

Instituto Politécnico Nacional

Escuela Superior de Cómputo

Administración de servicios en red

“Servidor de aplicación 2”

Manual de instalación/configuración y operación

Equipo 10

Campos Gómez Fernanda Ivette

Jiménez Rodríguez Luis Martin

Vivia Delgadillo Rocío

HTTP Hypertext transfer protocol

El Protocolo de transferencia de hipertexto (HTTP) es un protocolo de aplicación para sistemas de información distribuida, colaborativa e hipermedia. [1] HTTP es la base de la comunicación de datos para la World Wide Web.

El hipertexto es texto estructurado que utiliza enlaces lógicos (hipervínculos) entre nodos que contienen texto. HTTP es el protocolo para intercambiar o transferir hipertexto.

El desarrollo de HTTP fue iniciado por Tim Berners-Lee en el CERN en 1989. El desarrollo de estándares de HTTP fue coordinado por Internet Engineering Task Force (IETF) y World Wide Web Consortium (W3C), culminando en la publicación de una serie de Requests for Comments (RFCs).

The Apache HTTP Server Project

Servidor Apache HTTP versión 2.4

Configuración

Instalación en Ubuntu/Debian

Hay dos maneras de instalarlo: por fuente, o por medio de la terminal, nosotros elegimos por medio de la terminal.

```
sudo apt-get update && sudo apt-get upgrade  
sudo apt-get install apache2
```

Una vez instalado, para iniciarlo solo basta con escribir el siguiente comando:

```
sudo service apache2 start
```

1. Contenedor virtual por IP y por dominio

Virtual hosts

Se refiere a la práctica de correr más de un sitio web sobre una sola máquina.

Pueden ser basado en IP que significa que se puede tener diferentes direcciones IP para cada sitio web, o basado por nombre, que significa que puede tener diferentes nombres corriendo en cada dirección IP.[2]

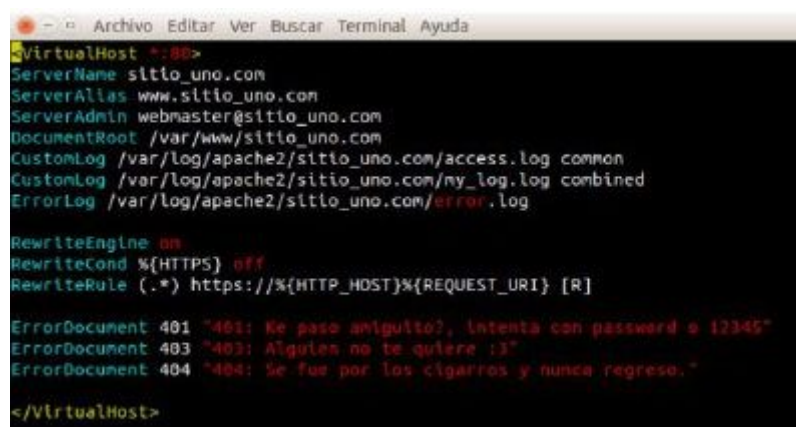
Apache nos brinda la oportunidad de tener más de un sitio como host en el servidor. Hay dos opciones que podemos utilizar: contenedor virtual por IP y contenedor virtual por dominio.

A continuación mostramos cómo realizar la configuración de los virtual hosts en mediante el método de contenedores por dominio.

Primero debemos ubicar la estructura general de los archivos en nuestro servidor. Contamos con la ruta **/etc/apache2**, la cual tendrá archivos para configurar el servidor. Para la creación de los virtual hosts, iremos a la carpeta **/etc/apache2/sites-available** y crearemos el archivo **sitio_uno.com.conf**, el cual será el archivo de configuración para nuestro primer host.

```
root@erreuvede:/home/rocio# vim /etc/apache2/sites-available/sitio_uno.com.conf
root@erreuvede:/home/rocio#
```

