

INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

MANUAL DE CONFIGURACIÓN
PSA 1: SERVIDOR DE RECURSOS WEB



Cubas Rodríguez Francisco Ángel
Hernández López Alfredo
Estrada Mateos Elisabeth Eunice

Manuel Alejandro Soto Ramos

Administración de Servicios en Red

4CV4

20/Febrero/2019

Índice

Resumen	3
Requerimientos de software	3
Instalación del servidor	3
Configuración del servidor	4
Contenedor virtual por IP o dominio	4
Restricción de acceso al recurso	7
Configuración del puerto de operación	8
Servidor de aplicación HTTPS	8
Personalización de páginas de error para todos los sitios	10
Configuración de archivos de bitácoras y mensajes de error	11
Resumen de operación de forma dinámica	11
Pruebas	13
Referencias	15

Resumen

En este presente documento se realiza la configuración desde cero del servidor de protocolo HTTP NGINX y está dirigido para cualquier persona que desee realizar su instalación y configuración en el sistema operativo Ubuntu 16.04 LTS.

Requerimientos de Software

Para realizar la instalación y configuración de este servidor se requiere lo siguiente:

- S.O. Ubuntu 16.04 LTS
- NGINX
- OpenSSL

Instalación del servidor

Antes de realizar la instalación es necesario el siguiente comando a través de la terminal para se instale la base estándar de Linux para una autoridad certificadora de GNU privacy guard con el herramienta curl que nos permite la transferencia de datos de un servidor, en este caso será del servidor de Linux.

```
sudo apt install curl gnupg2 ca-certificates lsb-release
```

```
angelrdgz@HP-Pavilion14:~$ sudo apt install curl gnupg2 ca-certificates lsb-release
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
ca-certificates ya está en su versión más reciente (20170717~16.04.2).
curl ya está en su versión más reciente (7.47.0-1ubuntu2.12).
gnupg2 ya está en su versión más reciente (2.1.11-6ubuntu2.1).
lsb-release ya está en su versión más reciente (9.20160110ubuntu0.2).
```

Se importa la llave de firmado de nginx para verificar la autenticidad de los paquetes

```
curl -fsSL https://nginx.org/keys/nginx_signing.key | sudo apt-key add
```

Se verifica que ya tengamos la llave y nos mostrará los detalles de la firma

```
sudo apt-key fingerprint ABF5BD827BD9BF62
```

```
angelrdgz@HP-Pavilion14:~$ sudo apt-key fingerprint ABF5BD827BD9BF62
pub 2048R/7BD9BF62 2011-08-19 [[caduca: 2024-06-14]]
    Huella de clave = 573B FD6B 3D8F BC64 1079 A6AB ABF5 BD82 7BD9 BF62
uid nginx signing key <signing-key@nginx.com>
```

