

Instituto Politécnico Nacional

Escuela Superior de Cómputo

Administración de Servicios en Red

Equipo 9

**Servidor de aplicación utilizando el protocolo HTTP: manual de
instalación/configuración/operación de servidor.**

Rocío Vivia Delgadillo
Bryan Domínguez de la Rosa
Fernando Jair Pacheco Díaz

Grupo 4CV3

Profesor: Manuel Alejandro Soto Ramos



Índice

Protocolo HTTP -----	3
Servidor Apache -----	3
Instalación -----	3
Configuración -----	3
Operación -----	5
Contenedor virtual por IP y por dominio -----	5
Configuración de puerto de operación-----	6
Servidor de aplicación utilizando el protocolo HTTPS-----	8
Personalización de páginas de error para todos los sitios-----	13
Configuración de archivos de bitácoras y mensajes de error-----	14
Resumen de operación de forma dinámica -----	15
Referencias -----	18

Protocolo HTTP

El Protocolo de Transferencia de Hipertexto (Hypertext Transfer Protocol) es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP. La especificación completa del protocolo HTTP 1/0 está recogida en el RFC 1945.

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).

Servidor Apache HTTP versión 2.4

El Proyecto Apache HTTP Server es un esfuerzo para desarrollar y mantener un servidor HTTP de código abierto para los sistemas operativos modernos, incluidos UNIX y Windows. El objetivo de este proyecto es proporcionar un servidor seguro, eficiente y extensible que proporcione servicios HTTP en sincronización con los estándares HTTP actuales.

El Servidor Apache HTTP ("httpd") se lanzó en 1995 y ha sido el servidor web más popular en Internet desde abril de 1996. En febrero de 2015, celebró su vigésimo aniversario como proyecto.

Instalación en Ubuntu 16.04 LTS

Actualizamos e instalamos el servidor escribiendo las siguientes líneas en consola:

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

Una vez instalado el servidor, iniciamos el servicio introduciendo la siguiente línea en consola:

```
$sudo service apache2 start
```

Verificamos que el servicio haya iniciado realmente con la siguiente línea:

```
$sudo systemctl status apache2
```

Nota: Para detener el servicio basta con cambiar "start" por "stop". Existen otras opciones como: reload, restart, status.

Configuración

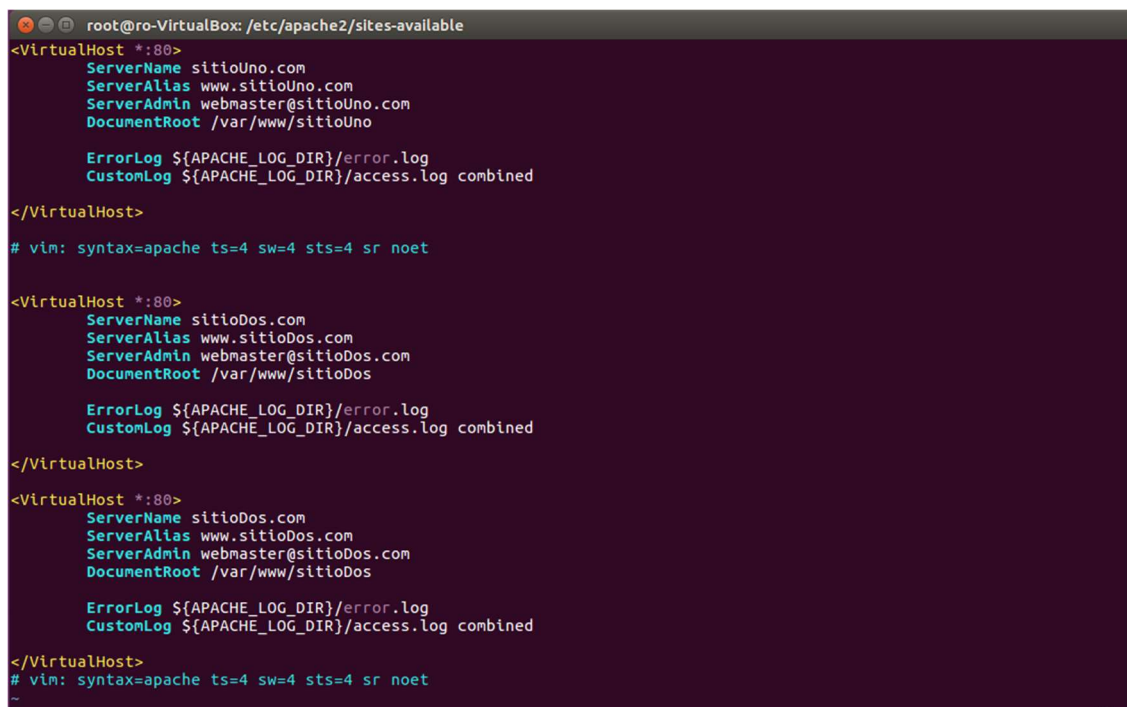
Con permisos de superusuario, creamos las carpetas contenedoras de la información que los sitios web requieren:

```
$sudo su
#mkdir /var/www/sitioUno
```

