

Una vez lo tenemos creado lo activamos mediante:

```
sudo a2enmod virtualphpmyadmin.conf
```

5. Acceder remotamente a PHPMyAdmin.

Para poder administrar de forma remota nuestras Bases de Datos, es necesario crear un VirtualHost con certificado SSL, una vez creado, es necesario crear una nueva base de datos desde cuyo nombre será *phpmyadmin* para ello haremos uso del siguiente comando dentro de la consola de MariaDB:

```
CREATE DATABASE `phpmyadmin`

DEFAULT CHARACTER SET utf8 COLLATE utf8_bin;
```

A continuación, crearemos un usuario para la gestión de esta base de datos, con el nombre admin:

```
CREATE USER admin@localhost

IDENTIFIED BY '**********'
```

Una vez creado, debemos otorgarle todos los permisos con el comando:

```
GRANT ALL PRIVILEGES

ON phpmyadmin. * TO admin@localhost;

FLUSH PRIVILEGES;
```

Salimos y desde /var/www/phpmyadmin desde donde hemos descargado la configuración , ejecutamos el script de create_tables.sql y editamos el archivo config.inc.php.

Una vez realizados estos pasos podremos acceder a phpMyAdmin desde la url asignada en su VirtualHost.