Instalamos el servidor con el siguiente comando

```
sudo apt install nginx
```

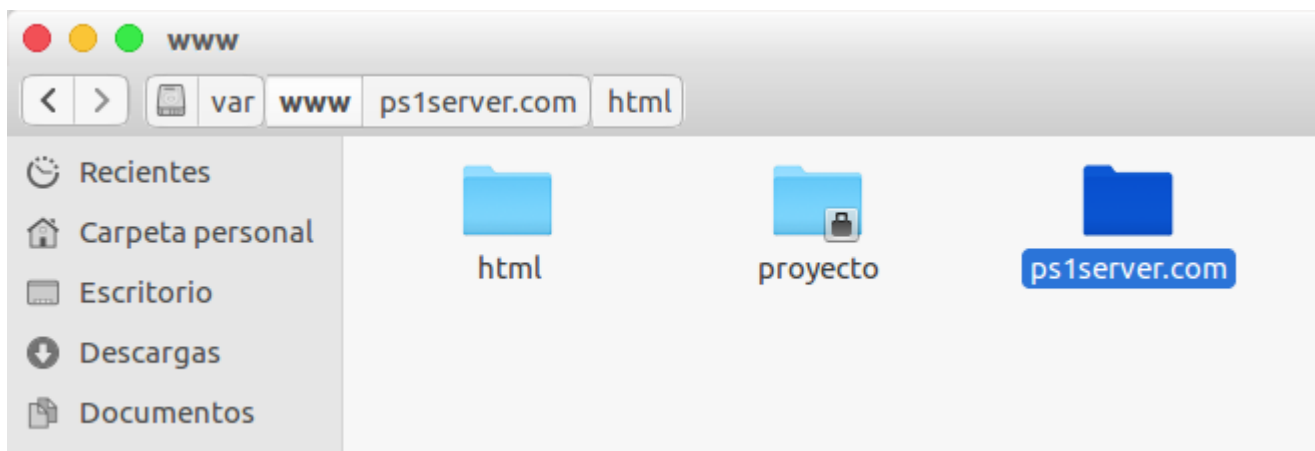
Configuracion del Servidor

1. Contenedor virtual por IP o dominio

Crearemos un contenedor o host virtual para nuestro servidor, por la ruta del servidor por default es `var/www/html`.

Crearemos una nueva carpeta para nuestro servidor dentro de `var/www/` con el nombre que queremos para nuestro dominio con el siguiente comando o desde el explorador de archivos (se deben tener permisos sobre la carpeta `www`)

```
sudo mkdir -p var/www/ps1server.com/html
```



Cambiamos de propietario la carpeta para que la podamos crear y editar directamente desde los archivos.

```
sudo chown -R $USER:$USER /var/www/ps1server.com/html
```

Copiamos el contenido del servidor default para nuestro nuevo host.

```
sudo cp /etc/nginx/sites-available/default /etc/nginx/sites-available/ps1server.com
```

Entramos a editar la configuración de nuestro server con el siguiente comando:

```
sudo nano /etc/nginx/sites-available/ps1server.com
```

Se mostrará el archivo como sigue:

```
GNU nano 2.5.3 Archivo: /etc/nginx/sites-available/ps1server.com Modificado
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# http://wiki.nginx.org/Pitfalls
# http://wiki.nginx.org/QuickStart
# http://wiki.nginx.org/Configuration
#
# Generally, you will want to move this file somewhere, and start with a clean
# file but keep this around for reference. Or just disable in sites-enabled.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    #
    listen 443 ssl default_server;
    listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    include snippets/snakeoil.conf;

    root /var/www/ps1server.com/html;

    # Add index.php to the list if you are using PHP
}
```

Toda la configuración irá dentro del bloque server como sigue:

```
server {
    listen 80;
    listen [::]:80;
    root /var/www/ps1server.com/html;
    index index.php index.html index.htm index.nginx-debian.html;

    server_name ps1server.com www.ps1server.com

    location / {
        try_files $uri $uri/ =404;
    }
}
```

Se continua a habilitar nuestro servidor haciendo un link simbólico a la carpeta sites-enabled

```
sudo ln -s /etc/nginx/sites-available/ps1server.com /etc/nginx/sites-enabled/
```

Entramos al archivo configuración de NGINX

```
sudo nano /etc/nginx/nginx.conf
```

Habilitamos la opción quitando el # de comentario.

```
#server_names_hash_bucket_size 64;
```

```
server_names_hash_bucket_size 64;  
# server_name_in_redirect off;
```

Reiniciamos el servidor para que se aplique la nueva configuración

```
sudo systemctl restart nginx
```

Finalmente agregamos la IP para nuestro host en el archivo de hosts de nuestro equipo