Lo siguiente será configurar los VirtualHost, para ello, nos movemos a la carpeta /etc/apache2/sites-available/ y crearemos el archivo de configuración para los hosts, copiaremos el que tenemos por default:

```
#cd /etc/apache2/sites-available/
#cp 000-default.conf sitioUno.conf
```

Una vez copiado el archivo .conf, podemos modificar el nombre, alias, tenemos que proporcionar la ruta de los sitios y podemos también cambiar el correo electrónico:



```
root@ro-VirtualBox: /etc/apache2/sites-available
<VirtualHost *:80>
    ServerName sitioUno.com
    ServerAlias www.sitioUno.com
    ServerAdmin webmaster@sitioUno.com
    DocumentRoot /var/www/sitioUno

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

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

<VirtualHost *:80>
    ServerName sitioDos.com
    ServerAlias www.sitioDos.com
    ServerAdmin webmaster@sitioDos.com
    DocumentRoot /var/www/sitioDos

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

<VirtualHost *:80>
    ServerName sitioDos.com
    ServerAlias www.sitioDos.com
    ServerAdmin webmaster@sitioDos.com
    DocumentRoot /var/www/sitioDos

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

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

Agregamos las direcciones de los hosts con sus respectivos nombres de dominio en el archivo /etc/hosts de la siguiente manera:

```
root@ro-VirtualBox: /etc/apache2/sites-available
127.0.0.1 localhost
127.0.1.1 ro-VirtualBox
127.0.1.1 sitioUno.com
127.0.1.1 sitioDos.com
127.0.1.1 sitioTres.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
~
~
~
```

Por último, deshabilitamos el archivo host por default y habilitamos el nuevo:

```
#a2ensite sitioUno.conf
#a2dissite 000-default.conf
#service apache2 reload
```

Operación

1. Contenedor virtual por IP y por dominio

1. Restringir acceso al recurso por dirección IP del cliente: entramos al archivo /etc/apache2/apache2.conf y agregamos las directivas.

```
<Directory (ruta del directorio)>
```

```
Options All
AllowOverride All
<RequireAll>
    Require all granted
    Require ip x.x.x.x #IP a la que se le niega acceso.
</RequireAll>
</Directory>
```



2. Restringir acceso al recurso por segmento de red: entramos al archivo `/etc/apache2/apache2.conf` y agregamos las directivas:

```
<Directory (ruta del directorio)>
    Options All
    AllowOverride All
    <RequireAll>
        Require all granted
        Require ip x.x.x.x x.x.x.x #Segmento de red.
    </RequireAll>
</Directory>
```

3. Restringir acceso al recurso por nombre de usuario (grupo de usuarios) /clave de acceso: aquí definiremos un grupo de usuarios que tienen los permisos para acceder al sitio web, para ello primero necesitaremos crear el archivo `/etc/apache2/htpasswd`:

