



Universidad Nacional  
Autónoma de México

DIRECCIÓN GENERAL DE CÓMPUTO Y DE  
**TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN**

**03-marzo-2016**

# PROYECTO MODULO 2 “Documento Técnico”

## **Integrantes:**

- Armenta Segura José Juan
- Morales Pablo Samuel Abraham
- Vega Tellez Jesus Antonio



Programa de Becas de Formación en Seguridad  
Informática 10ª Generación

# **Contenido**

Descripción General .....	2
Actividades a desarrollar .....	2
• Back End .....	2
• Front End .....	2
✓ Vistas Del Front End .....	2
• Protección Del Front End .....	2
Diagrama .....	3
Introducción .....	4
Metas .....	4
Requerimientos .....	4
Instalaciones .....	5
Back End.....	5
Instalación Drupal 8.....	5
Instalación sitios .....	6
Instalación PostgreSQL .....	7
Instalación Active Directory .....	8
Front End .....	10
Drupal Console .....	10
Servidor Nginx.....	10
Creando el sitio para Drupal .....	11
Activar FastCGI un Nginx .....	11
Instalar Drupal en Nginx.....	12
Conexión cifrada para el front-end .....	14
Modulo Front End.....	16
Nginx ModSecurity .....	17
ModSecurity .....	17
Proxy inverso .....	27
REFERENCIAS .....	30

## Descripción General

Implementación de una arquitectura de contenido distribuida que comunique mediante Web Services la capa de presentación Front End con la capa de lógica de negocio y contenido Back End para la creación de un concentrador de contenidos Web.

## Actividades a desarrollar

- Back End

El Back será un Drupal 8 multisite que deberá proveer 2 sitios web, donde se programará un módulo que implemente algún web service para transmitir el contenido almacenado en el sitio al servidor Front End, se puede hacer eso de REST, SOAP, JSON o algún otro mecanismo bien conocido para implementar el servicio web.

La instalación del Drupal 8 será utilizando el binario Drupal console.

Adicionalmente se deberá tener un equipo con Windows Server 2012 R2 instalado con un Active Directory habilitado para autenticar usuarios.

La base de datos del servidor Back End se implementara con el manejador PostgreSQL y un servidor web Apache 2.4, donde se deberá configurar el servidor web apache para que agregue una autenticación extra por medio de Active Directory para poder acceder al sitio Drupal.

- Front End

Desarrollo e implementación de una arquitectura de contenidos distribuida que comunique mediante Web Services al Front End con la capa de lógica del negocio y contenido Back End, para mostrar el contenido de los sitios generados por el Back End.

Se implementara un Drupal 8 y se programara un módulo, la función de esta aplicación es mostrar la información obtenida mediante Web Services desde el Back End y presentarlo, Los usuarios podrán ver los diferentes contenidos.

La instalación de Drupal 8 será utilizando el binario Drupal console. El Front End se debe implementar en un servidor web Nginx y el manejador será PostgreSQL.

- ✓ Vistas Del Front End

Debe mostrar 2 entradas de contenido, una por cada sitio del Drupal del Back End. El contenido se obtiene a través del Web Service proporcionado por el Back End.

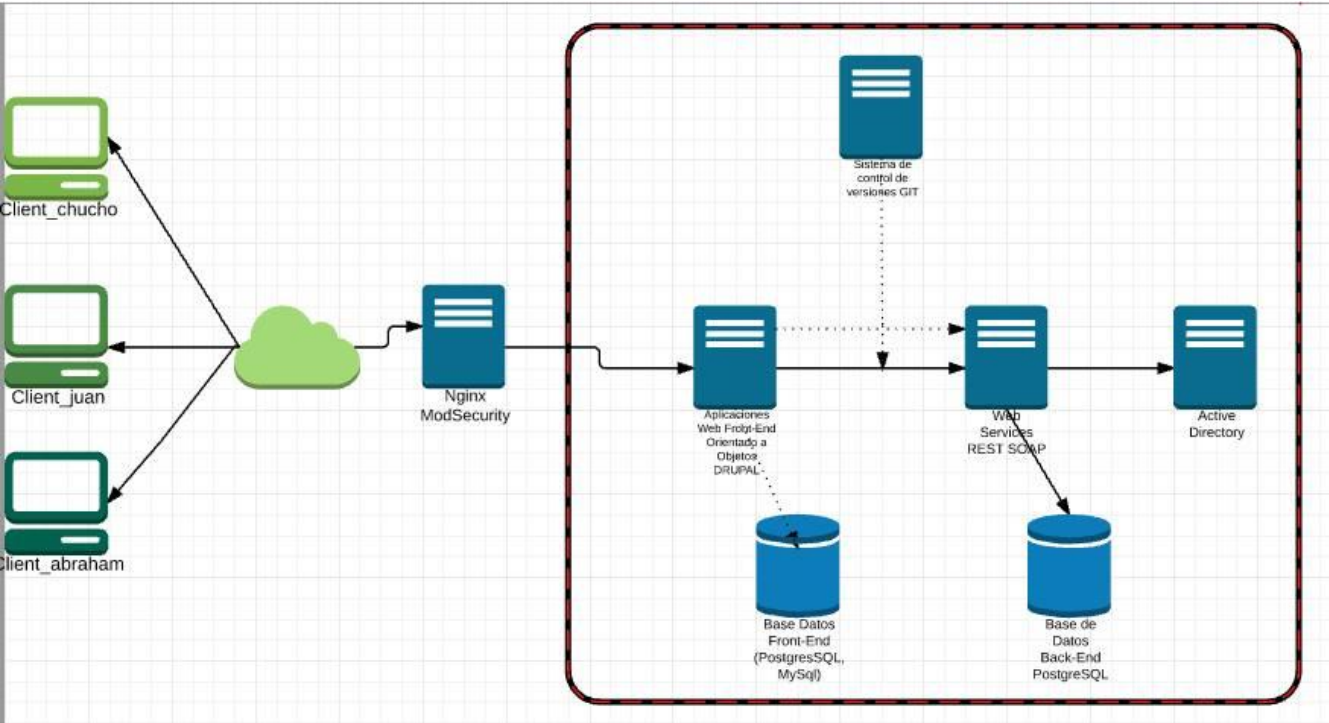
- Protección Del Front End

El Front End deberá ser protegido con un WAF que se encuentre en un equipo separado operando con un servidor web Nginx y utilizando ModSecurity como proxy inverso.

Este WAF podrá operar en los varios modos que se tiene ModSecurity:

- ✓ Solo detección
- ✓ Apagado
- ✓ Encendido

Diagrama



## Introducción

El presente proyecto titulado "Implementación de aplicaciones web de forma segura", tiene como finalidad la aplicación de los conocimientos adquiridos a lo largo del segundo módulo. El proyecto comprende el establecimiento de un Black End que provee servicio a 2 sitios web y utiliza Web Service para transmitir contenido almacenado; Aunado a lo anterior, también se cuenta con Active Directory para realizar las autenticaciones pertinentes. Para mostrar los contenidos a los usuarios, se desarrolló una arquitectura de contenidos distribuida que mediante Web Service comunica al Front End y al Back End.

## Metas

Implementación de arquitectura de contenido distribuida. Establecer comunicación entre la capa de presentación (Front End) y la capa lógica de negocio mediante web Services. Creación de un concentrado de contenidos Web. Desarrollo de la totalidad del proyecto en el tiempo asignado. Entrega del proyecto final del módulo 2 el día 3 de marzo de 2016 a las 18:00 horas Obtención de una calificación aprobatoria (superior a 8.5) en la evaluación del proyecto realizado.

Implementar un WAF (Web Application Firewall) que se encuentre en un equipo separado operando con un servidor web Nginx

Implementar un proxy inverso utilizando ModSecurity

Creación de una base de datos (back End) usando PostgreSQL en un servidor apache2.4

## Requerimientos

Requerimiento	Prioridad	Descripción
<b>Modulo Back End</b>	Alta	Programación de un módulo que implemente un Web Service y este envíe el contenido del sitio al Front End.
<b>Modulo Front End</b>	Alta	Programación de un módulo que, mediante Web Service obtenga el contenido desde el Back End y lo presente.
<b>Back End Apache y Drupal 8</b>	Alta	Implementar apache con instalación de Drupal 8 multisite, haciendo uso de Drupal console
<b>Servidor Active Directory</b>	Media	Implementación para autenticación ldap para los sitios de Back End. El sistema deberá ser capaz de crear, modificar y eliminar usuarios de Active Directory para que estos puedan loguearse al ingresar a dichos sitios.
<b>Front End Nginx y Drupal 8</b>	Alta	Implementación de Nginx con instalación de Drupal 8, haciendo uso de Drupal console
<b>WAF Nginx</b>	Alta	El Front End debe estar protegido con un WAF, el cual debe estar implementado en otro servidor.
<b>Nginx proxy inverso</b>	Alta	El servidor donde se encuentra el Front End deberá estar protegido con un servidor proxy inverso.
<b>Bitácoras Nginx</b>	Media	Todas las peticiones maliciosas al servidor que tiene implementado ModSecurity y los accesos deben registrarse en bitácora.
<b>Servidor PostgreSQL</b>	Media	Implementación de un servidor que contenga las bases de datos usadas para la implementación de los sitios en Drupal.
<b>Respaldos incrementales PostgreSQL</b>	Baja	Las bases de datos deben contar con respaldos incrementales.

# Instalaciones

## Back End

### Instalación Drupal 8

Debido a que se pide que para el back End se instale Drupal 8 sobre apache, como primer paso instalamos Apache.

- apt-get install apache2 apache2-doc

Uno de los prerequisites para la instalación de Drupal 8 Console es la instalación de las siguientes bibliotecas.

- apt-get install php5 php5-gd
- apt-get install curl libcurl3 libcurl3-dev php5-curl
- apt-get install sqlite3 libsqlite3-dev
- apt-get install php5-pgsql

Ahora descargamos Drupal console y hacemos las siguientes configuraciones necesarias.

- curl https://drupalconsole.com/installer -L -o drupal.phar
- mv drupal.phar /usr/local/bin/drupal
- chmod +x /usr/local/bin/drupal
- drupal
- drupal init --override
- drupal chain --file=~/.console/chain/quick-start.yml

NOTA: Esto nos va a generar un directorio llamado 'drupal8.dev'

Ahora procedemos a crear nuestro sitio, para lo cual creamos un nuevo directorio que será donde tendremos todos los archivos necesarios para Drupal.

El Site debe tener un directorio que nosotros queramos en mi caso en /var/www/drupal/ y creamos archivo de configuración para el drupal principal.

- a2ensite back-end.com.conf <sup>1</sup>
- apt-get install php5-gd php5-curl libssh2-php

Posterior a esto modificamos el archivo php.ini <sup>2</sup> cambiando un par de líneas haciendo que queden de esta manera.

- nano /etc/php5/apache2/php.ini

NOTA: La modificación será buscar las siguientes dos líneas y ponerlas tal cual están aquí.

```
////////////////////////////////
```

```
expose_php = off
```

```
...
```

```
allow_url_fopen = off
```

```
...
```

```
////////////////////////////////
```

Ahora agregamos el siguiente modulo

- a2enmod rewrite

Ahora nos posicionamos en el directorio donde tenemos el directorio 'drupal8.dev' que se descargó en pasos anteriores y ejecutamos el siguiente comando (debemos tener instalado rsync si no lo descargamos con apt-get install rsync):

- rsync -avz . /var/www/drupal

(Ojo, apuntamos a nuestro directorio que pusimos en el virtualhost creado con anterioridad)

- Ahora nos movemos al directorio '/var/www/drupal' y realizamos los siguientes pasos.
- mkdir /var/www/drupal/sites/default/files
- cp /var/www/drupal/sites/default/default.settings.php /var/www/drupal/sites/default/settings.php
- chmod 664 /var/www/drupal/sites/default/settings.php
- chown -R :www-data /var/www/drupal/\*
- chmod 777 /var/www/drupal/sites/default/files
- chown -R :www-data /var/www/drupal/\*

Una vez realizado esto tecleamos estando dentro del directorio '/var/www/drupal':

- drupal site:install

Seguimos todos los pasos que nos piden dando los datos necesarios como el host donde esta PostgreSQL, el nombre de la base, el nombre del usuario, su contraseña y demás cosas necesarias.