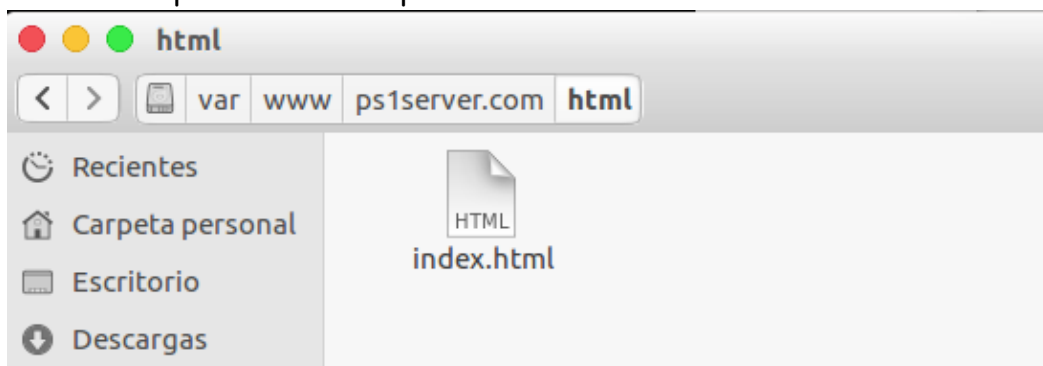
```
sudo nano /etc/hosts
```

```
GNU nano 2.5.3 Arc  
127.0.0.1 localhost  
127.0.1.1 HP-Pavilion14  
192.168.0.8 ps1server.com www.ps1server.com
```

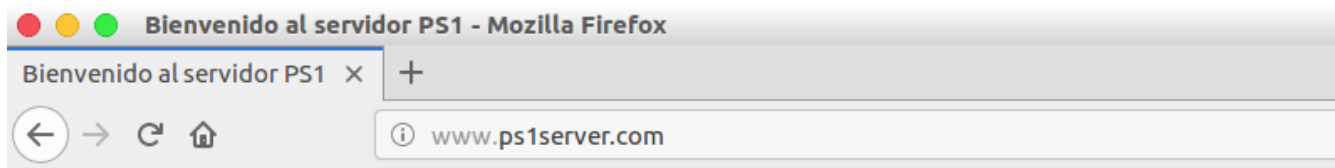
Creamos el siguiente archivo index.html para realizar la prueba de nuestro contenedor.

```
01. <!DOCTYPE html>  
02. <html>  
03.   <head>  
04.     <meta charset="utf-8">  
05.     <title>Bienvenido al servidor PS1</title>  
06.   </head>  
07.   <body>  
08.     <h1>El servidor está funcionando!</h1>  
09.   </body>  
10. </html>
```

Lo guardamos en la carpeta html creada para nuestro servidor

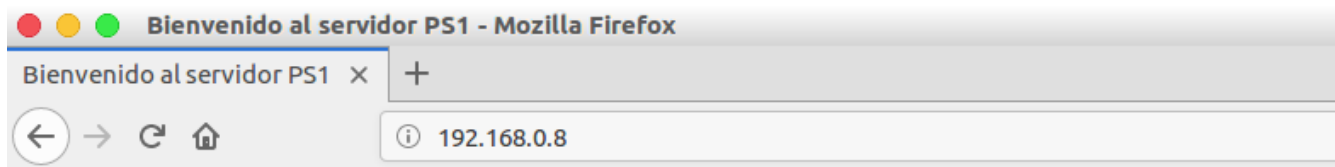


Realizamos la prueba de que funciona nuestro contenedor ya sea con IP o dominio que fue establecido.



El servidor está funcionando!

Prueba usando el dominio www.ps1server.com



El servidor está funcionando!

Prueba usando IP

1.1 Restricción de acceso al recurso por dirección IP del cliente

Dentro de la configuración del servidor se agrega la directiva deny seguida de la IP.

```
location / {  
    try_files $uri $uri/ =404;  
    deny 192.168.0.3  
}
```

1.2 Restricción de acceso al recurso por segmento de red

De igual manera se usa la directiva deny seguida del segmento de red a negar.

```
location / {  
    try_files $uri $uri/ =404;  
    deny 192.168.0.1/24  
}
```