```
# htpasswd -c /etc/apache2/.htpasswd "usuario"
```

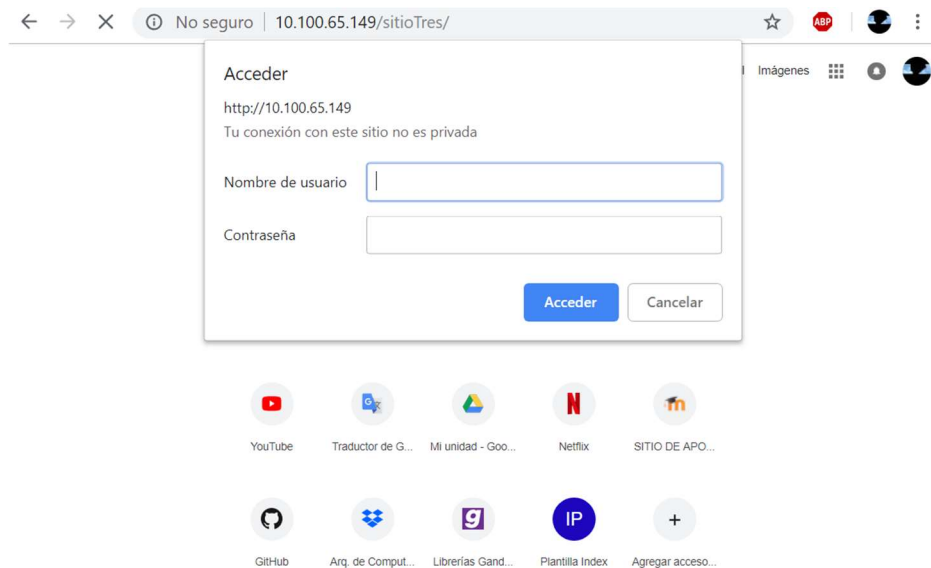
Y a continuación nos pedirá ingresar la contraseña. Para consultar la lista de usuarios basta con acceder al archivo `.htpasswd` y también podemos observar que las respectivas contraseñas aparecen de forma cifrada.

Una vez creado el archivo de usuarios, procedemos a restringir el recurso por nombre de usuario:

```
<Directory (ruta del directorio)>
    Options All
    AllowOverride All
    <RequireAll>
        AuthName "Private"
        AuthType Basic
```

```
AuthUserFile /etc/apache2/.htpasswd
Require valid user
</RequireAll>
</Directory>
```

Utilizamos **AuthType** Basic indicando que el método de autenticación es básico, **AuthFile** para indicar la ruta del archivo que define los usuarios válidos.



2. Configuración de puerto de operación

La configuración de los puertos se realiza en el archivo `/etc/apache2/ports.conf`, el puerto que se usa por defecto es el 80, pero para agregar un basta con entrar al archivo y agregar la siguiente línea:

```
# vim /etc/apache2/ports.conf
```

```
Listen 80
Listen puertoNuevo
```

```
ro@ro-VirtualBox: /
# 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
Listen 8080

<IfModule ssl_module>
    Listen 443
</IfModule>

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

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

Lo siguiente sería agregar en los VirtualHost el puerto nuevo:

```
<VirtualHost *:80 *:puertoNuevo>
```

...

```
</VirtualHost
```

```
ro@ro-VirtualBox: /

</VirtualHost>
<VirtualHost *:80 *:8080>
    ServerName sitioUno.com
    ServerAlias www.sitioUno.com
    ServerAdmin webmaster@sitioUno.com
    DocumentRoot /var/www/sitioUno

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

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

<VirtualHost *:80>
    ServerName sitioDos.com
    ServerAlias www.sitioDos.com
    ServerAdmin webmaster@sitioDos.com
    DocumentRoot /var/www/sitioDos
-- INSERT --
```