### Instalación sitios

Para la instalación de los sitios es necesario ingresar al directorio /var/www/drupal/sites y en este crear un archivo con nombre settings.php<sup>3</sup>

Una vez hecho eso, creamos dos directorios en /var/www/drupal/sites

- mkdir site1
- mkdir site2

Una vez creados copiados los archivos default.settings.php y default.services.yml que están en la ruta /var/www/drupal/sites/default en los nuevos directorios creados, renombrándolos como settings.php y services.yml

- cp /var/www/drupal/sites/default/default.services.yml /var/www/drupal/sites/site1/services.yml
- cp /var/www/drupal/sites/default/default.services.yml /var/www/drupal/sites/site2/services.yml
- cp /var/www/drupal/sites/default/default.settings.php /var/www/drupal/sites/site1/settings.php
- cp /var/wr/drupal/sites/default/default.settings.php /var/www/drupal/sites/site2/settings.php

Ahora modificamos nuestro archivo 000-default.conf <sup>4</sup> y agregamos los directorios donde estarán los sites.

Dado esto, quitamos el Site creado con anterioridad.

- a2dissite back-end.com.conf <sup>1</sup>

Por ultimo reiniciamos el servicio de apache.

- service apache2 restart

Ahora creamos el directorio 'files' dentro de cada sitio y le cambiamos los permisos a este. De igual manera cambiamos los permisos al archivo settings.php de cada uno de los sitios.

- mkdir /var/www/drupal/site/site1/files
- mkdir /var/www/drupal/site/site2/files
- chmod 777 /var/www/drupal/site/site1/files
- chmod 777 /var/www/drupal/site/site2/files
- chmod 777 /var/www/drupal/site/site1/settings.php
- chmod 777 /var/www/drupal/site/site2/settings.php

Ahora podemos ingresar a los sitios desde nuestro navegador y seguir el wizard de instalación de cada uno de los sitios, dando todos los datos solicitados (es necesario que para cada sitio se cree una base de datos diferente).

### Instalación PostgreSQL

En un una maquina nuevo con Debian 8 hacemos la instalación de PostgreSQL.

- apt-get install postgresql-9.4 postgresql-client-9.4

Ahora nos cambiamos al usuario postgres

- su postgres

Hecho esto ejecutamos nuestro script <sup>5</sup>

- psql -f script.sql

Ahora modificamos el archivo pg\_hba.conf <sup>6</sup> y postgresql.conf <sup>7</sup>



- nano /etc/postgresql/9.4/main/postgresql.conf
- nano /etc/postgresql/9.4/main/ pg\_hba.conf

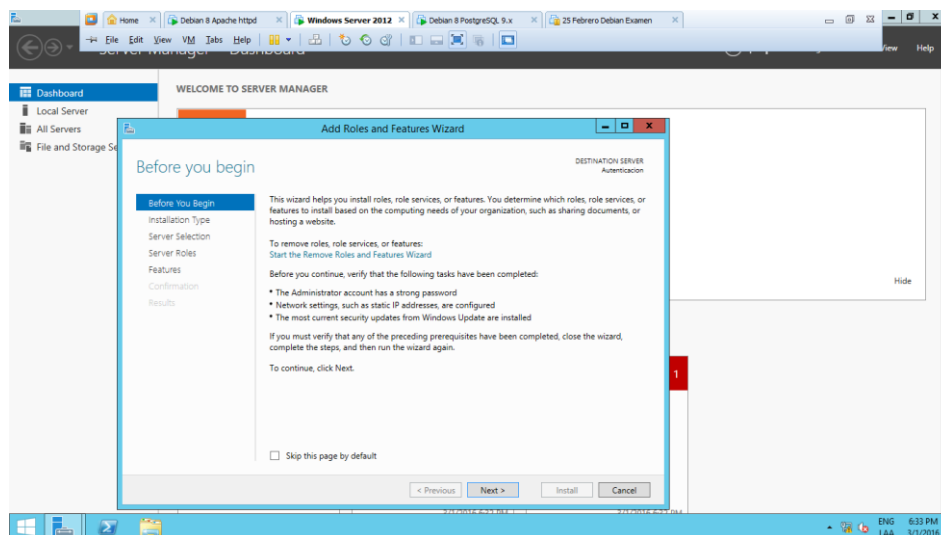
Por ultimo reiniciamos el servicio de postgresql

- service postgresql restart

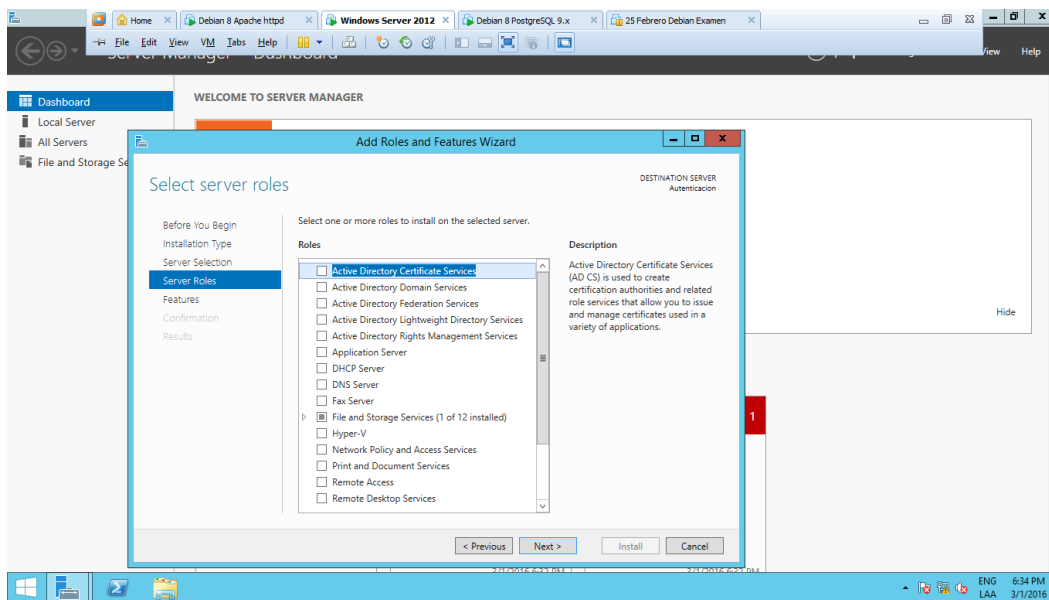
### Instalación Active Directory

Para la instalación de Active Directory lo primero que se debe realizar es ponerse una ip estática, así como cambiar el nombre de nuestra máquina.

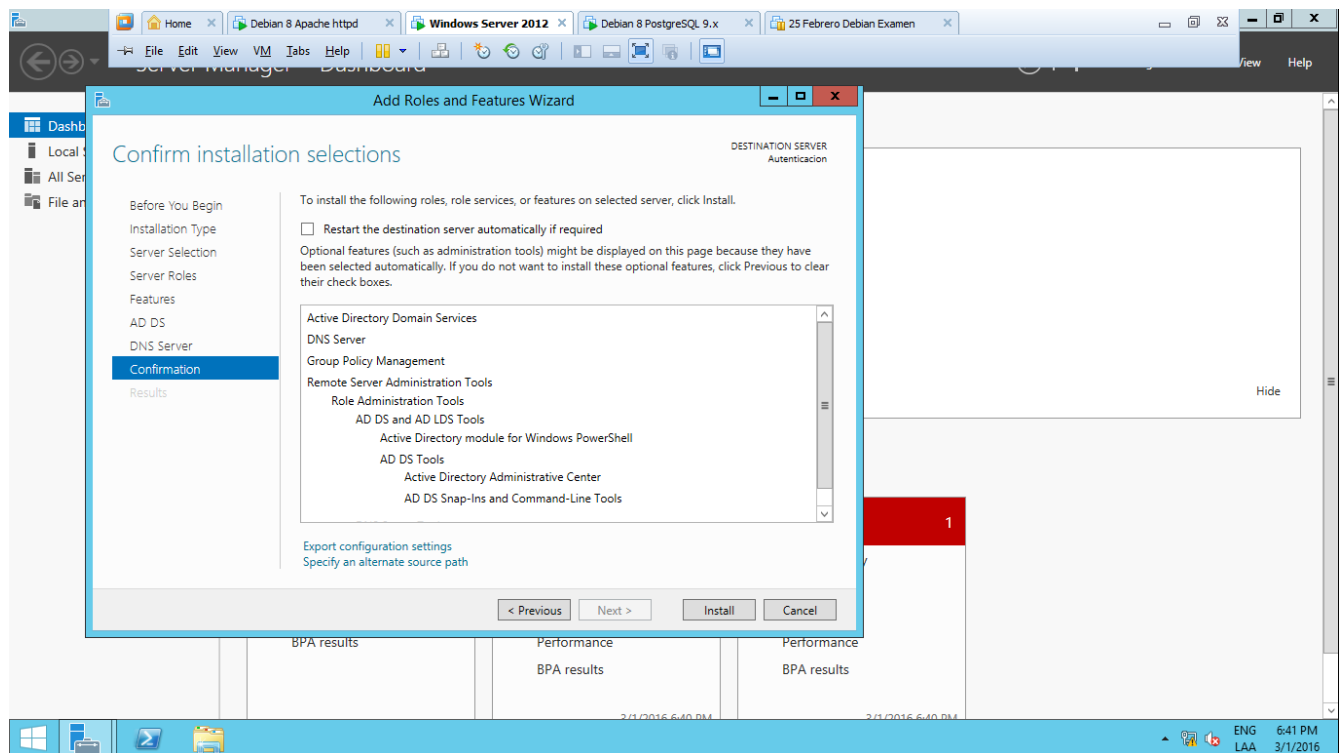
Posterior a esto agregamos un nuevo rol.



Seguimos el wizard de instalación y agregamos los roles de Active Directory Domain Services y DNS Server.



Continuamos con el wizar de instalacion hasta que nos aparezca que podemos comenzar la instalacion y damos click al boton Install.



Una vez instalado, nos pedirá una configuración para lo cual solo es necesario agregar un Nuevo Forest, al cual le dimos el nombre de active.local, dado esto nos pedirá una contraseña y lo demás solo es dar clic en Next hasta que termine la instalación.

## Front End

### Drupal Console

Drupal console es un conjunto de herramientas que se ejecutan en una línea de comandos para interactuar con una instalación de drupal 8.

#### Paquetes que puede ser necesarios de instalar antes

```
apt-get install php5-curl
apt-get install php5-gd
apt-get install php5-sqlite
```

Descargamos drupal console, lo hacemos en /usr/bin para poder ejecutarlo usando únicamente drupal

```
root@frontend:/usr/bin# curl https://drupalconsole.com/installer -L -o drupal.phar
```

#### Damos permisos de ejecución

```
root@frontend:/usr/bin # chmod +x drupal.phar
root@frontend:/usr/bin # mv drupal.phar drupal
```

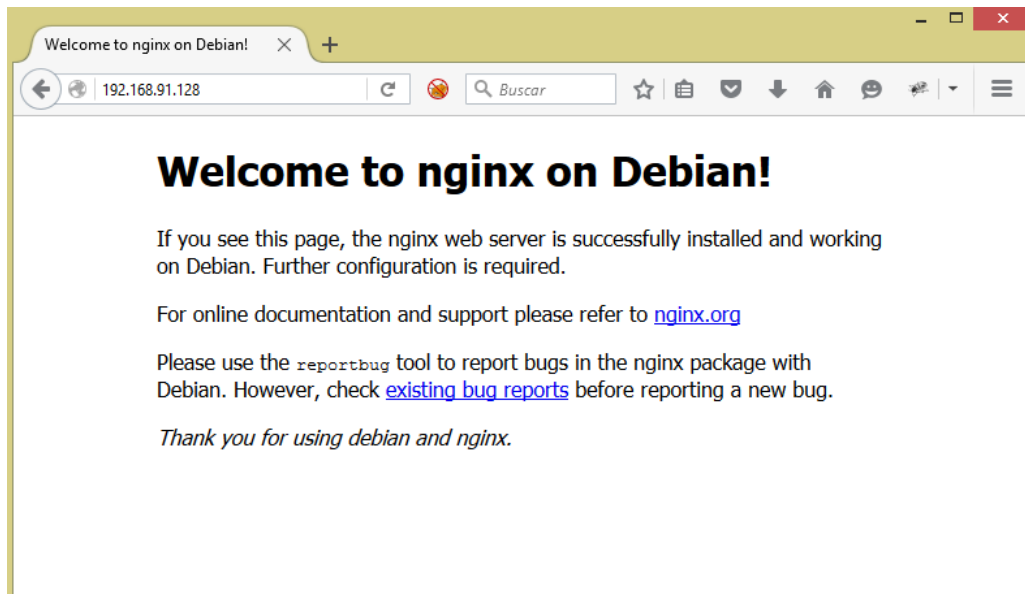
### Servidor Nginx

Ya que Nginx no soporta de manera nativa PHP para instalar Nginx con soporte para PHP es necesario instalar el módulo FastCGI de php5

#### Paquetes de instalación de Nginx y php5 FastCGI

```
apt-get install nginx
apt-get install php5-fpm
apt-get install php5
```

Ingresamos al localhost en un navegador web para comprobar que nginx se instaló correctamente.