1.3 Restricción de acceso al recurso por nombre de usuario (grupo de usuarios)/clave de acceso

Se crea un usuarios a través de la herramienta htpasswd como sigue:

```
sudo htpasswd -c /etc/nginx/.htpasswd user1
```

```
angelrdgz@HP-Pavilion14:~$ sudo htpasswd -c /etc/nginx/.htpasswd user1
New password:
Re-type new password:
Adding password for user user1
```

Agregamos la siguiente directiva en nuestra configuración para que considere el archivo donde están almacenados los usuarios.

```
location / {
    try_files $uri $uri/ =404;
    auth_basic "Ingresa tu credencial para acceder al sitio"
    auth_basic_user_file /etc/nginx/.htpasswd
}
```

2. Configuración del puerto de operación

El puerto se configura en el archivo de configuración de nuestro servidor con la directiva listen.

```
server {
    listen 80;
    listen [::]:80;
}
```

Se hace uso de dos diferentes notaciones, listen 80 para IPv4 y listen [::]:80 para IPv6.

3. Servidor de aplicación utilizando el protocolo HTTPS

Creamos la carpeta SSL dentro de la ruta /etc/nginx

```
mkdir /etc/nginx/ssl
```

Generamos los archivos de certificado y llave privada

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/nginx/ssl/nginx.key -out /etc/nginx/ssl/nginx.cr
```

donde:

- openssl: La herramienta de línea de comandos para administrar certificados, llaves y otros archivos SSL.
- req: especifica que queremos usar la solicitud de firma de certificado (Certificate Signing Request CSR) X.509. X.509 es una infraestructura estándar de llave pública que SSL y TLS adhieren para su gestión de llaves y certificados.
- nodes: omite la opción de asegurar nuestros certificados con una contraseña.
- days 365: indica el tiempo que el certificado será considerado como válido.
- newkey rsa:2048: especifica que queremos generar un nuevo certificado y una nueva llave al mismo tiempo con el sistema criptográfico RSA de 2048 bits de longitud.
- keyout y out: Indican las rutas de salida de la llave privada y el certificado respectivamente.

Al finalizar la ejecución del comando nos pedirá llenar una encuesta de por qué estás solicitando un certificado. Es importante que en el campo `common name` se ponga el dominio del servidor.

```
angelrdgz@HP-Pavilion14:~$ sudo 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]:Ciudad de Mexico
Locality Name (eg, city) []:CDMX
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ESCOM
Organizational Unit Name (eg, section) []:School
Common Name (e.g. server FQDN or YOUR name) []:ps1server.com
Email Address []:facr_07@hotmail.com
```

Agregamos la llave y certificado a nuestro archivo de configuración en la sección de SSL, además debemos habilitar ambas directivas listenquitando el `#` de comentario.

```
sudo nano /etc/nginx/sites-available/ps1server.com
```

```
# SSL configuration
#
listen 443;
listen [::]:443;
ssl_certificate /etc/nginx/ssl/nginx.crt;
ssl_certificate_key /etc/nginx/ssl/nginx.key;
```

Verificamos que no haya error de sintaxis en el archivo de configuración.

```
sudo nginx -t
```

Reiniciamos NGINX

```
sudo systemctl restart nginx
```

4. Personalización de páginas de error para todos los sitios

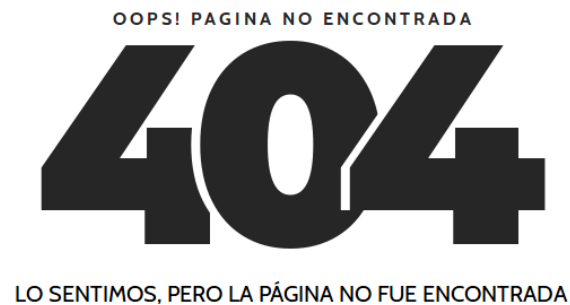
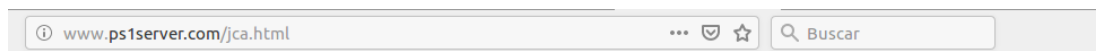
Se agrega la directiva `error_page` en nuestro archivo de configuración.