Por último, reiniciamos el servidor:

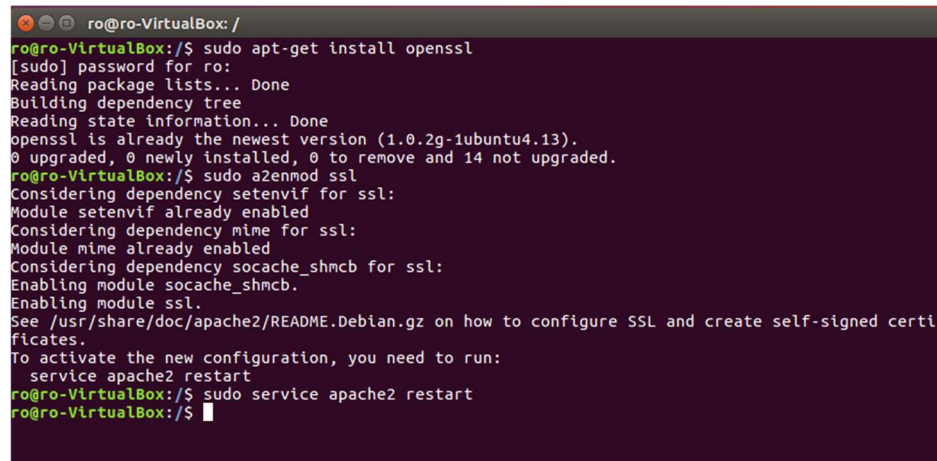
```
#service apache2 restart
```


3. Servidor de aplicación utilizando el protocolo HTTPS

1. Definición de certificados /llaves de operación

Necesitamos tener instalado openssl, un Ubuntu 16.04 LTS lo está, para confirmar podemos intentar instalarlo con el siguiente comando:

```
$sudo apt-get install openssl
```



```
ro@ro-VirtualBox: /  
ro@ro-VirtualBox:/$ sudo apt-get install openssl  
[sudo] password for ro:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
openssl is already the newest version (1.0.2g-1ubuntu4.13).  
0 upgraded, 0 newly installed, 0 to remove and 14 not upgraded.  
ro@ro-VirtualBox:/$ sudo a2enmod ssl  
Considering dependency setenvif for ssl:  
Module setenvif already enabled  
Considering dependency mime for ssl:  
Module mime already enabled  
Considering dependency socache_shmcb for ssl:  
Enabling module socache_shmcb.  
Enabling module ssl.  
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.  
To activate the new configuration, you need to run:  
    service apache2 restart  
ro@ro-VirtualBox:/$ sudo service apache2 restart  
ro@ro-VirtualBox:/$
```

Habilitamos el modulo introduciendo la siguiente línea en la terminal y reiniciamos el servidor:

```
$sudo a2enmod ssl  
$sudo service apache2 restart
```

Lo siguiente es crear una carpeta destino para los certificados que crearemos en un momento

```
$sudo mkdir /etc/apache2/ssl
```

Creamos el archivo de certificado y la clave, para ello introducimos la siguiente línea de comando:

```
$sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/clave.key -out /etc/apache2/ssl/certificado.crt
```

```

ro@ro-VirtualBox: /
ro@ro-VirtualBox:/$ sudo service apache2 restart
ro@ro-VirtualBox:/$ sudo mkdir /etc/apache2/ssl
ro@ro-VirtualBox:/$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache
2/ssl/clave.key -out /etc/apache2/ssl/certificado.crt
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to '/etc/apache2/ssl/clave.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) []:Iztacalco
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ESCOM
Organizational Unit Name (eg, section) []:Certificados
Common Name (e.g. server FQDN or YOUR name) []:sitioTres.com
Email Address []:webadmin@sitioTres.com
ro@ro-VirtualBox:/$