## Creando el sitio para Drupal

Primero debemos crear los directorios y archivos necesarios para el sitio de drupal

### Creando el directorio para drupal

```
root@frontend:~# mkdir /opt/drupal8
```

### Creando el archivo de configuración para drupal

```
root@frontend:/etc/nginx/sites-available# touch front-end.com
```

### Creamos la liga simbólica con el directorio sites-enabled

```
ln -s /etc/nginx/sites-available/front-end.com /etc/nginx/sites-enabled/front-end.com
```

### Creamos la entrada por nombre de dominio para en el archivo /etc/hosts

```
192.168.1.6 front-end.com
```

Dentro del archivo de configuración front-end.com van las siguientes líneas de código. Esta configuración está basada por el modelo proporcionado por Nginx (Nginx Inc., s.f.)

### Creando el directorio para drupal

```
root@frontend:~# mkdir /opt/drupal8
```

### Creando el archivo de configuración para drupal

```
root@frontend:/etc/nginx/sites-available# touch front-end.com
```

### Creamos la liga simbólica con el directorio sites-enabled

```
ln -s /etc/nginx/sites-available/front-end.com /etc/nginx/sites-enabled/front-end.com
```

## Activar FastCGI un Nginx

Dentro del archivo de configuración del sitio debemos agregar las siguientes líneas

### Activar FastCGI en Nginx

```
location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    # With php5-fpm:
    fastcgi_pass unix:/var/run/php5-fpm.sock;
}
```

<https://ftp.drupal.org/files/projects/drupal-8.0.x-dev.tar.gz>

```
postgres=# CREATE USER drupalAdmin;
```

```
CREATE ROLE
```

```
postgres=# ALTER USER drupalAdmin with password 'ola123';
```

```
CREATE DATABASE drupal8 OWNER drupalAdmin;
```

```
192.168.91.129 - PuTTY
root@drupal:/opt/drupal8# service nginx restart
root@drupal:/opt/drupal8# su postgres
postgres@drupal:/opt/drupal8$ psql
postgres@drupal:/opt/drupal8$ psql
psql (9.4.6)
Type "help" for help.

postgres=# CREATE USER drupalAdmin;
CREATE ROLE
postgres=# ALTER USER drupalAdmin with password 'ola123';
ALTER ROLE
postgres=# CREATE DATABASE drupal8.0.3 OWNER drupalAdmin;
ERROR:  syntax error at or near ".0"
LINE 1: CREATE DATABASE drupal8.0.3 OWNER drupalAdmin;
                             ^
postgres=# CREATE DATABASE drupal8 OWNER drupalAdmin;
CREATE DATABASE
postgres=# \q
postgres@drupal:/opt/drupal8$ clear
postgres@drupal:/opt/drupal8$
```

```
192.168.91.129 - PuTTY
GNU nano 2.2.6 File: /etc/postgresql/9.4/main/pg_hba.conf

# DO NOT DISABLE!
# If you change this first entry you will need to make sure that the
# database superuser can access the database using some other method.
# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket
local all postgres md5

# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all md5
# IPv4 local connections:
host all all 127.0.0.1/32 md5
# IPv6 local connections:
host all all ::1/128 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
#local replication postgres peer
#host replication postgres 127.0.0.1/32 md5
#host replication postgres ::1/128 md5

[ Read 99 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

## Instalar Drupal en Nginx

Primero instalamos los paquetes necesarios para poder instalar la drupal con una base de datos postgres.

Paquetes que puede ser necesarios de instalar antes

```
apt-get install postgres postgres-client
```

```
apt-get install php5-pgsql
```

Utilizaremos drupal console para instalar la última versión de drupal disponible

Creamos un nuevo sitio y elegimos la última versión [0]

```
root@frontend:/opt/drupal8#drupal site: new
```

```
192.168.91.129 - PuTTY
root@drupal:/opt/drupal8# drupal site:new

Enter the directory name when downloading Drupal:
> .

Getting releases for Drupal

Select a core release:
[0 ] 8.0.4
[1 ] 8.0.3
[2 ] 8.0.2
[3 ] 8.0.1
[4 ] 8.0.0
[5 ] 8.0.0-rc4
```

Llenamos los datos de la base de datos [0] postgres el host la base de datos el usuario, contraseña y el puerto

```
192.168.91.129 - PuTTY
root@drupal:/opt/drupal8# drupal site:install

Select Drupal profile to be installed:
[0] Minimal
[1] Standard
> 1

Select language for your Drupal installation [English]:
>

Drupal Database type:
[0] PostgreSQL
> 0

Database Host [127.0.0.1]:
>

Database Name:
> drupal8

Database User:
> drupaladmin

Database Pass [ ]:
>

Database Port [3306]:
> 5432

Database Prefix [ ]:
>

Provide your Drupal site name [Drupal 8 Site Install]:
> Front End

Provide your Drupal site mail [admin@example.com]:
> root@localhost

Provide your Drupal administrator account name [admin]:
> Admin

Provide your Drupal administrator account mail [root@localhost]:
>
```

Nos dirá que el sitio ha sido creado

```
192.168.91.129 - PuTTY

Database Port [3306]:
> 5432

Database Prefix [ ]:
>

Provide your Drupal site name [Drupal 8 Site Install]:
> Front End

Provide your Drupal site mail [admin@example.com]:
> root@localhost

Provide your Drupal administrator account name [admin]:
> Admin

Provide your Drupal administrator account mail [root@localhost]:
>

Provide your Drupal administrator account password:
>

Starting Drupal 8 install process

[OK] Your Drupal 8 installation was completed successfully

// settings:check

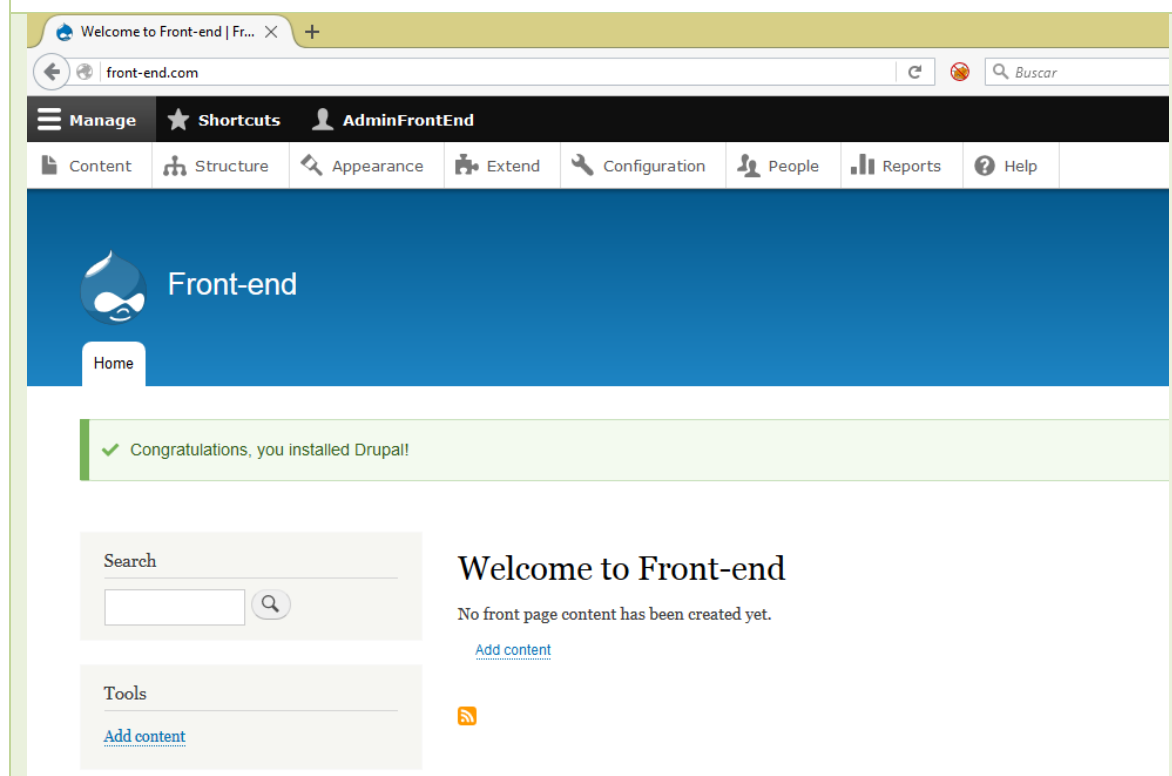
The extension mysql is recommended to install.

The extension sqlite3 is recommended to install.

The configuration date.timezone was missing and overwritten with America/Tijuana.

root@drupal:/opt/drupal8#
```

Vamos a la ruta y mostrara el contenido del front-end ya instalado



Conexión cifrada para el front-end

Creamos un directorio para contener los certificados de Nginx

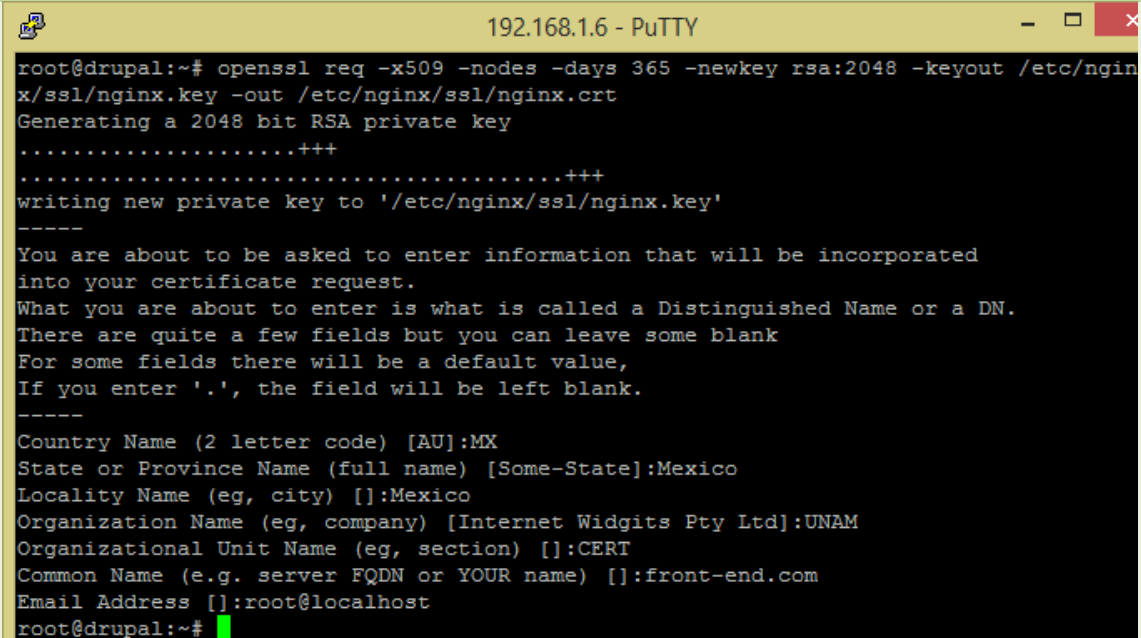
```
root@drupal:/etc/nginx# mkdir ssl
```

Creamos los certificados con openssl, crearemos un certificado de 2048 bits, con el estándar x509 y valido por 1 año