```
error_page 404 /404.html;  
location /404.html {  
    root /var/www/error_pages;  
}
```

```
error_page 401 /401.html;  
location /401.html {  
    root /var/www/error_pages;  
}
```

```
error_page 403 /403.html;  
location /403.html {  
    root /var/www/error_pages;  
}
```

Ejemplo:



5. Configuración de archivos de bitácoras y mensajes de error.

Se agrega la directiva `error_log` para errores y `access_log` para bitácoras a nuestro archivo de configuración

```
error_log /var/log/nginx/ps1server.error_log debug;  
dentro del bloque server.
```

Se indica la ruta del archivo que guardará los mensajes y el nivel de severidad. En nuestro caso utilizamos el nivel debug.

```
access_log /var/log/nginx/access.log  
dentro del bloque http
```

6. Resumen de operación de forma dinámica (Sitios, solicitudes, estado del sistema y recursos consumidos)

Instalamos pip de python

```
sudo apt-get install python pip
```

```
angelrdgz@HP-Pavilion14:~$ sudo apt-get install python-pip  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Se instalarán los siguientes paquetes adicionales:  
  libexpat1-dev libpython-all-dev libpython-dev libpython2.7-dev python-all  
  python-all-dev python-dev python-pip-whl python-setuptools python-wheel  
  python2.7-dev  
Paquetes sugeridos:  
  python-setuptools-doc  
Se instalarán los siguientes paquetes NUEVOS:  
  libexpat1-dev libpython-all-dev libpython-dev libpython2.7-dev python-all  
  python-all-dev python-dev python-pip python-pip-whl python-setuptools
```

Instalamos la herramienta para monitoreo NGXTOP a través de pip

```
sudo pip install ngxtop
```

```
angelrdgz@HP-Pavilion14:~$ sudo pip install ngxtop
[sudo] password for angelrdgz:
The directory '/home/angelrdgz/.cache/pip/http' or its parent directory is not owned by the current user and the cache has been disabled. Please check the permissions and owner of that directory. If executing pip with sudo, you may want sudo's -H flag.
The directory '/home/angelrdgz/.cache/pip' or its parent directory is not owned by the current user and caching wheels has been disabled. check the permissions and owner of that directory. If executing pip with sudo, you may want sudo's -H flag.
Collecting ngxtop
  Downloading https://files.pythonhosted.org/packages/c7/20/9aafbe173b596dfbc996f1b8b957da88d160cf1c5279179d3b4990f40e00/ngxtop-0.0.2-py2.py3-none-any.whl
Collecting tabulate (from ngxtop)
  Downloading https://files.pythonhosted.org/packages/c2/fd/202954b3f0eb896c53b7b6f07390851b1fd2ca84aa95880d7ae4f434c4ac/tabulate-0.8.3.tar.gz (46kB)
    100% |████████████████████████████████████████| 51kB 534kB/s
Collecting docopt (from ngxtop)
  Downloading https://files.pythonhosted.org/packages/a2/55/8f8cab2afd404cf578136ef2cc5dfb50baa1761b68c9da1fb1e4eed343c9/docopt-0.6.2.tar.gz
Collecting pyparsing (from ngxtop)
  Downloading https://files.pythonhosted.org/packages/de/0a/001be530836743d8be6c2d85069f46fecf84ac6c18c7f5fb8125ee11d854/pyparsing-2.3.1-py2.py3-none-any.whl (61kB)
```

Ejecutamos la herramienta de monitoreo
sudo ngxtop

```

angelrdgz@HP-Pavilion14: ~
running for 591 seconds, 20 records processed: 0.03 req/sec

Summary:
| count | avg_bytes_sent | 2xx | 3xx | 4xx | 5xx |
|-----+-----+-----+-----+-----+-----|
| 20 | 6645.500 | 2 | 14 | 4 | 0 |