Ya teniendo el archivo creado, lo editaremos para agregar diferente información acerca de la configuración del host. Entre las cosas importantes por el momento para destacar, es que en esta sección podemos agregar el nombre de nuestro servidor, un alias, el nombre del administrador del host, el puerto en el que el sitio dará el servicio, la ruta hacia la página principal del sitio, así como la ruta de los logs que serán creados posteriormente.



```
VirtualHost *:80
ServerName sitio_uno.com
ServerAlias www.sitio_uno.com
ServerAdmin webmaster@sitio_uno.com
DocumentRoot /var/www/sitio_uno.com
CustomLog /var/log/apache2/sitio_uno.com/access.log common
CustomLog /var/log/apache2/sitio_uno.com/my_log.log combined
ErrorLog /var/log/apache2/sitio_uno.com/error.log

RewriteEngine on
RewriteCond %{HTTPS} off
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI} [R]

ErrorDocument 401 "401: ¿Ese paso amiguito?, intenta con password o 12345"
ErrorDocument 403 "403: Alguien no te quiere :3"
ErrorDocument 404 "404: Se fue por los cigarrillos y nunca regreso."

</VirtualHost>
```

Después de crear el archivo con la información, procedemos a agregar el siguiente comando.

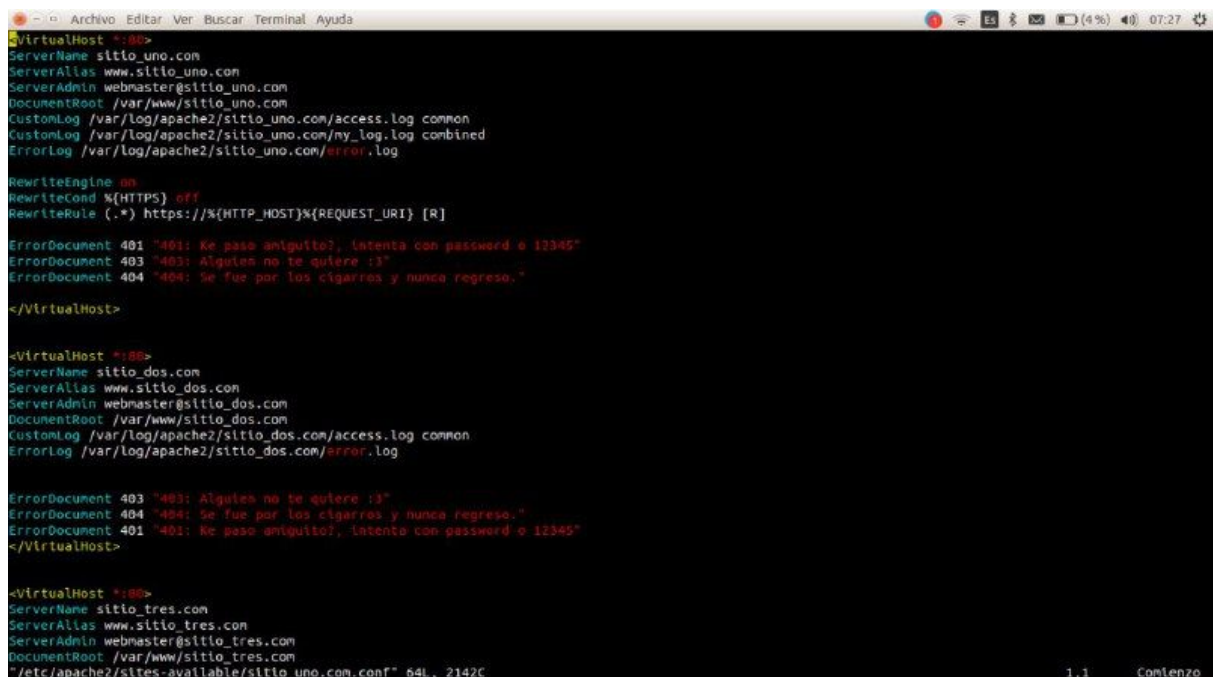
```
rocio@erreuve:~$ sudo su
root@erreuve:/home/rocio# chown -R www-data:www-data /var/www/sitio_uno.com
```