```
root@drupal:~# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/nginx/ssl/nginx.key -out /etc/nginx/ssl/nginx.crt
```

```
root@drupal:~# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/nginx/ssl/nginx.key -out /etc/nginx/ssl/nginx.crt
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to '/etc/nginx/ssl/nginx.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
```

Llenamos los datos de información, el valor FQDN que introducimos será front-end.com



```
192.168.1.6 - PuTTY
root@drupal:~# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/nginx/ssl/nginx.key -out /etc/nginx/ssl/nginx.crt
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to '/etc/nginx/ssl/nginx.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:MX
State or Province Name (full name) [Some-State]:Mexico
Locality Name (eg, city) []:Mexico
Organization Name (eg, company) [Internet Widgits Pty Ltd]:UNAM
Organizational Unit Name (eg, section) []:CERT
Common Name (e.g. server FQDN or YOUR name) []:front-end.com
Email Address []:root@localhost
root@drupal:~#
```

Configuramos el archivo de configuración de Nginx /etc/nginx/sites-available/front-end.com

Agregamos las siguientes líneas

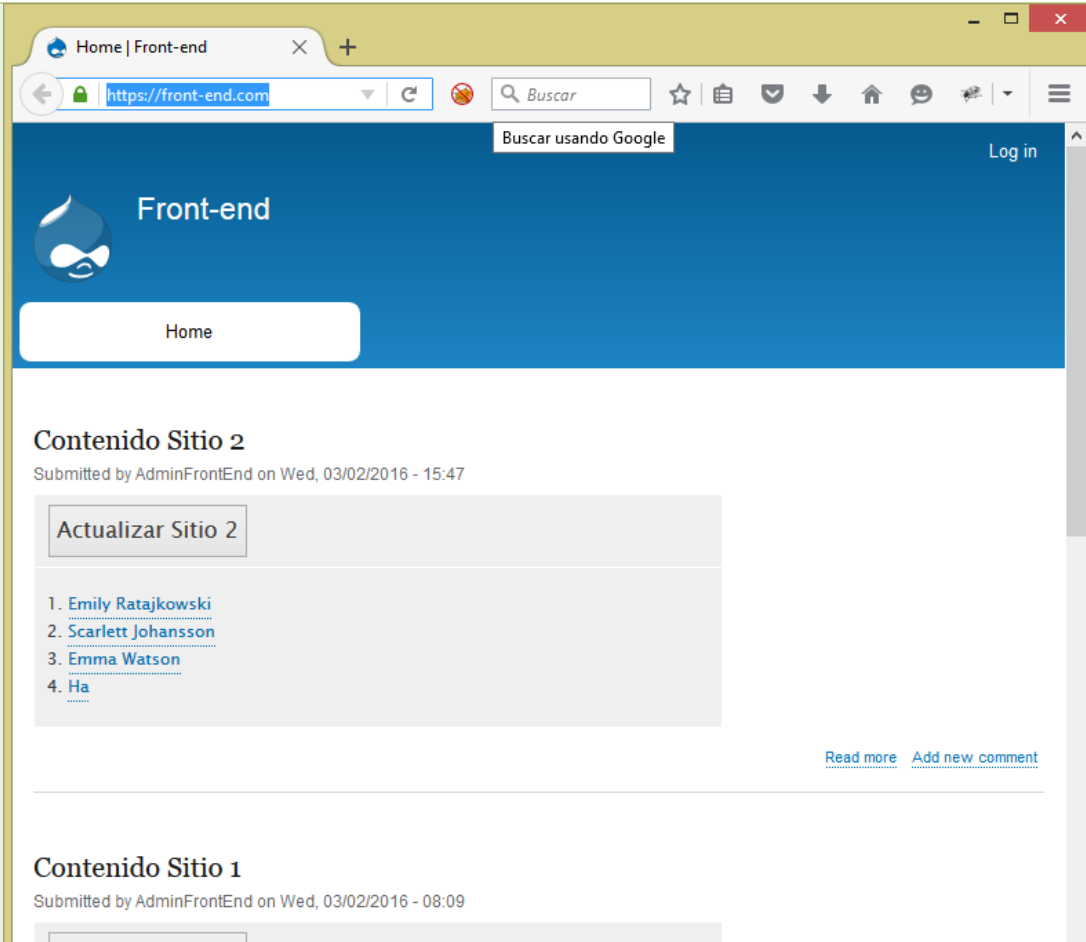


```
...
listen 443 ssl;
ssl_certificate /etc/nginx/ssl/nginx.crt;
ssl_certificate_key /etc/nginx/ssl/nginx.key;
...
```

Reiniciamos el servicio Nginx

```
root@drupal:~# service nginx restart
```

Prueba de la conexión



### Modulo Front End

En esta parte se muestra como instalar el modulo para el front end

Vamos a utilizar el archivo de Nginx vamos a editar el archivo `/etc/nginx/sites-available/default` y agregar las siguientes líneas para agregar un nuevo dominio con soporte de php5

```
...
server_name get-data.com;
...
location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    fastcgi_pass unix:/var/run/php5-fpm.sock;
}
```

```
}
```

```
...
```

Creamos la entrada por nombre de dominio para en el archivo /etc/hosts

```
192.168.1.6 get-data.com
```

Copiamos los scripts getSitio1.php y getSitio2.php en la carpeta

```
/opt/drupal8/scripts
```

Copiamos el archivo index.php en la carpeta

```
/var/www/html
```

Copiamos el archivo formulario\_style.css en la carpeta

```
/var/www/html/style
```

Y finalizamos cambiando el dueño de los directorios por www-data

```
root@drupal:/etc/nginx# chown www-data:www-data -R /var/www/html
```

```
root@drupal:/etc/nginx# chown www-data:www-data -R /opt/drupal8
```

## Nginx ModSecurity

### ModSecurity

Se instalaran todos los paquetes necesarios para compilar Nginx y ModSecurity.

- apt-get install git build-essential libpcre3 libpcre3-dev libssl-dev libtool autoconf apache2-prefork-dev libxml2-dev libcurl4-openssl-dev

```
root@debian:~# apt-get install git build-essential libpcre3 libpcre3-dev libssl-dev libtool autoconf apache2-prefork-dev libxml2-dev libcurl4-openssl-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'apache2-dev' instead of 'apache2-prefork-dev'
apache2-dev is already the newest version.
autoconf is already the newest version.
build-essential is already the newest version.
git is already the newest version.
libtool is already the newest version.
libxml2-dev is already the newest version.
libpcre3 is already the newest version.
libpcre3-dev is already the newest version.
libcurl4-openssl-dev is already the newest version.
libssl-dev is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
root@debian:~#
```

Nos movemos al directorio **cd /usr/src** y allí clonaremos el siguiente repositorio.

- git clone https://github.com/SpiderLabs/ModSecurity.git modsecurity

```
root@debian:~# cd /usr/src/
root@debian:/usr/src# git clone https://github.com/SpiderLabs/ModSecurity.git modsecurity
Cloning into 'modsecurity'...
remote: Counting objects: 19137, done.
remote: Total 19137 (delta 0), reused 0 (delta 0), pack-reused 19137
Receiving objects: 100% (19137/19137), 36.72 MiB | 577.00 KiB/s, done.
Resolving deltas: 100% (12814/12814), done.
Checking connectivity... done.
root@debian:/usr/src#
```

Una vez realizado esto, vamos a descargar Nginx con el comando `wget` usaremos la versión 1.8 con el comando:

- `wget http://nginx.org/download/nginx-1.8.0.tar.gz`

```
root@debian:/usr/src# wget http://nginx.org/download/nginx-1.8.0.tar.gz
--2016-03-01 06:46:04-- http://nginx.org/download/nginx-1.8.0.tar.gz
Resolving nginx.org (nginx.org)... 95.211.80.227, 206.251.255.63
Connecting to nginx.org (nginx.org)|95.211.80.227|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 832104 (813K) [application/octet-stream]
Saving to: 'nginx-1.8.0.tar.gz'

nginx-1.8.0.tar.g 100%[=====>] 812.60K  436KB/s  in 1.9s

2016-03-01 06:46:22 (436 KB/s) - 'nginx-1.8.0.tar.gz' saved [832104/832104]

root@debian:/usr/src#
```

Y se descomprime con:

- `tar -zxvf nginx-1.8.0.tar.gz`

```
root@debian:/usr/src# ls
modsecurity nginx-1.8.0 nginx-1.8.0.tar.gz
root@debian:/usr/src# tar -zxvf nginx-1.8.0.tar.gz
```

Vamos al directorio `cd /usr/src/modsecurity` dentro de allí vamos a compilar el modulo independiente en el servidor, por lo que podemos incluirlo a Nginx.

- `./autogen.sh`
- `./configure --enable-standalone-module --disable-mlogc`
- `make`

```
root@debian:/usr/src/modsecurity# ./autogen.sh
libtoolize: putting auxiliary files in AC_CONFIG_AUX_DIR, 'build'.
libtoolize: copying file 'build/ltmain.sh'
libtoolize: putting macros in AC_CONFIG_MACRO_DIR, 'build'.
libtoolize: copying file 'build/libtool.m4'
libtoolize: copying file 'build/ltoptions.m4'
libtoolize: copying file 'build/ltugar.m4'
libtoolize: copying file 'build/ltversion.m4'
libtoolize: copying file 'build/ltobsolete.m4'
configure.ac:704: warning: PKG_PROG_PKG_CONFIG is m4_require'd but not m4_defun'd
build/find_lua.m4:7: CHECK_LUA is expanded from...
configure.ac:704: the top level
configure.ac:710: warning: PKG_PROG_PKG_CONFIG is m4_require'd but not m4_defun'd
build/find_yajl.m4:9: CHECK_YAJL is expanded from...
configure.ac:710: the top level
```

```
root@debian:/usr/src/modsecurity# ./configure --enable-standalone-module --disable-mlogc
```

```
root@debian:/usr/src/modsecurity# make
```

Ahora nos situamos en directorio nginx `cd ../nginx-1.8.0` para compilar e incluir el módulo de ModSecurity.

```
root@debian:/usr/src/modsecurity# cd ../nginx-1.8.0/
root@debian:/usr/src/nginx-1.8.0#
```

```
./configure \
--user=www-data \
--group=www-data \
--with-debug \
--with-ipv6 \
--with-http_ssl_module \
--add-module=/usr/src/modsecurity/nginx/modsecurity
```

```
root@debian:/usr/src/nginx-1.8.0# ./configure \
> --user=www-data \
> --group=www-data \
> --with-debug \
> --with-ipv6 \
> --with-http_ssl_module \
> --add-module=/usr/src/modsecurity/nginx/modsecurity
```

*Nota: Nginx se ejecutará con el usuario y el grupo " www -data", y activar los módulos de depuración , IPv6 y SSL . Y, finalmente, se incluye el módulo de ModSecurity en Nginx.*

Ahora instalaremos Nginx

- `make`
- `make install`

```
root@debian:/usr/src/nginx-1.8.0# make
```

```
root@debian:/usr/src/nginx-1.8.0# make install
```

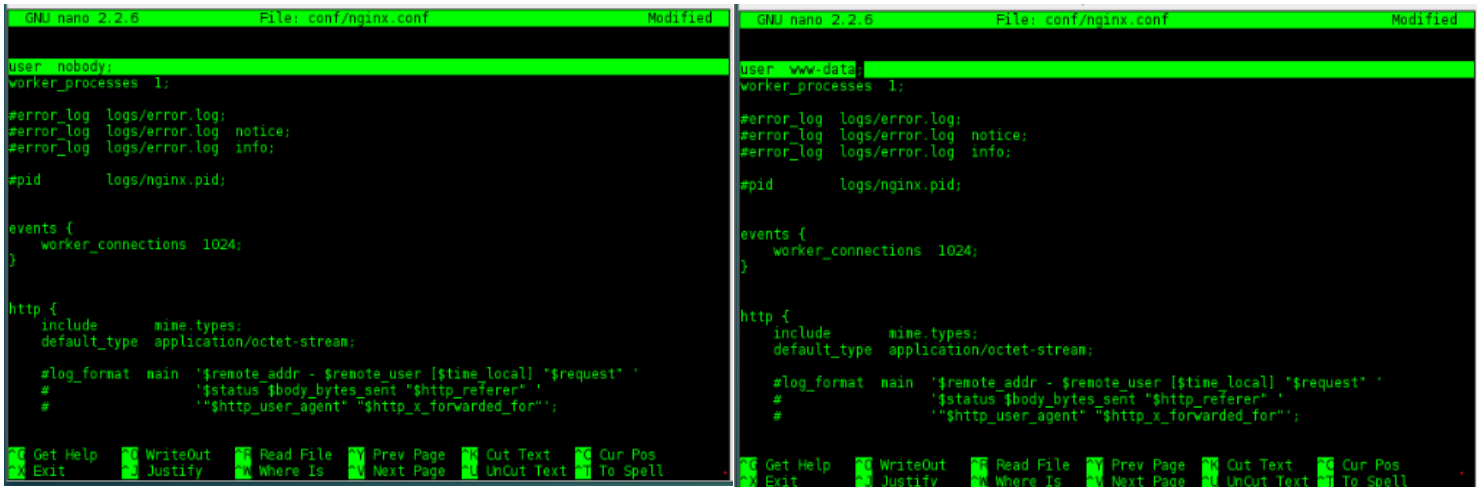
Cuando el comando `make install` está terminado, se puede ver que Nginx se instala en el directorio "`/usr / local / nginx`"

```
root@debian:~# cd /usr/local/nginx/
root@debian:/usr/local/nginx# ls -l
total 16
drwxr-sr-x 2 root staff 4096 Mar  1 22:33 conf
drwxr-sr-x 2 root staff 4096 Mar  1 22:33 html
drwxr-sr-x 2 root staff 4096 Mar  1 22:33 logs
drwxr-sr-x 2 root staff 4096 Mar  1 22:33 sbin
root@debian:/usr/local/nginx#
```

Ahora vamos al directorio **cd /usr/local/nginx/conf** y vamos a editar el archivo *nginx.conf*.