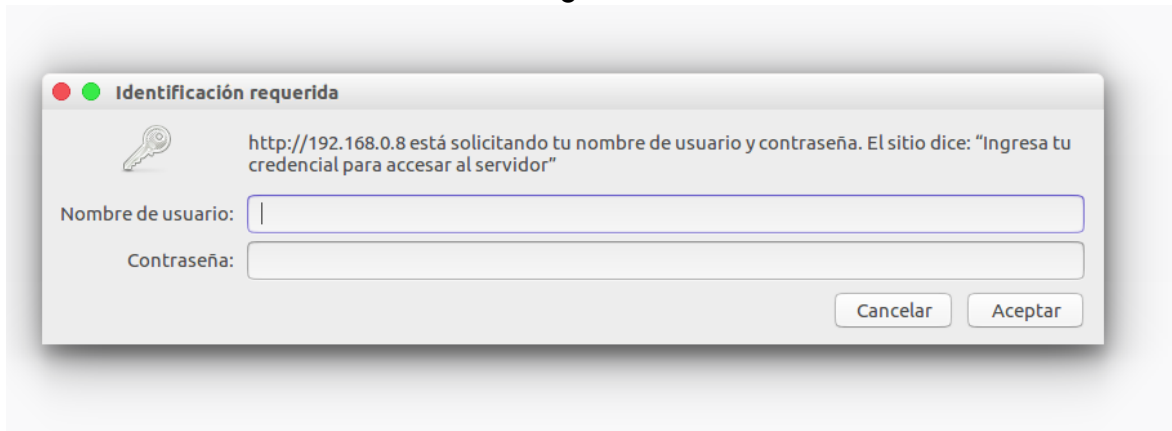
Detailed:
| request_path | count | avg_bytes_sent | 2xx | 3xx | 4xx | 5xx |
|-----+-----+-----+-----+-----+-----+-----|
| / | 2 | 104.500 | 0 | 1 | 1 | 0 |
| /man.html | 2 | 209.000 | 0 | 0 | 2 | 0 |
| /css/bootstrap.min.css | 1 | 121200.000 | 1 | 0 | 0 | 0 |
| /css/style.css | 1 | 0.000 | 0 | 1 | 0 | 0 |
| /favicon.ico | 1 | 1181.000 | 0 | 0 | 1 | 0 |
| /img/copyright.jpg | 1 | 0.000 | 0 | 1 | 0 | 0 |
| /img/dise%C3%B1oInd.jpg | 1 | 0.000 | 0 | 1 | 0 | 0 |
| /img/home-background.jpg | 1 | 0.000 | 0 | 1 | 0 | 0 |
| /img/logo-alt.png | 1 | 0.000 | 0 | 1 | 0 | 0 |
| /img/logo.png | 1 | 9902.000 | 1 | 0 | 0 | 0 |

```

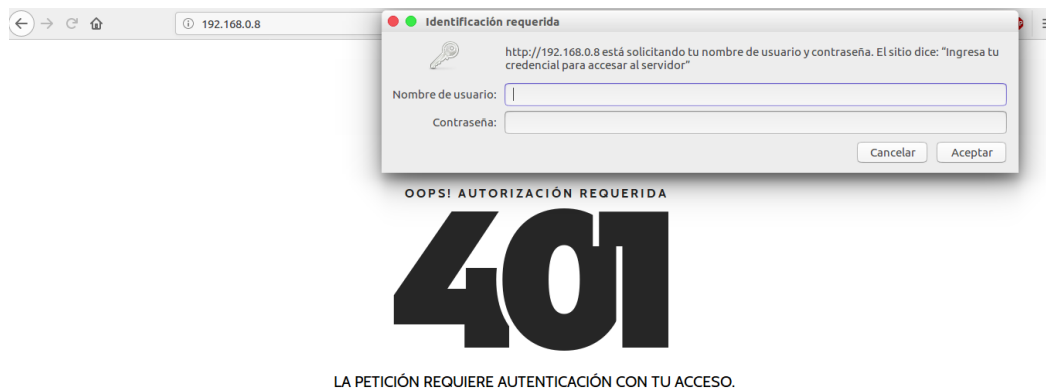
En lapantalla de monitoreo se muestran los recursos que se ocuparon de nuestro servidor y un resumen que muestra el total de recursos y total de códigos de estado según su tipo 2xx: peticiones correctas, 3xx:redirecciones, 4xx:errores del cliente, 5xx: errores del servidor.

Pruebas

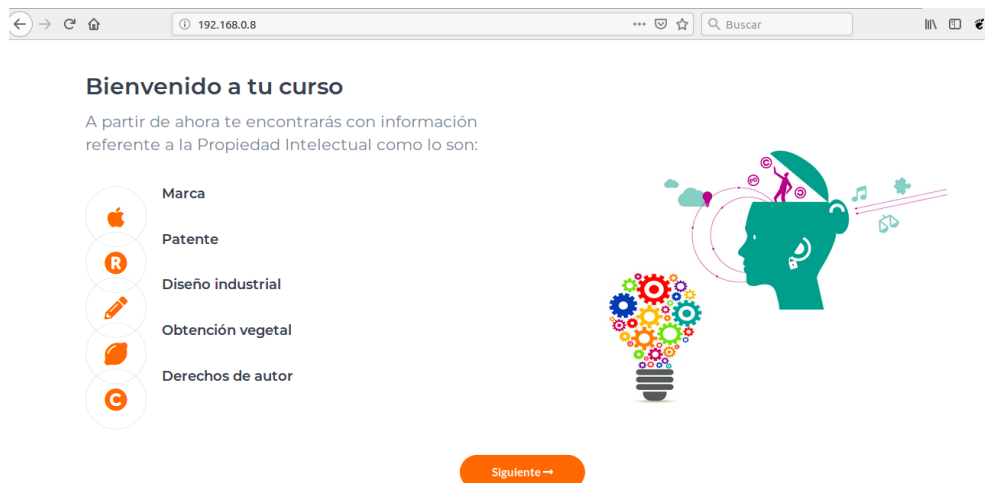
Cargamos el contenido de nuestro sitio web cargado en el servidor.



Se solicita el acceso con usuario.



En caso de error se mostrará la página personalizada para el error generado, si se ingresó el usuario correcto mostrará el contenido del sitio



Si abrimos nuestro archivo access.log podemos observar de igual forma los recursos ocupados y con la IP que fueron consumidos, además del navegador y S.O en el que fue abierto.