Otra ruta importante durante la configuración de nuestro servidor es la ruta de todos aquellos ficheros en donde se estará guardando la información de los sitios de nuestro servidor, en este caso contamos con la ruta: **/var/www/** en donde añadiremos los sitios que estarán en servicio.

```
rocio@erreuve:~$ sudo su
root@erreuve:/home/rocio# sudo mkdir /var/www/sitio_uno.com
```

Dentro de esa carpeta agregaremos la información del **sitio_uno.com**.

Ya teniendo el **sitio_uno.com** configurado, procedemos de la misma forma para configurar los demás sitios, la diferencia entre los demás sitios, es que en lugar de crear los archivos de configuración para cada uno de los hosts, agregaremos la información de la configuración de todos los sitios dentro de **sitio_uno.com.conf**, como se muestra a continuación, en donde agregamos otros dos sitios más.



```
Archivo  Editor  Ver  Buscar  Terminal  Ayuda
<VirtualHost *:80>
ServerName sitio_uno.com
ServerAlias www.sitio_uno.com
ServerAdmin webmaster@sitio_uno.com
DocumentRoot /var/www/sitio_uno.com
CustomLog /var/log/apache2/sitio_uno.com/access.log common
ErrorLog /var/log/apache2/sitio_uno.com/error.log combined
RewriteEngine on
RewriteCond %{HTTPS} off
RewriteRule (.*?) https://%{HTTP_HOST}%{REQUEST_URI} [R]

ErrorDocument 401 "401: Ke paso amiguito?, intenta con password o 12345"
ErrorDocument 403 "403: Alguien no te quiere :)"
ErrorDocument 404 "404: Se fue por los cigarras y nunca regreso."

</VirtualHost>

<VirtualHost *:80>
ServerName sitio_dos.com
ServerAlias www.sitio_dos.com
ServerAdmin webmaster@sitio_dos.com
DocumentRoot /var/www/sitio_dos.com
CustomLog /var/log/apache2/sitio_dos.com/access.log common
ErrorLog /var/log/apache2/sitio_dos.com/error.log

ErrorDocument 403 "403: Alguien no te quiere :)"
ErrorDocument 404 "404: Se fue por los cigarras y nunca regreso."
ErrorDocument 401 "401: Ke paso amiguito?, intenta con password o 12345"

</VirtualHost>

<VirtualHost *:80>
ServerName sitio_tres.com
ServerAlias www.sitio_tres.com
ServerAdmin webmaster@sitio_tres.com
DocumentRoot /var/www/sitio_tres.com
</etc/apache2/sites-available/sitio_uno.com.conf> 64L, 2142C
1,1 Contenze
```

Una vez configurado el archivo de configuración, procederemos a crear los ficheros correspondiente en **/var/www/sitio_dos.com**, **/var/www/sitio_tres.com**.

Para terminar con la configuración de los sitios, debemos de agregar las direcciones de los hosts con sus nombres de dominios en el archivo **/etc/apache2/hosts**, de la siguiente manera:

```
127.0.0.1      localhost
127.0.1.1      erreuvede
127.0.1.1      sitio_dos.com
127.0.1.1      sitio_uno.com
127.0.1.1      sitio_tres.com

# The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0      ip6-localnet
ff00::0      ip6-mcastprefix
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
~
~
```

Finalmente hacemos un reinicio del servidor mediante el comando **service apache2 restart**, y seguido de el utilizando el comando **service apache2 reload**.

Para probar su funcionamiento, podemos probar entrando al navegador ingresando cualquiera de los tres sitios: **sitio_uno.com**, **sitio_dos.com**, **sitio_tres.com**, y observar que los sitios se pueden acceder normalmente.

Ya teniendo el servicio corriendo con los tres sitios, es importante destacar que en ocasiones necesitaremos restringir el acceso a nuestro recursos por diferentes razones: por seguridad, permisos de acceso, o bloquear usuarios que han hecho un mal uso del servicio.