```
root@debian:/usr/local/nginx# cd /usr/local/nginx/  
root@debian:/usr/local/nginx# nano conf/nginx.conf
```

Cambiaremos la primera línea de *user nobody* → *user www-data* guardamos y salimos



The image shows two side-by-side screenshots of the nano text editor. The left screenshot shows the original configuration with the line `user nobody;` highlighted in green. The right screenshot shows the same file after editing, with the line `user www-data;` highlighted in green. Both screenshots show the full content of the `nginx.conf` file, including worker processes, error logs, pid file, events, and http settings.

Crearemos un enlace simbólico para el binario nginx para que podamos sacar el comando “nginx” directamente

- `ln -s /usr/local/nginx/sbin/nginx /bin/nginx`

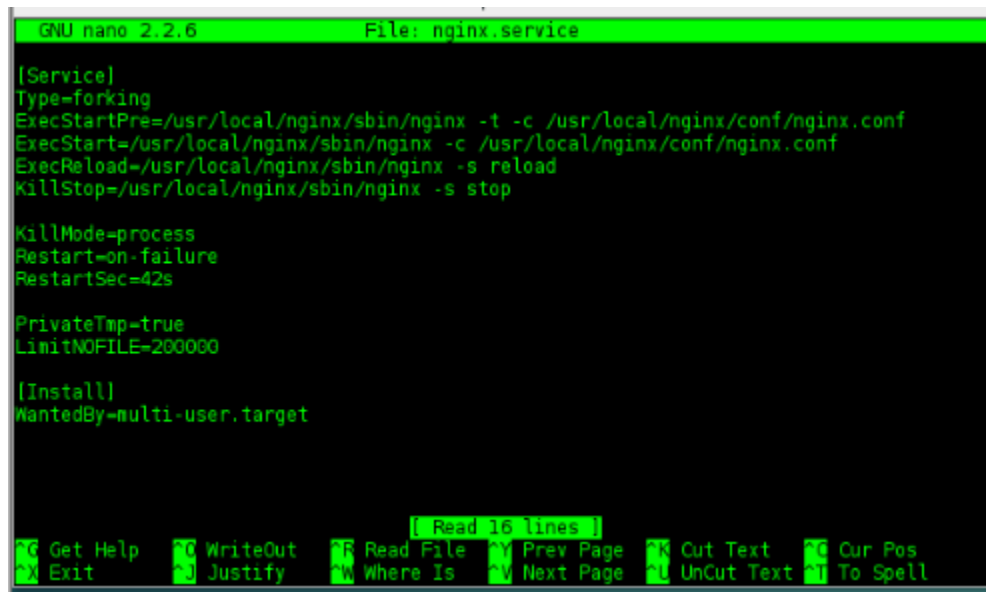
```
root@debian:/usr/local/nginx# ln -s /usr/local/nginx/sbin/nginx /bin/nginx
```

El siguiente paso es cambiarnos de directorio a **cd /lib/systemd/system/** y dentro de allí editar el archivo `nginx.service` (agregar lo siguiente al código) cuando se haya hecho guardar y salir.

```
root@debian:/lib/systemd/system# nano nginx.service
```

```
[Service]  
Type=forking  
ExecStartPre=/usr/local/nginx/sbin/nginx -t -c /usr/local/nginx/conf/nginx.conf  
ExecStart=/usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf  
ExecReload=/usr/local/nginx/sbin/nginx -s reload  
KillStop=/usr/local/nginx/sbin/nginx -s stop  
  
KillMode=process  
Restart=on-failure  
RestartSec=42s  
  
PrivateTmp=true  
LimitNOFILE=200000
```

```
[Install]
WantedBy=multi-user.target
```



```
GNU nano 2.2.6      File: nginx.service

[Service]
Type=forking
ExecStartPre=/usr/local/nginx/sbin/nginx -t -c /usr/local/nginx/conf/nginx.conf
ExecStart=/usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf
ExecReload=/usr/local/nginx/sbin/nginx -s reload
KillStop=/usr/local/nginx/sbin/nginx -s stop

KillMode=process
Restart=on-failure
RestartSec=42s

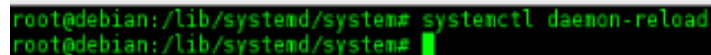
PrivateTmp=true
LimitNOFILE=200000

[Install]
WantedBy=multi-user.target

[ Read 16 lines ]
Get Help  WriteOut  Read File  Prev Page  Cut Text  Cur Pos
Exit      Justify    Where Is  Next Page  UnCut Text To Spell
```

Ahora recargaremos systemd-daemon para que el systemd cargue nuestro archivo de servicio NGINX.

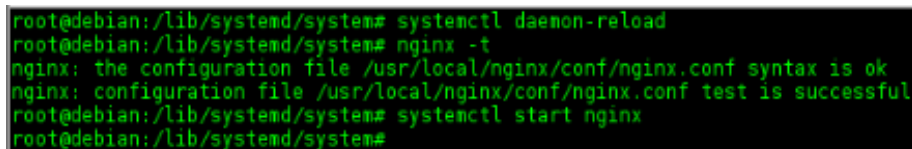
- `systemctl daemon-reload`



```
root@debian:/lib/systemd/system# systemctl daemon-reload
root@debian:/lib/systemd/system#
```

Se verificara la configuración de Nginx y se reiniciar el servicio

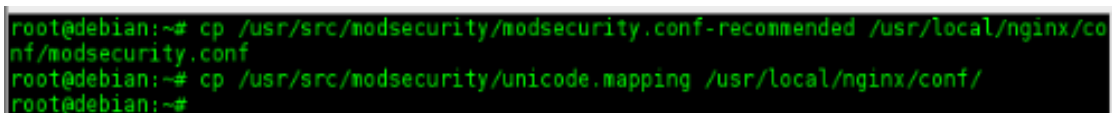
- `nginx -t`
- `systemctl start nginx`



```
root@debian:/lib/systemd/system# systemctl daemon-reload
root@debian:/lib/systemd/system# nginx -t
nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok
nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful
root@debian:/lib/systemd/system# systemctl start nginx
root@debian:/lib/systemd/system#
```

Copiaremos el archivo de configuración de ModSecurity al directorio Nginx con el nombre de "modsecurity.conf"

- `cp /usr/src/modsecurity/modsecurity.conf-recommended /usr/local/nginx/conf/modsecurity.conf`
- `cp /usr/src/modsecurity/unicode.mapping /usr/local/nginx/conf/`



```
root@debian:~# cp /usr/src/modsecurity/modsecurity.conf-recommended /usr/local/nginx/conf/modsecurity.conf
root@debian:~# cp /usr/src/modsecurity/unicode.mapping /usr/local/nginx/conf/
root@debian:~#
```

Cambiamos al directorio **cd /usr/local/nginx/conf** y editamos el archivo **modsecurity.conf** en las siguientes líneas:

```
root@debian:~# cd /usr/local/nginx/conf/
root@debian:/usr/local/nginx/conf# nano modsecurity.conf
```

Línea 7 cambiamos “Detection Only” → “Detection On”

Línea 38 aumentamos el valor a: **SecRequestBodyLimit 13107200** → **SecRequestBodyLimit 100000000**

Línea 192 cambiamos el valor de: **SecAuditLogType Serial** → **SecAuditLogTypeSerial Concurrent**

Línea 193 la comentamos

**SecAuditLog /var/log/modsec\_audit.log** → **# SecAuditLog /var/log/modsec\_audit.log**

Línea 196 se descomenta la línea

**#SecAuditLogStorageDir /opt/modsecurity/var/audit/** → **SecAuditLogStorageDir /opt/modsecurity/var/audit/**

Guardamos y salimos

```
GNU nano 2.2.6 File: modsecurity.conf
# -- Rule engine initialization -----
# Enable ModSecurity, attaching it to every transaction. Use detection
# only to start with, because that minimises the chances of post-installation
# disruption.
#
SecRuleEngine DetectionOnly
#
# -- Request body handling -----
# Allow ModSecurity to access request bodies. If you don't, ModSecurity
# won't be able to see any POST parameters, which opens a large security
# hole for attackers to exploit.
#
SecRequestBodyAccess On
#
# Enable XML request body parser.
# Initiate XML Processor in case of xml content-type
#
# Get Help  WriteOut  Read File  Prev Page  Cut Text  Cur Pos
# Exit      Justify   Where Is  Next Page  UnCut Text  To Spell
```

```
GNU nano 2.2.6 File: modsecurity.conf Modified
# -- Rule engine initialization -----
# Enable ModSecurity, attaching it to every transaction. Use detection
# only to start with, because that minimises the chances of post-installation
# disruption.
#
SecRuleEngine On
#
# -- Request body handling -----
# Allow ModSecurity to access request bodies. If you don't, ModSecurity
# won't be able to see any POST parameters, which opens a large security
# hole for attackers to exploit.
#
SecRequestBodyAccess On
#
# Enable XML request body parser.
# Initiate XML Processor in case of xml content-type
#
# Get Help  WriteOut  Read File  Prev Page  Cut Text  Cur Pos
# Exit      Justify   Where Is  Next Page  UnCut Text  To Spell
```

```
GNU nano 2.2.6 File: modsecurity.conf Modified
#id:'200000',phase:1,t:none,t:lowercase,pass,nolog,ctl:requestBodyProcessor=XML"
# Enable JSON request body parser.
# Initiate JSON Processor in case of JSON content-type; change accordingly
# if your application does not use 'application/json'
#
SecRule REQUEST_HEADERS:Content-Type "application/json" \
#id:'200001',phase:1,t:none,t:lowercase,pass,nolog,ctl:requestBodyProcessor=JSON"
#
# Maximum request body size we will accept for buffering. If you support
# file uploads then the value given on the first line has to be as large
# as the largest file you are willing to accept. The second value refers
# to the size of data, with files excluded. You want to keep that value as
# low as practical.
#
SecRequestBodyLimit 13107200
SecRequestBodyNoFilesLimit 131072
#
# Store up to 128 KB of request body data in memory. When the multipart
# parser reaches this limit, it will start using your hard disk for
#
# line 38/227 (16%), col 17/29 (58%), char 1436/8417 (17%)
# Get Help  WriteOut  Read File  Prev Page  Cut Text  Cur Pos
# Exit      Justify   Where Is  Next Page  UnCut Text  To Spell
```