```
192.168.43.153 - - [25/Feb/2019:19:14:27 -0600] "GET /img/logo.png HTTP/1.1" 200 9902 "http://192.168.43.124/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.119 Safari/537.36"
192.168.43.153 - - [25/Feb/2019:19:14:27 -0600] "GET /img/obtencionVeg.jpg HTTP/1.1" 200 265905 "http://192.168.43.124/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.119 Safari/537.36"
192.168.43.153 - - [25/Feb/2019:19:14:28 -0600] "GET /img/home-background.jpg HTTP/1.1" 200 733677 "http://192.168.43.124/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.119 Safari/537.36"
192.168.43.153 - - [25/Feb/2019:19:14:33 -0600] "GET /favicon.ico HTTP/1.1" 404 1181 "http://192.168.43.124/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.119 Safari/537.36"
192.168.43.153 - - [25/Feb/2019:19:15:05 -0600] "GET /man.html HTTP/1.1" 404 1181 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.119 Safari/537.36"
192.168.43.153 - - [25/Feb/2019:19:16:39 -0600] "GET / HTTP/1.1" 404 209 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.119 Safari/537.36"
192.168.43.153 - - [25/Feb/2019:19:16:45 -0600] "GET / HTTP/1.1" 404 209 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.119 Safari/537.36"
192.168.43.69 - - [25/Feb/2019:19:17:08 -0600] "GET / HTTP/1.1" 200 7400 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:65.0) Gecko/20100101 Firefox/65.0"
192.168.43.69 - - [25/Feb/2019:19:17:08 -0600] "GET /css/style.css HTTP/1.1" 200 24533 "http://192.168.43.124/" "Mozilla/5.0 (X11; Linux x86_64; rv:65.0) Gecko/20100101 Firefox/65.0"
192.168.43.69 - - [25/Feb/2019:19:17:08 -0600] "GET /js/main.js HTTP/1.1" 200 266 "http://192.168.43.124/" "Mozilla/5.0 (X11; Linux x86_64; rv:65.0) Gecko/20100101 Firefox/65.0"
192.168.43.69 - - [25/Feb/2019:19:17:09 -0600] "GET /js/bootstrap.min.js HTTP/1.1" 200 37045 "http://192.168.43.124/" "Mozilla/5.0 (X11; Linux x86_64; rv:65.0) Gecko/20100101 Firefox/65.0"
192.168.43.69 - - [25/Feb/2019:19:17:09 -0600] "GET /js/jquery.min.js HTTP/1.1" 200 85582 "http://192.168.43.124/" "Mozilla/5.0 (X11; Linux x86_64; rv:65.0) Gecko/20100101 Firefox/65.0"
192.168.43.69 - - [25/Feb/2019:19:17:09 -0600] "GET /css/bootstrap.min.css HTTP/1.1" 200 121200 "http://192.168.43.124/" "Mozilla/5.0 (X11; Linux x86_64; rv:65.0) Gecko/20100101 Firefox/65.0"
192.168.43.69 - - [25/Feb/2019:19:17:09 -0600] "GET /img/logo-alt.png HTTP/1.1" 200 4870 "http://192.168.43.124/" "Mozilla/5.0 (X11; Linux x86_64; rv:65.0) Gecko/20100101 Firefox/65.0"
192.168.43.69 - - [25/Feb/2019:19:17:10 -0600] "GET /img/propInt.png HTTP/1.1" 200 96042 "http://192.168.43.124/" "Mozilla/5.0 (X11; Linux x86_64; rv:65.0) Gecko/20100101 Firefox/65.0"
```

De la misma manera en el archivo de errores se podrá ver el registro de los códigos de estado que se van presentando en tiempo real. Considerando el mismo tiempo que el anterior se indica el código de estado con su detalle.