Las siguientes tres opciones nos brindan la posibilidad de crear estas restricciones de acceso.

1. Restringir acceso al recurso por dirección IP del cliente:

Este tipo de restricción nos permite crear lo que podríamos conocer como lista “blanca” o lista “negra”, mediante esta configuración podemos especificar qué usuarios son los únicos que tienen permiso de acceder, y de la misma forma cuáles usuarios está prohibida su entrada al servidor.

Para su configuración, deberemos modificar el archivo **apache2.conf** que se encuentra en la ruta **/etc/apache2/**.

```
root@erreuvede:/# vim /etc/apache2/apache2.conf
root@erreuvede:/#
```

Agregaremos un directorio para el sitio uno en el cual mediante las directivas de **require ip x.x.x.x**, y **require not ip x.x.x.x** podemos registrar quienes tienen o no permisos para acceder a ese directorio.

```
<Directory "/var/www/sitio_uno.com">
    Options All
    AllowOverride All
    <RequireAll>
        Require all granted
        Require ip 192.168.43.253
    </RequireAll>
</Directory>
```

En este ejemplo creamos un directorio para **/var/www/sitio_uno.com** y configuramos para que la ip 192.168.43.253 tenga permisos para entrar al sitio uno.

2. Restringir acceso al recurso por segmento de red:

La restricción por acceso de red es similar a la restricción ip individual, la diferencia entre ambas es que en el segmento de red, nosotros podemos indicar todo un conjunto de direcciones de ip continuas en la red para si tienen o no acceso al servicio.

La configuración se realiza mediante la creación de un directorio en el archivo **/etc/apache2/apache2.conf**, indicando con las directivas **require ip x.x.x.x/y.y.y.y** o **require not ip x.x.x.x/y.y.y.y** indicando qué segmentos tienen o no permisos.

```
<Directory "/var/www/sitio_tres.com">
    Options All
    AllowOverride All
    <RequireAll>
        Require all granted
        Require not ip 192.168.43.21 192.168.43.30
    </RequireAll>
</Directory>
```

En el ejemplo anterior, estamos creando un directorio para el **sitio_tres.com**, negando el acceso a este sitio en la sección de direcciones IP que van de 192.168.43.21 hasta 192.168.43.30.

3. Restringir acceso al recurso por nombre de usuario (grupo de usuarios)/clave de acceso:

Este tipo de restricción, define a un grupo de usuarios los cuales serán capaces de entrar al sitio que estemos configurando. Un usuario al tratar de acceder al sitio, obtendrá una ventana de autenticación en la cual deberá ingresar su usuario y contraseña para ingresar.

Para este tipo de restricción, primero deberemos crear un archivo `/etc/apache2/.htpasswd`, en el cual podremos dar de alta usuarios, así como asignarles una contraseña para que puedan ingresar. Cabe destacar que las contraseñas se guardaran en el archivo de manera encriptada, utilizando cifrado md5.

Utilizando el siguiente comando, crearemos el archivo de los usuarios con sus contraseñas.

```
root@erreuvede:/# htpasswd -c /etc/apache2/.htpasswd Usuario
```

Se nos pedirá la contraseña que asignaremos al usuario en cuestión.

```
root@erreuvede:/# htpasswd /etc/apache2/.htpasswd Usuario2
New password:
Re-type new password:
```

De la misma manera procedemos para crear tantos usuarios como queramos registrar. Posteriormente podemos revisar el archivo `.htpasswd` para poder consultar la lista de usuarios registrados, puede notar como las contraseñas se nos muestran de manera cifrada, de modo que no podemos conocer su valor original.


```

root@erreuvede:/# cat /etc/apache2/.htpasswd
hola:$apr1$t1Jx5ZFm$VKgak9or5MU55z3YrYQ8U0
Frederick:$apr1$6YEKBM4b$EW5BHUbIUWNqbeUBa0Xwy/
Fefe:$apr1$f8pj29fI$DXJyk6MrgFCfHeT4adH/F/
capitanaza:$apr1$2DJd04IE$9kKaYaaTErmL7NSRafvkP1
Usuario2:$apr1$VFCZCUDx$DnVEKXbYrBZhsRp8BK87F/
root@erreuvede:/# █