```
GNU nano 2.2.6 File: modsecurity.conf Modified
#id:'200000',phase:1,t:none,t:lowercase,pass,nolog,ctl:requestBodyProcessor=XML"
# Enable JSON request body parser.
# Initiate JSON Processor in case of JSON content-type; change accordingly
# if your application does not use 'application/json'
#
SecRule REQUEST_HEADERS:Content-Type "application/json" \
#id:'200001',phase:1,t:none,t:lowercase,pass,nolog,ctl:requestBodyProcessor=JSON"
#
# Maximum request body size we will accept for buffering. If you support
# file uploads then the value given on the first line has to be as large
# as the largest file you are willing to accept. The second value refers
# to the size of data, with files excluded. You want to keep that value as
# low as practical.
#
SecRequestBodyLimit 100000000
SecRequestBodyNoFilesLimit 131072
#
# Store up to 128 KB of request body data in memory. When the multipart
# parser reaches this limit, it will start using your hard disk for
#
# line 38/227 (16%), col 17/29 (58%), char 1436/8417 (17%)
# Get Help  WriteOut  Read File  Prev Page  Cut Text  Cur Pos
# Exit      Justify   Where Is  Next Page  UnCut Text  To Spell
```

```
GNU nano 2.2.6 File: modsecurity.conf Modified
# Use a single file for logging. This is much easier to look at, but
# assumes that you will use the audit log only occasionally.
#
SecAuditLogType Serial
SecAuditLog /var/log/modsec_audit.log

# Specify the path for concurrent audit logging.
#SecAuditLogStorageDir /opt/modsecurity/var/audit/

# -- Miscellaneous -----
# Use the most commonly used application/x-www-form-urlencoded parameter
# separator. There's probably only one application somewhere that uses
# something else so don't expect to change this value.
#
SecArgumentSeparator &

# Settle on version 0 (zero) cookies, as that is what most applications
[ line 192/227 (84%), col 23/23 (100%), char 7138/8418 (84%) ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

```
GNU nano 2.2.6 File: modsecurity.conf Modified
# Use a single file for logging. This is much easier to look at, but
# assumes that you will use the audit log only occasionally.
#
SecAuditLogType Concurrent
SecAuditLog /var/log/modsec_audit.log

# Specify the path for concurrent audit logging.
#SecAuditLogStorageDir /opt/modsecurity/var/audit/

# -- Miscellaneous -----
# Use the most commonly used application/x-www-form-urlencoded parameter
# separator. There's probably only one application somewhere that uses
# something else so don't expect to change this value.
#
SecArgumentSeparator &

# Settle on version 0 (zero) cookies, as that is what most applications
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

```
GNU nano 2.2.6 File: modsecurity.conf Modified

# Use a single file for logging. This is much easier to look at, but
# assumes that you will use the audit log only occasionally.
#
SecAuditLogType Concurrent
SecAuditLog /var/log/modsec_audit.log

# Specify the path for concurrent audit logging.
SecAuditLogStorageDir /opt/modsecurity/var/audit/

# -- Miscellaneous -----
# Use the most commonly used application/x-www-form-urlencoded parameter
# separator. There's probably only one application somewhere that uses
# something else so don't expect to change this value.
#
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Ahora crearemos un nuevo directorio para el registro de Modsecurity y cambiar el propietario a www-data

- `mkdir -p /opt/modsecurity/var/audit/`
- `chown -R www-data:www-data /opt/modsecurity/var/audit/`

```
root@debian:/usr/local/nginx/conf# mkdir -p /opt/modsecurity/var/audit/
root@debian:/usr/local/nginx/conf# chown -R www-data:www-data /opt/modsecurity/var/audit/
root@debian:/usr/local/nginx/conf#
```



Nos cambiamos de directorio a **cd /usr/src** y clonamos el siguiente repositorio:

- **git clone** <https://github.com/SpiderLabs/owasp-modsecurity-crs.git>

```
root@debian:~# cd /usr/src/
root@debian:/usr/src# git clone https://github.com/SpiderLabs/owasp-modsecurity-crs.git
Cloning into 'owasp-modsecurity-crs'...
remote: Counting objects: 1603, done.
remote: Total 1603 (delta 0), reused 0 (delta 0), pack-reused 1602
Receiving objects: 100% (1603/1603), 11.48 MiB | 280.00 KiB/s, done.
Resolving deltas: 100% (1031/1031), done.
Checking connectivity... done.
root@debian:/usr/src#
```

Luego vamos al directorio **cd owasp-modsecurity-crs** y copiamos el directorio “base\_rules” al directorio **nginx**.

- **cp -R base\_rules/ /usr/local/nginx/conf/**

```
root@debian:/usr/src# cd owasp-modsecurity-crs/
root@debian:/usr/src/owasp-modsecurity-crs# cp -R base_rules/ /usr/local/nginx/conf/
root@debian:/usr/src/owasp-modsecurity-crs#
```

Editamos **modsecurity.conf** que está dentro del directorio **cd /usr/local/nginx/conf/** y agregamos OWASP CRS al final del archivo

```
#DefaultAction
SecDefaultAction "log,deny,phase:1"

#If you want to load single rule /usr/local/nginx/conf
#include base_rules/modsecurity_crs_41_sql_injection_attacks.conf

#Load all Rule
include base_rules/*.conf
```

```
GNU nano 2.2.6      File: modsecurity.conf      Modified
SecUnicodeMapFile unicode.mapping 20127

# Improve the quality of ModSecurity by sharing information about your
# current ModSecurity version and dependencies versions.
# The following information will be shared: ModSecurity version,
# Web Server version, APR version, PCRE version, Lua version, Libxml2
# version, Anonymous unique id for host.
SecStatusEngine On

#DefaultAction
SecDefaultAction "log,deny,phase.1"

#If you want to load single rule /usr/local/nginx/conf

#include base_rules/modsecurity_crs_41_sql_injection_attacks.conf
include base_rules/*.conf

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^N Next Page  ^L UnCut Text ^T To Spell
```

Ingresa a la ruta `cd /usr/local/nginx/conf` y agrega las siguientes líneas en el archivo `nginx.conf`.

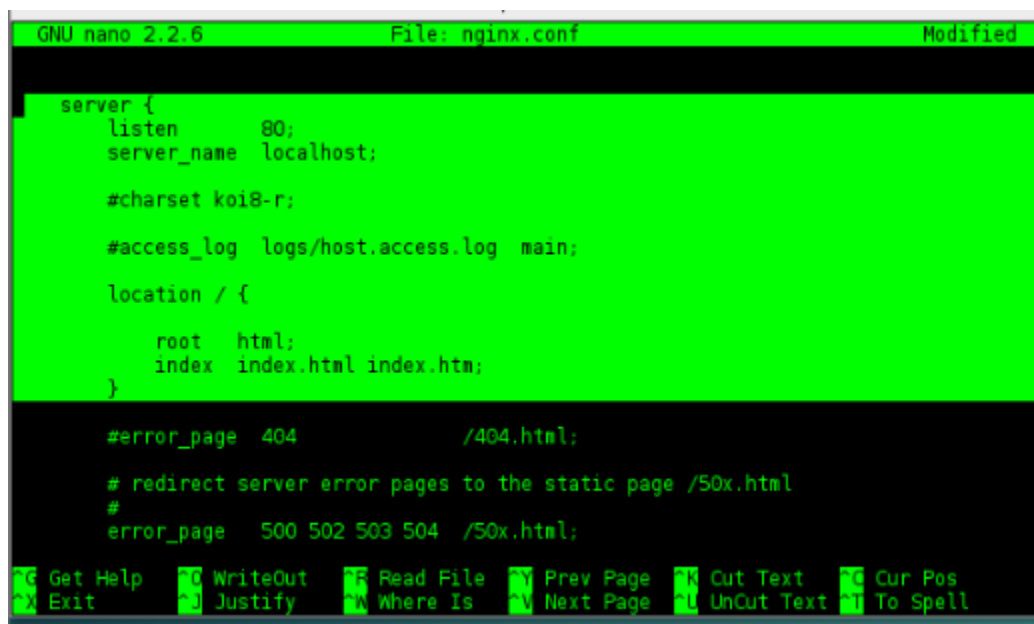
```
root@debian:/usr/local/nginx/conf# nano nginx.conf
```

[.....]

```
#Enable ModSecurity
ModSecurityEnabled on;
ModSecurityConfig modsecurity.conf;
```

```
root html;
index index.php index.html index.htm;
```

[.....]



```
GNU nano 2.2.6      File: nginx.conf      Modified

server {
    listen      80;
    server_name localhost;

    #charset koi8-r;

    #access_log logs/host.access.log main;

    location / {
        root    html;
        index   index.html index.htm;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;

Get Help  WriteOut  Read File  Prev Page  Cut Text  Cur Pos
Exit      Justify   Where Is   Next Page  UnCut Text To Spell
```

Ultimo paso (reiniciar nginx para aplicar los cambios): **Systemctl** restart nginx

```
root@debian:~# systemctl restart nginx
root@debian:~#
```

Probamos que nuestro nginx esté funcionando, ingresando al nombre del server que se le puso en este caso es “prueba.com” y nos debe mandar una pantalla así.



## 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](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

Thank you for using nginx.

Elaborado por Jesus\_Vega\_CERT\_10G

Para quitar la versión de nginx al producir un error se modificará el siguiente archivo

- `cd /usr/local/nginx/conf`
- `nano nginx.conf`

```
root@debian:~# cd /usr/local/nginx/conf/  
root@debian:/usr/local/nginx/conf# nano nginx.conf
```

```
GNU nano 2.2.6      File: nginx.conf      Modified  
  
keepalive_timeout 65;  
  
#gzip on;  
  
server {  
    listen      80;  
    server_name prueba.com;  
  
    #Para quitar la version de nginx  
    server_tokens off;  
  
    #charset koi8-r;  
  
    #access_log logs/host.access.log main;  
  
    location / {  
  
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos  
^X Exit      ^J Justify   ^W Where Is   ^N Next Page  ^U UnCut Text ^T To Spell
```

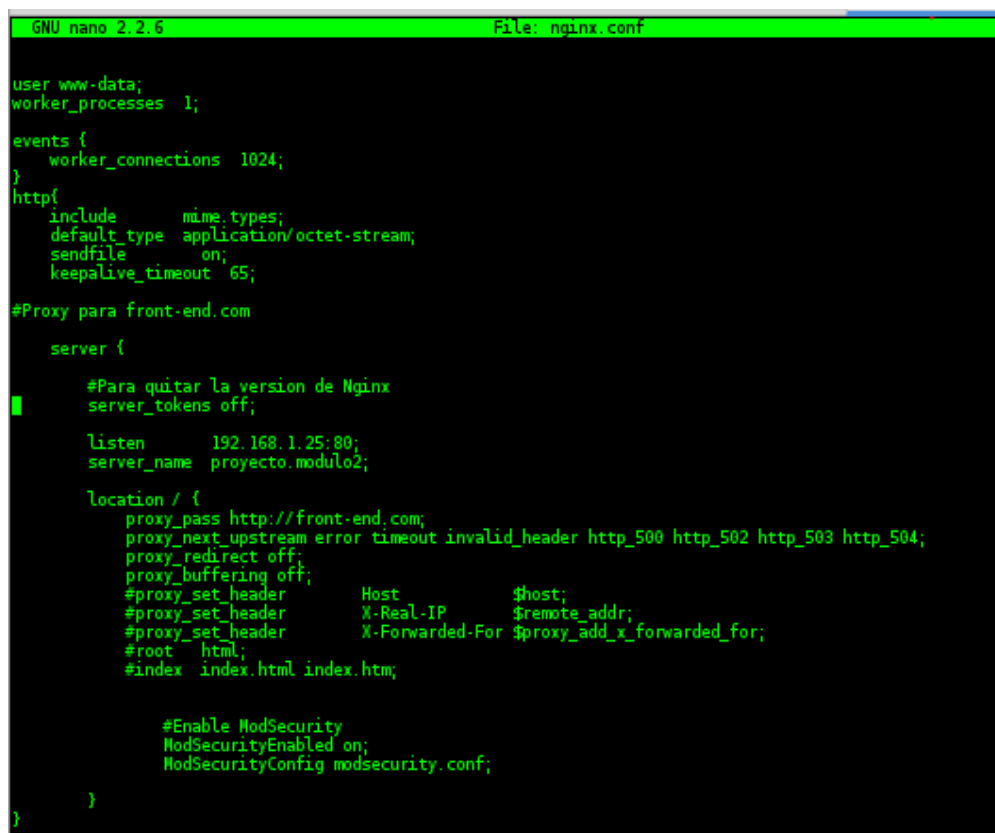
## Proxy inverso

Ingresamos a la siguiente ruta `cd /usr/local/nginx/conf` y modificaremos el archivo `nginx.conf`

```
root@debian:~# cd /usr/local/nginx/conf/  
root@debian:/usr/local/nginx/conf# nano nginx.conf
```