```
2019/02/25 19:17:09 [debug] 4325#4325: *6 HTTP/1.1 200 OK
Server: nginx/1.10.3 (Ubuntu)
Date: Tue, 26 Feb 2019 01:17:09 GMT
Content-Type: image/png
Content-Length: 4870
Last-Modified: Thu, 16 Nov 2017 08:52:04 GMT
Connection: keep-alive
ETag: "5a0d51b4-1306"
Accept-Ranges: bytes
```

Referencias

- nginx documentation. (s.f.). Recuperado 19 febrero, 2019, de <http://nginx.org/en/docs/>
- Justin Ellingwood, J. E. (2018, 6 abril). How To Set Up Nginx Server Blocks (Virtual Hosts) on Ubuntu 16.04. Recuperado 20 febrero, 2019, de <https://www.digitalocean.com/community/tutorials/how-to-set-up-nginx-server-blocks-virtual-hosts-on-ubuntu-16-04>
- Justin Ellingwood, J. E. (2018b, 24 julio). How To Create a Self-Signed SSL Certificate for Nginx in Ubuntu 16.04. Recuperado 20 febrero, 2019, de <https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-nginx-in-ubuntu-16-04>
- Mitchell Anicas, M. A. (2017, 18 julio). How To Troubleshoot Common HTTP Error Codes. Recuperado 22 febrero, 2019, de <https://www.digitalocean.com/community/tutorials/how-to-troubleshoot-common-http-error-codes>
- NGINX Docs | Configuring Logging. (s.f.). Recuperado 24 febrero, 2019, de <https://docs.nginx.com/nginx/admin-guide/monitoring/logging/>