```

Una vez creado el archivo de usuarios, procedemos a crear las restricciones en el sitio; así como en las restricciones pasadas, crearemos un directorio en el cual indicaremos que su acceso está restringido y que solo los usuarios que aparezcan en la lista de usuarios tiene permitido acceder al sitio, siempre y cuando conozcan su contraseña.

Para indicar las restricciones, ocuparemos las directivas **AuthType Basic** indicando que se utilizará el método básico de autenticación, **AuthUserFile "ruta_htpasswd"** indicando en donde se encuentra el archivo que define a los usuarios, y la directiva **Require valid-user** la cual nos indica que solo está permitido acceder al recurso a los usuarios que se encuentren en la lista de usuarios definidos previamente en el archivo **htpasswd**.

```

<Directory "/var/www/sitio_dot.com">
    Options All
    AllowOverride
    <RequireAll>
        AuthName "Private"
        AuthType Basic
        AuthUserFile /etc/apache2/.htpasswd
        Require valid-user
    </RequireAll>
</Directory>

```

2. Configuración de puerto de operación

La configuración de puertos de operación define en qué puertos se estará dando el servicio HTTP. Cómo se vio en la configuración de los hosts, se indicaba en qué puerto se mantendrían escuchando.

Para habilitar puertos de operación, tenemos que modificar el archivo `/etc/apache2/ports.conf`. Utilizando la directiva `Listen PORT`, indicamos que puertos debe escuchar el servidor. En este caso agregamos `Listen 80`, para indicar que mediante el puerto 80 se realizará el servicio. Otro puerto que utilizaremos para el

siguiente punto en la configuración es el puerto 443, para agregarlo debemos escribir en el archivo la directiva Listen 443.

```
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
~
```

3. Servidor de aplicación utilizando el protocolo HTTPS

HTTPS (Protocolo Seguro de de Transferencia de Hyper Texto) es la versión segura de HTTP. La letra S al final de HTTP representa “seguridad”, al implementar este protocolo, significa que todas nuestras comunicaciones entre en navegador (cliente) y el sitio web (servidor) están cifradas. [3]

1. Definición de certificados //llaves de operación. /Certificados auto firmados.

Para añadir un certificado HTTPS a un sitio dentro de nuestro servidor, primero crearemos dos archivos: .cert y .key.

Para crear los archivos utilizamos el siguiente comando:

```
root@fercg: /home/fernanda 231x52
~ - sudo su
[sudo] password for fernanda:
root@fercg:/home/fernanda# openssl req -new -x509 -days 365 -keyout vhost1.key -out vhost1.crt -nodes -subj '/O=Company/OU=Department/CN=www.sitio_uno.com'
Generating a 2048 bit RSA private key
.....+++
writing new private key to 'vhost1.key'
-----
root@fercg:/home/fernanda#
```

Podemos verificar que los archivos creados estén dentro de nuestra carpeta:



Finalmente, debemos de volver a configurar el archivo dentro de **/etc/apache2/sites-available/sitio_uno.com.conf** en donde anteriormente habíamos declarado los hosts.

Agregamos las directivas **RewriteEngine**, **RewriteCond** y **RewriteRule** en el host que teníamos descrito, en este ejemplo el **sitio_tres.com**; posteriormente agregamos otro virtual host que configurará también el sitio_tres, pero en esta ocasión se estará escuchando en el puerto 443, y utilizaremos las directivas **SSLEngine**, **SSLCertificateFile** y **SSLCertificateKeyFile** para indicar que usaremos el protocolo HTTP.