Se copiarán las siguientes líneas en el archivo:

```
server {  
    listen 192.168.1.25:80; (que es la dirección ip de nuestra máquina y el puerto)  
    server_name proyecto.modulo2; (que es el nombre de nuestro servidor)  
  
    listen / {  
        proxy_pass http://front-end.com (a donde nos redirige)  
        proxy_next_upstream error timeout invalid_header http_500 http_502 http_503 http_504;  
        proxy_redirect off;  
        proxy_buffering off;  
  
        #Enable ModSecurity  
        ModSecurity on;  
        ModSecurityConfig modsecurity.conf;  
    }  
}
```

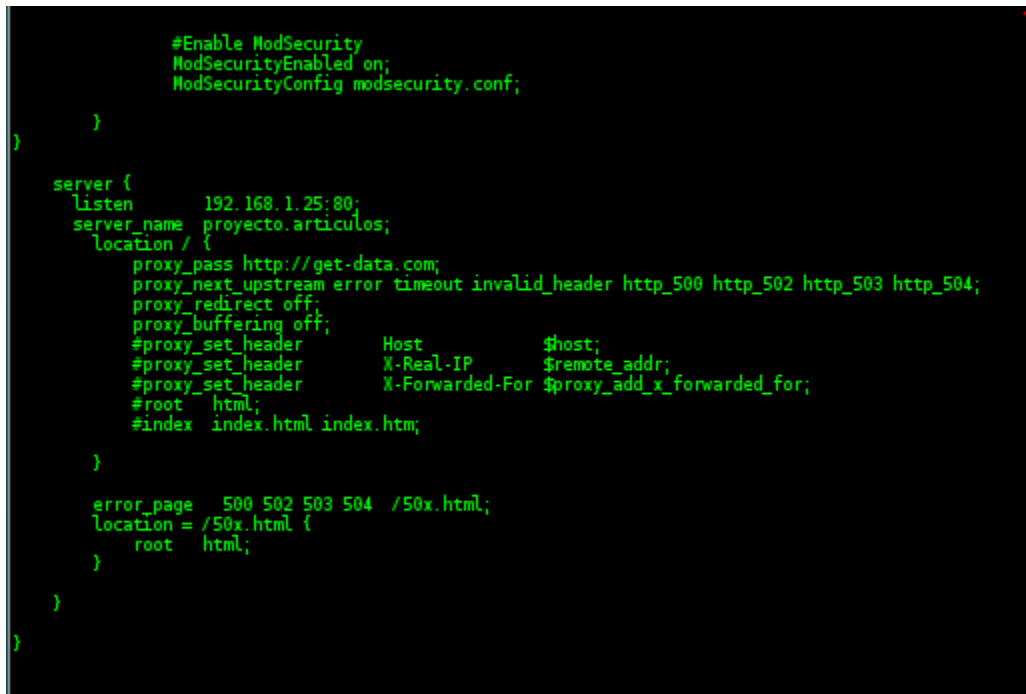


```
GNU nano 2.2.6 File: nginx.conf  
  
user www-data;  
worker_processes 1;  
  
events {  
    worker_connections 1024;  
}  
  
http {  
    include mime.types;  
    default_type application/octet-stream;  
    sendfile on;  
    keepalive_timeout 65;  
  
    #Proxy para front-end.com  
    server {  
        #Para quitar la version de Nginx  
        server_tokens off;  
  
        listen 192.168.1.25:80;  
        server_name proyecto.modulo2;  
  
        location / {  
            proxy_pass http://front-end.com;  
            proxy_next_upstream error timeout invalid_header http_500 http_502 http_503 http_504;  
            proxy_redirect off;  
            proxy_buffering off;  
            #proxy_set_header Host $host;  
            #proxy_set_header X-Real-IP $remote_addr;  
            #proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
            #root html;  
            #index index.html index.htm;  
  
            #Enable ModSecurity  
            ModSecurityEnabled on;  
            ModSecurityConfig modsecurity.conf;  
        }  
    }  
}
```

Agregaremos otro `server_name` debido a que cuando nos redirecciona a la página del front-end y abrimos un artículo nos aparece una página llamada `http://get-data.com` y NO nos muestra los artículos al agregar esto lo omite y nos muestra el contenido.

```
server{
    listen 192.168.1.25:80; (que es la dirección ip de nuestra máquina y el puerto)
    server_name proyecto.articulos; (que es el nombre de nuestro servidor)

listen / {
    proxy_pass http://get-data.com (a donde nos dirige)
    proxy_next_upstream error timeout invalid_header http_500 http_502 http_503 http_504;
    proxy_redirect off;
    proxy_buffering off;
}
```



```
#Enable ModSecurity
ModSecurityEnabled on;
ModSecurityConfig modsecurity.conf;

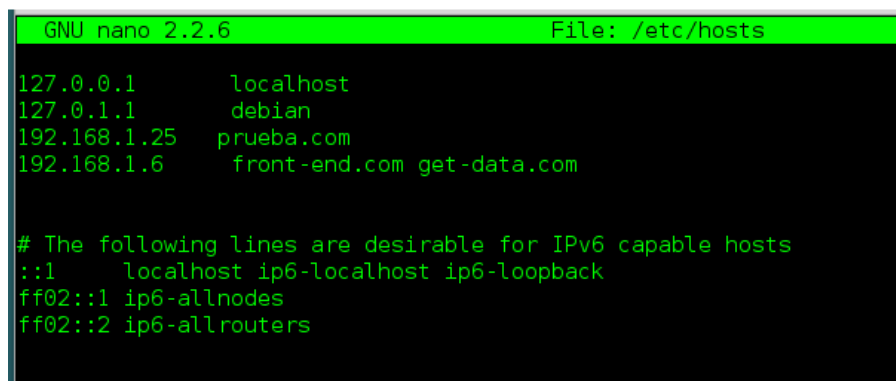
}

server {
    listen 192.168.1.25:80;
    server_name proyecto.articulos;
    location / {
        proxy_pass http://get-data.com;
        proxy_next_upstream error timeout invalid_header http_500 http_502 http_503 http_504;
        proxy_redirect off;
        proxy_buffering off;
        #proxy_set_header Host $host;
        #proxy_set_header X-Real-IP $remote_addr;
        #proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        #root html;
        #index index.html index.htm;
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root html;
    }
}

}
```

Y dentro de `/etc/hosts` agregaremos la ip del servidor front-end, get-data y la de nuestra máquina.



```
GNU nano 2.2.6 File: /etc/hosts

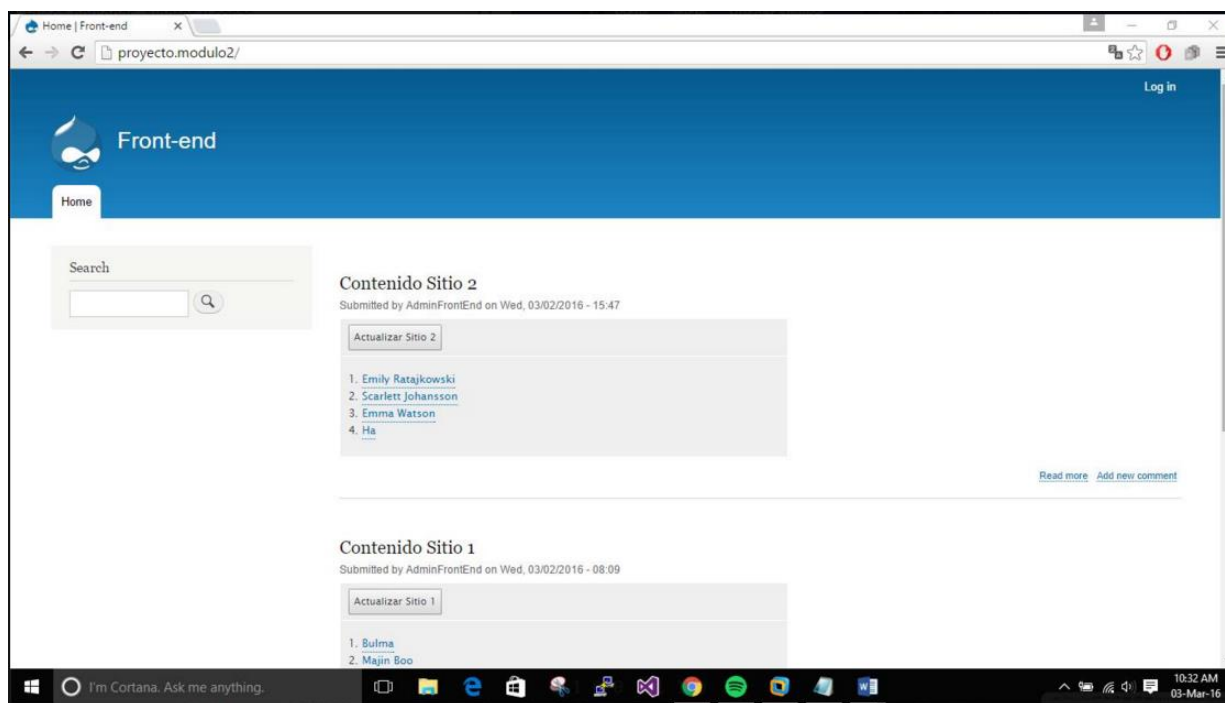
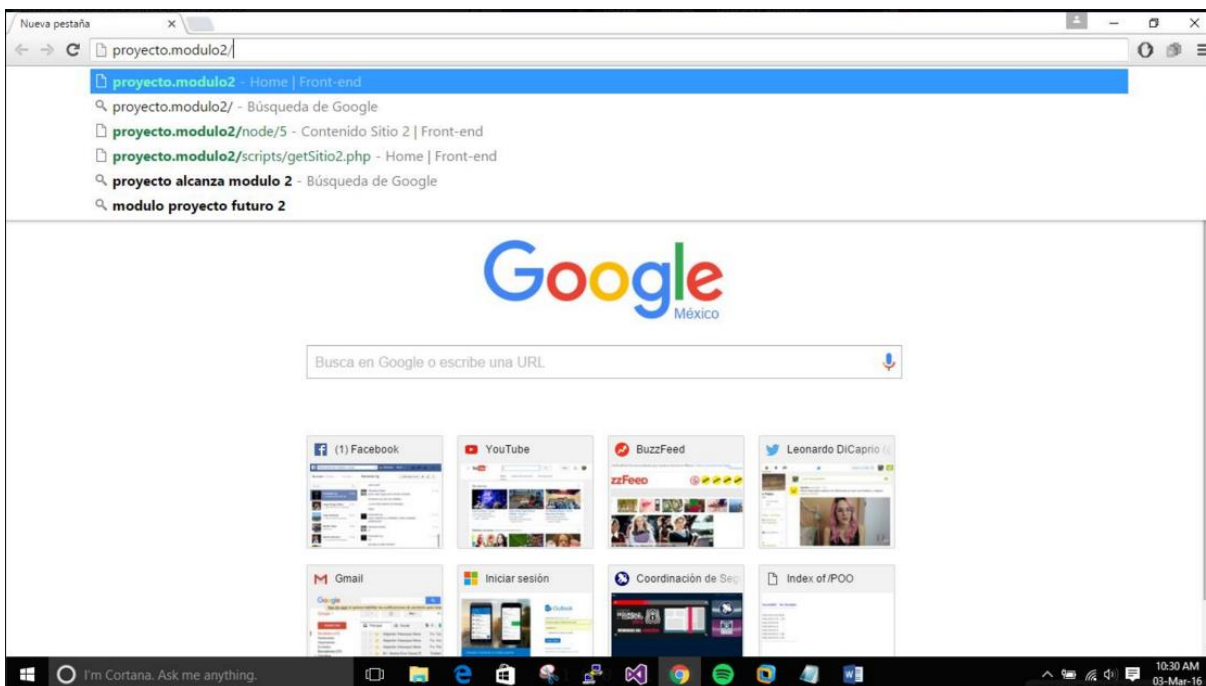
127.0.0.1 localhost
127.0.1.1 debian
192.168.1.25 prueba.com
192.168.1.6 front-end.com get-data.com

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

Se reinicia el servicio de nginx y se prueba en un cliente.

```
root@debian:/usr/local/nginx/conf# service nginx restart
root@debian:/usr/local/nginx/conf# nginx -t
nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok
nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful
root@debian:/usr/local/nginx/conf#
```

## CLIENTE



## REFERENCIAS

- Gautam, S. (2015). Drupal 8: Setting Up Multi-site. YouTube. Retrieved 26 February 2016, from <https://www.youtube.com/watch?v=uZ71WcQSc7o>
- Gautam, S. (2015). Drupal 8: Setting Up Multi-site | Blog • Sudhanshu Gautam. Sudhanshug.com. Retrieved 26 February 2016, from [http://www.sudhanshug.com/blog/multi\\_sites\\_with\\_d8](http://www.sudhanshug.com/blog/multi_sites_with_d8)
- Gautam, S. (2015). Drupal 8: Setting Up Multi-site. YouTube. Retrieved 27 February 2016, from <https://www.youtube.com/watch?v=uZ71WcQSc7o>
- Buytaert., D. (2014). Multi-site on Drupal 8 | Drupal.org. Drupal.org. Retrieved 1 March 2016, from <https://www.drupal.org/node/2297419>
- Hechoendrupal.gitbooks.io,. (2015). How to use Drupal Console in a multi-site installation | Drupal Console. Retrieved 28 February 2016, from <https://hechoendrupal.gitbooks.io/drupal-console/content/vn/using/how-to-use-drupal-console-in-a-multisite-installation.html>
- GORIPARTH, P. (2013). How to do point in time recovery with PostgreSQL 9.2 : PITR in CentOS 6/Redhat EL6. Open source database management systems - PostgreSQL & MySQL. Retrieved 3 March 2016, from <https://opensourcedbms.com/dbms/how-to-do-point-in-time-recovery-with-postgresql-9-2-pitr-3/>
- Postgresql.org,. (2015). PostgreSQL: Documentation: 9.1: Continuous Archiving and Point-in-Time Recovery (PITR). Retrieved 29 February 2016, from <http://www.postgresql.org/docs/9.1/static/continuous-archiving.html>
- Drupalconsole.com,. Home | Drupal Console. Retrieved 28 February 2016, from <https://drupalconsole.com/>
- Kalose, A. (2016). Drupal Console: Generate Module & Theme Code. YouTube. Retrieved 1 March 2016, from <https://www.youtube.com/watch?v=Si8azuZig78>
- nirmohi, a. (2008). Multi-site - Sharing the same code base | Drupal.org. Drupal.org. Retrieved 1 March 2016, from <https://www.drupal.org/documentation/install/multi-site>
- Postgresql.org,. (2015). PostgreSQL: Documentation: 8.0: CREATE USER. Retrieved 1 March 2016, from <http://www.postgresql.org/docs/8.0/static/sql-createuser.html>
- Postgresql.org,. (2015). PostgreSQL: Documentation: 9.0: CREATE DATABASE. Retrieved 1 March 2016, from <http://www.postgresql.org/docs/9.0/static/sql-createdatabase.html>
- Askubuntu.com,. (2015). Cannot connect to postgresql on port 5432. Retrieved 1 March 2016, from <http://askubuntu.com/questions/50621/cannot-connect-to-postgresql-on-port-5432>
- Postgresql.org,. (2015). PostgreSQL: Documentation: 8.1: psql. Retrieved 1 March 2016, from <http://www.postgresql.org/docs/8.1/static/app-psql.html>
- Ellingwood, J. (2014). How To Install Drupal on an Ubuntu 14.04 Server with Apache | DigitalOcean. Digitalocean.com. Retrieved 1 March 2016, from <https://www.digitalocean.com/community/tutorials/how-to-install-drupal-on-an-ubuntu-14-04-server-with-apache>
- Askubuntu.com,. (2016). Any way to search for text within nano?. Retrieved 2 March 2016, from <http://askubuntu.com/questions/47515/any-way-to-search-for-text-within-nano>
- Api.drupal.org,. system\_requirements | system.install | Drupal 8 | Drupal API. Retrieved 1 March 2016, from [https://api.drupal.org/api/drupal/core%21modules%21system%21system.install/function/system\\_re](https://api.drupal.org/api/drupal/core%21modules%21system%21system.install/function/system_re)

quirements/8

- Drupal.org,. drupal 8.0.4 | Drupal.org. Retrieved 29 February 2016, from <https://www.drupal.org/drupal-8.0.4-release-notes>
- Davies, M. (2016). Connecting your Drupal site to a Microsoft Active Directory server | Code Enigma.Codeenigma.com. Retrieved 29 February 2016, from <https://www.codeenigma.com/community/blog/connecting-your-drupal-site-microsoft-active-directory-server>
- Api.drupal.org,. example.sites.php | Drupal 8 | Drupal API. Retrieved 29 February 2016, from <https://api.drupal.org/api/drupal/sites%21example.sites.php/8>
- Drupal.org,. (2013). [Solved] The requested URL /drupal/admin/content was not found on this server. | Drupal.org. Retrieved 1 March 2016, from <https://www.drupal.org/node/2134281>
- r, n. (2008). Step 6: Configure clean URLs | Drupal.org. Drupal.org. Retrieved 1 March 2016, from <https://www.drupal.org/getting-started/clean-urls>
- r, n. (2008). Step 6: Configure clean URLs | Drupal.org. Drupal.org. Retrieved 1 March 2016, from <https://www.drupal.org/getting-started/clean-urls#dedicated>
- James, C. (2014). Multi-site on Drupal 8 | Drupal.org. Drupal.org. Retrieved 1 March 2016, from <https://www.drupal.org/node/2297419>
- Kalose, A. (2014). Drupal 8: Setting Up Multi-site | Akshay Kalose. Akshay Kalose. Retrieved 1 March 2016, from <http://www.kalose.net/oss/drupal-8-setting-multi-site/>
- Shalev, E. (2014). Drupal 8 Console | Drupal.org. Drupal.org. Retrieved 1 March 2016, from <https://www.drupal.org/node/2361869>
- Leers, W. (2016). Serialization module: (de)serializing data to/from JSON & more | Drupal.org.Drupal.org. Retrieved 1 March 2016, from <https://www.drupal.org/documentation/modules/serialization>
- Kalose, A. (2016). Drupal 8: Create a Simple Module. YouTube. Retrieved 1 March 2016, from <https://www.youtube.com/watch?v=79zYcloheCc>
- Api.drupal.org,. (2010). drupal\_json\_output | common.inc | Drupal 7 | Drupal API. Retrieved 1 March 2016, from [https://api.drupal.org/api/drupal/includes%21common.inc/function/drupal\\_json\\_output/7](https://api.drupal.org/api/drupal/includes%21common.inc/function/drupal_json_output/7)
- Kalose, A. (2016). Drupal 8: Create a Simple Module | Akshay Kalose. Akshay Kalose. Retrieved 1 March 2016, from <http://www.kalose.net/oss/drupal-8-create-simple-module/>
- Ditcheva, B. (2014). My first Drupal 8 module: step-by-step example | Drupalwoo. Drupalwoo.com. Retrieved 1 March 2016, from <http://www.drupalwoo.com/content/blog/my-first-drupal-8-module>
- Api.drupal.org,. (2014). default.settings.php | Drupal 6 | Drupal API. Retrieved 1 March 2016, from <https://api.drupal.org/api/drupal/sites!default!default.settings.php/6>
- (s.f.). Obtenido de <http://php.net/manual/en/reserved.variables.get.php>
- (s.f.). Obtenido de <http://php.net/manual/en/function.json-decode.php>
- dxx, ashish\_nirmohi, harings\_rob, & kenisha.lehari. (25 de Diciembre de 2015). Obtenido de <https://www.drupal.org/documentation/clearing-rebuilding-cache>
- 8, D. (s.f.). *api.drupal*. Obtenido de <https://api.drupal.org/api/drupal/core!lib!Drupal!Core!Form!FormStateInterface.php/function/FormStateInterface%3A%3AsetMethod/8>



- *api.drupal*. (s.f.). Obtenido de <https://api.drupal.org/api/drupal/core%21lib%21Drupal%21Core%21Entity%21EntityInterface.php/function/EntityInterface%3A%3Acreate/8>
- Comunnity. (s.f.). *http://php.net/manual*. Obtenido de <http://php.net/manual/en/function.empty.php>
- *Drupal 8*. (s.f.). Obtenido de <https://api.drupal.org/api/drupal/core%21modules%21comment%21comment.module/8>
- Ellingwood, J. (9 de Mayo de 2014). *Nginx, Security Ubuntu*. Obtenido de <https://www.digitalocean.com/community/tutorials/how-to-create-an-ssl-certificate-on-nginx-for-ubuntu-14-04>
- Garcia, E. (22 de Diciembre de 2015). *http://enzolutions.com/*. Obtenido de <http://enzolutions.com/articles/2015/12/22/how-to-get-content-type-fields-in-drupal-8/>
- jonawebb. (16 de Julio de 2012). *Community Documentation*. Obtenido de <https://www.drupal.org/node/1104482>
- Nginx Inc. (s.f.). *Nginx*. Obtenido de <https://www.nginx.com/resources/wiki/start/topics/recipes/drupal/>
- Ngnix. (s.f.). Obtenido de [http://nginx.org/en/docs/http/configuring\\_https\\_servers.html](http://nginx.org/en/docs/http/configuring_https_servers.html)
- Olivas, J. M. (27 de Marzo de 2014). *Console*. Obtenido de Drupal: <https://www.drupal.org/project/console>
- reallifedigital. (18 de Septiembre de 2014). Obtenido de <https://www.drupal.org/node/2011026>
- Sánchez, V. (17 de Diciembre de 2014). *Conocimiento Plus*. Obtenido de <https://conocimientoplus.wordpress.com/2014/12/17/installing-drupal-8-on-ubuntu-14-04-with-nginx/>
- Sipos, D. (14 de Junio de 2014). *sitepoint*. Obtenido de <http://www.sitepoint.com/build-drupal-8-module-routing-controllers-menu-links/>
- Rm-rf.es,. (2015). *Configurar NGINX como proxy inverso de Tomcat | rm-rf.es*. Retrieved 2 March 2016, from <http://rm-rf.es/configurar-nginx-como-proxy-inverso-de-tomcat/>
- YouTube,. (2015). *Demo Nginx As Reverse Proxy Server*. Retrieved 2 March 2016, from <https://www.youtube.com/watch?v=Nlnk3AMXU7s>
- Risager, T. (2015). *Compiling Nginx with ModSecurity on Ubuntu 14.04 LTS*. Stickleback. Retrieved 29 February 2016, from <https://blog.stickleback.dk/nginx-modsec-on-ubuntu-14-04-lts/>
- Howtoforge.com,. (2016). *Installing Nginx With PHP5 (And PHP-FPM) And MySQL Support (LEMP) On Ubuntu 14.04 LTS*. Retrieved 2 March 2016, from <https://www.howtoforge.com/installing-nginx-with-php5-fpm-and-mysql-on-ubuntu-14.04-lts-lemp>
- Ueland, C. (2013). *How to install Mod\_Security on Nginx | Nginx Tips*. ScaleScale.com. Retrieved 2 March 2016, from [https://www.scalescale.com/tips/nginx/how-to-install-mod\\_security-on-nginx/](https://www.scalescale.com/tips/nginx/how-to-install-mod_security-on-nginx/)
- Arul, M. (2015). *How to Install Nginx with ModSecurity on Ubuntu 15.04*. Howtoforge.com. Retrieved 2 March 2016, from [https://www.howtoforge.com/tutorial/install-nginx-with-mod\\_security-on-ubuntu-15-04/](https://www.howtoforge.com/tutorial/install-nginx-with-mod_security-on-ubuntu-15-04/)

- Ueland, C. (2013). How to hide Nginx version | Nginx Tips. ScaleScale.com. Retrieved 2 March 2016, from <https://www.scalescale.com/tips/nginx/how-to-hide-nginx-version/>