Recordemos también que debemos de agregar el puerto 443 en el archivo **/etc/apache2/ports.conf**, como se describió anteriormente.

```
<VirtualHost *:80>
    ServerName sitio_tres.com
    ServerAlias www.sitio_tres.com
    ServerAdmin webmaster@sitio_tres.com
    DocumentRoot /var/www/sitio_tres.com
    CustomLog /var/log/apache2/sitio_tres.com/access.log common
    ErrorLog /var/log/apache2/sitio_tres.com/error.log
    RewriteEngine on
    RewriteCond %{HTTPS} off
    RewriteRule (.*?) https://%{HTTP_HOST}%{REQUEST_URI} [R]
</VirtualHost>

<VirtualHost *:443>
    ServerName sitio_tres.com
    DocumentRoot /var/www/sitio_tres.com
    CustomLog /var/log/apache2/sitio_uno.com/access.log common
    ErrorLog /var/log/apache2/sitio_uno.com/error.log
    SSLEngine On
    SSLCertificateFile /etc/apache2/ssl/crt/vhost1.crt
    SSLCertificateKeyFile /etc/apache2/ssl/key/vhost1.key
</VirtualHost>
```

2. Certificados firmados por un tercero (Autoridad certificadora)

Los certificados proporcionados por terceros, nos garantizan algunas características a la hora de tener conexión en una red, brindan un alto nivel de seguridad. Los certificados de terceros tienen generalmente un costo para poder utilizarlos. También existen patrocinios que dadas

ciertas condiciones te permiten utilizar sus certificados de manera gratuita.

4. Personalización de páginas de error para todos los sitios

En ocasiones pueden ocurrir ciertos errores al momento de la interacción de cliente y el servidor, por ejemplo, puede ocurrir que el cliente solicite un archivo que no existe en el servidor, o tratar de ingresar a un sitio el cual el cliente no tiene permisos para acceder.

Apache cuenta con mensajes por default el cual nos informa que error acaba de suceder. En ocasiones, probablemente querremos identificar estos errores con nuestros propios mensajes personalizados. Para personalizar nuestros propios mensaje, podemos configurar el archivo **/etc/apache2/sites-available/sitio_uno.com.conf**, utilizando la directiva **ErrorDocument COD Mensaje_De_Error**.

A continuación mostramos un ejemplo de como configurarlo, y cómo sería la vista del cliente cuando le suceden los errores.

```
<VirtualHost *:80>
ServerName sitio_dos.com
ServerAlias www.sitio_dos.com
ServerAdmin webmaster@sitio_dos.com
DocumentRoot /var/www/sitio_dos.com
CustomLog /var/log/apache2/sitio_dos.com/access.log common
ErrorLog /var/log/apache2/sitio_dos.com/error.log

ErrorDocument 403 "403: Alguien no te quiere :)"
ErrorDocument 404 "404: Se fue por los cigarras y nunca regreso."
ErrorDocument 401 "401: Ke paso amiguito?, intenta con password o 12345"
</VirtualHost>
```

Pantalla de error 401



Pantalla error 403



Pantalla error 404



Operación

Después de haber configurado nuestro servidor, es importante que cuando esté activo el servicio, podamos observar el funcionamiento de lo que pasa dentro de las conexiones que se establecen con sus clientes.

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

Las bitácoras, son archivos logs que nos permiten ver información útil de las conexiones y peticiones que nuestros clientes realizan dependiendo el sitio que deseemos revisar.

Para la creación de estos logs, primero debemos, crearemos formatos de logs. Apache nos brinda varias opciones de banderas que podemos utilizar para agregar información que queramos ver en los archivos logs, y utilizando la directiva **LogFormat** **banderas nombre_Del_Formato** podemos configurar diferentes estilos de formatos. Los estilos deben ser agregados en el archivo **/etc/apache2/apache2.conf** como se muestra a continuación.

```
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %O" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.
```

Una vez agregados los formatos en el archivo de configuración de Apache, procedemos a indicar en el archivo de configuración del sitio en donde queremos agregar nuestro nuevo formato de log para que aparezcan los registros, en este caso modificaremos el archivo **/etc/apache2/sites-available/sitio_uno.com.conf**, agregando la siguiente directiva para indicar que solicitamos un log con el formato que acabamos de agregar: **CustomLog ruta_Donde_Guardar_Log formato_A_Usar**.

```

VirtualHost *:80>
ServerName sitio_uno.com
ServerAlias www.sitio_uno.com
ServerAdmin webmaster@sitio_uno.com
DocumentRoot /var/www/sitio_uno.com
CustomLog /var/log/apache2/sitio_uno.com/access.log common
CustomLog /var/log/apache2/sitio_uno.com/ny_log.log combined
ErrorLog /var/log/apache2/sitio_uno.com/error.log

RewriteEngine on
RewriteCond %{HTTPS} off
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI} [R]

ErrorDocument 401 "401: ¿Ke paso amiguito?, intenta con password o 12345"
ErrorDocument 403 "403: Alguien no te quiere :)"
ErrorDocument 404 "404: Se fue por los cigarros y nunca regreso."

</VirtualHost>

```

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

Finalmente, nos puede interesar conocer no solo información de las conexiones, también nos puede importar cuantas peticiones ha tenido nuestro sitio, el estado del sistema y cuántos recursos se han ocupado.

Apache nos brinda una útil herramienta, la cual nos muestra toda esta información relevante al momento de analizar el estado del sistema. Para ingresar a la información, basta con escribir desde nuestro navegador, estando del lado del servidor, **sitio.com/server-status**.

En el siguiente ejemplo mostramos el estatus del **sitio_tres.com** que tenemos activo.

Apache Status - Mozilla Firefox

http://sitio_tres.com/server-status

Apache Server Status for sitio_tres.com (via 127.0.1.1)

Server Version: Apache/2.4.7 (Ubuntu) OpenSSL/1.0.1f
 Server MPM: event
 Server Built: Sep 18 2017 16:37:54

Current Time: Thursday, 08-Mar-2018 19:33:42 CST
 Restart Time: Thursday, 08-Mar-2018 18:43:12 CST
 Parent Server Config. Generation: 2
 Parent Server MPM Generation: 1
 Server uptime: 50 minutes 29 seconds
 Server load: 0.38 0.26 0.23
 Total accesses: 88 - Total Traffic: 328 kB
 CPU Usage: u.65 s.7 cu0 cs0 - .0446% CPU load
 .0291 requests/sec - 110 B/second - 3816 B/request
 1 requests currently being processed, 49 idle workers

PID	Connections			Threads			Async connections		
	total	accepting	busy	idle	writing	keep-alive	closing		
215390	yes	1	24	0	0	0	0		
215400	yes	0	25	0	0	0	0		
Sum	0	1	49	0	0	0	0		

.....w.....

Scoreboard Key:
 " " Waiting for Connection, "s" Starting up, "r" Reading Request,
 "w" Sending Reply, "k" Keepalive (read), "b" DNS Lookup,
 "c" Closing connection, "L" Logging, "G" Gracefully finishing,
 "I" Idle cleanup of worker, "." Open slot with no current process

.....

Score	PID	Age	MPM	SS	Req	Conn	Child	Slot	Client	URI	Host	Request
.....

Apache Status - Mozilla Firefox

http://sitio... apache2 Firma de ab... core - Ap... Getting... 3 notific... Mi unida... Servido... 403 Forbidc... Apache S...

sitio_tres.com/server-status

Scoreboard Key:
 "W" Waiting for Connection, "S" Starting up, "R" Reading Request,
 "w" Sending Reply, "k" Keepalive (read), "D" DNS Lookup,
 "C" Closing connection, "L" Logging, "G" Gracefully finishing,
 "I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	VHost	Request
0-1	21539	0/3/3	0.64	253	0	0.0	0.02	0.02	0.02	192.168.43.21		
0-1	21539	0/3/3	0.64	231	0	0.0	0.00	0.00	0.00	192.168.43.21	sitio_dos.com:80	GET /favicon.ico HTTP/1.1
0-1	21539	0/3/3	0.66	119	0	0.0	0.00	0.00	0.00	127.0.0.1	sitio_dos.com:80	GET / HTTP/1.1
0-1	21539	0/3/3	0.68	54	0	0.0	0.00	0.00	0.00	127.0.0.1		
0-1	21539	0/3/3	0.67	74	0	0.0	0.00	0.00	0.00	127.0.0.1	sitio_uno.com:80	GET / HTTP/1.1
0-1	21539	0/2/2	0.68	29	0	0.0	0.00	0.00	0.00	127.0.0.1	sitio_tres.com:80	GET / HTTP/1.1
0-1	21539	0/2/2	0.69	29	0	0.0	0.00	0.00	0.00	127.0.0.1	sitio_tres.com:80	GET /css/style.css HTTP/1.1
0-1	21539	0/3/3	0.69	29	0	0.0	0.00	0.00	0.00	127.0.0.1	sitio_tres.com:80	GET /images/logo.png HTTP/1.1
0-1	21539	0/2/2	0.69	29	0	0.0	0.00	0.00	0.00	127.0.0.1	sitio_tres.com:80	GET /images/family-large.jpg HTTP/1.1
0-1	21539	0/3/3	0.69	29	0	0.0	0.00	0.00	0.00	127.0.0.1	sitio_tres.com:80	GET /images/smile.jpg HTTP/1.1
0-1	21539	0/3/3	0.69	29	0	0.0	0.00	0.00	0.00	127.0.0.1	sitio_tres.com:80	GET /images/meeting.jpg HTTP/1.1
0-1	21539	0/2/2	0.69	29	0	0.0	0.00	0.00	0.00	127.0.0.1	sitio_tres.com:80	GET /images/bg-highlight.jpg HTTP/1.1
0-1	21539	0/1/1	0.69	29	0	0.0	0.00	0.00	0.00	127.0.0.1	sitio_tres.com:80	GET /images/bg-adbox.png HTTP/1.1
0-1	21539	0/2/2	0.69	29	0	0.0	0.00	0.00	0.00	127.0.0.1	sitio_tres.com:80	GET /images/bg-header.jpg HTTP/1.1
0-1	21539	0/3/3	0.69	29	0	0.0	0.00	0.00	0.00	127.0.0.1	sitio_tres.com:80	GET /images/border.png HTTP/1.1
0-1	21539	0/2/2	0.69	29	0	0.0	0.00	0.00	0.00	127.0.0.1	sitio_tres.com:80	GET /images/frames.png HTTP/1.1
0-1	21539	0/3/3	0.69	29	0	0.0	0.00	0.00	0.00	127.0.0.1	sitio_tres.com:80	GET /images/bg-footer.png HTTP/1.1
0-1	21539	0/2/2	0.69	29	0	0.0	0.02	0.02	0.02	127.0.0.1	sitio_tres.com:80	GET /images/bg-footnote.jpg HTTP/1.1
0-1	21539	0/1/1	0.64	16	0	0.0	0.00	0.00	0.00	127.0.0.1		
0-1	21539	0/2/2	W 0.64	0	0	0.0	0.01	0.01	0.01	127.0.0.1	sitio_tres.com:80	GET /server-status HTTP/1.1
0-1	21539	0/2/2	0.64	254	0	0.0	0.02	0.02	0.02	192.168.43.21		

Bibliografia

- [1] The Apache HTTP Server Project, *Apache HTTP Server Version 2.4 Documentation*,
<https://httpd.apache.org/docs/2.4/>
- [2] THe Apache HTTP Server Project, *Apache Virtual Hosts Documentation*,
<http://httpd.apache.org/docs/2.4/vhosts/>
- [3] COMODO Instant SSL, What is HTTPs?,
<https://www.instantssl.com/ssl-certificate-products/https.html>