```

Una vez introduciendo la información requerida, ahora configuramos apache para hacer uso de los certificados ya creados, para ello nos ubicamos en el archivo `/etc/apache2/sites-available/sitioUno.conf` y agregamos la siguiente información:

```

CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin webmaster@sitioTres.com
        ServerName sitioTres.com
        ServerAlias www.sitioTres.com
        DocumentRoot /var/www/sitioTres.com

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        SSLEngine on
        SSLCertificateFile /etc/apache2/ssl/certificado.crt
        SSLCertificateKeyFile /etc/apache2/ssl/clave.key

        <FilesMatch "\.(cgi|shtml|phtml|php)$">
            SSLOptions +StdEnvVars
        </FilesMatch>
        <Directory /usr/lib/cgi-bin>
            SSLOptions +StdEnvVars
        </Directory>

        BrowserMatch "MSIE [2-6]" \
            nokeepalive ssl-unclean-shutdown \
            downgrade-1.0 force-response-1.0

        BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown

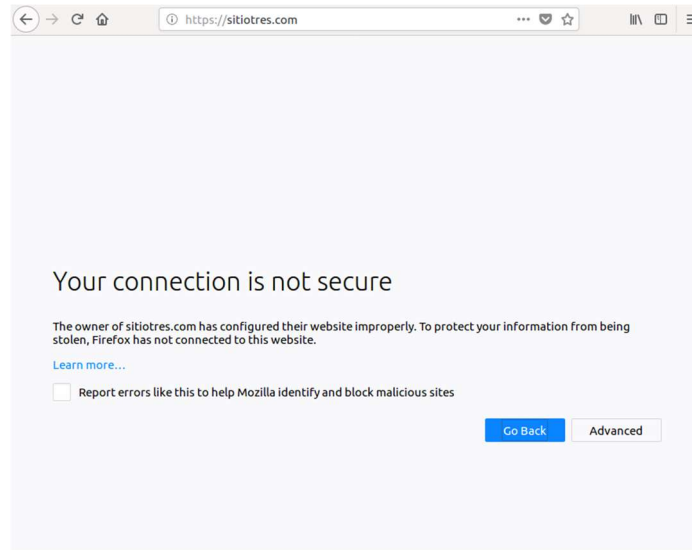
    </VirtualHost>
</IfModule>

```

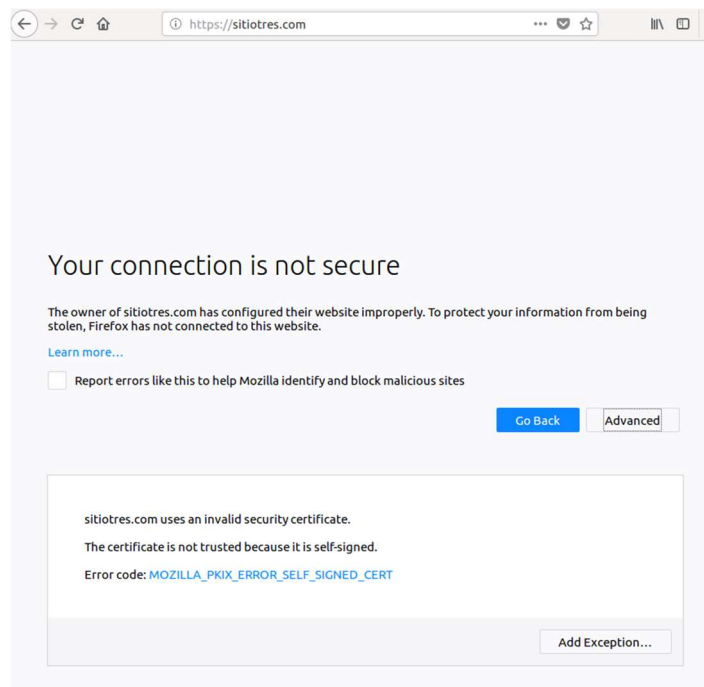
Y recargamos nuevamente el servidor

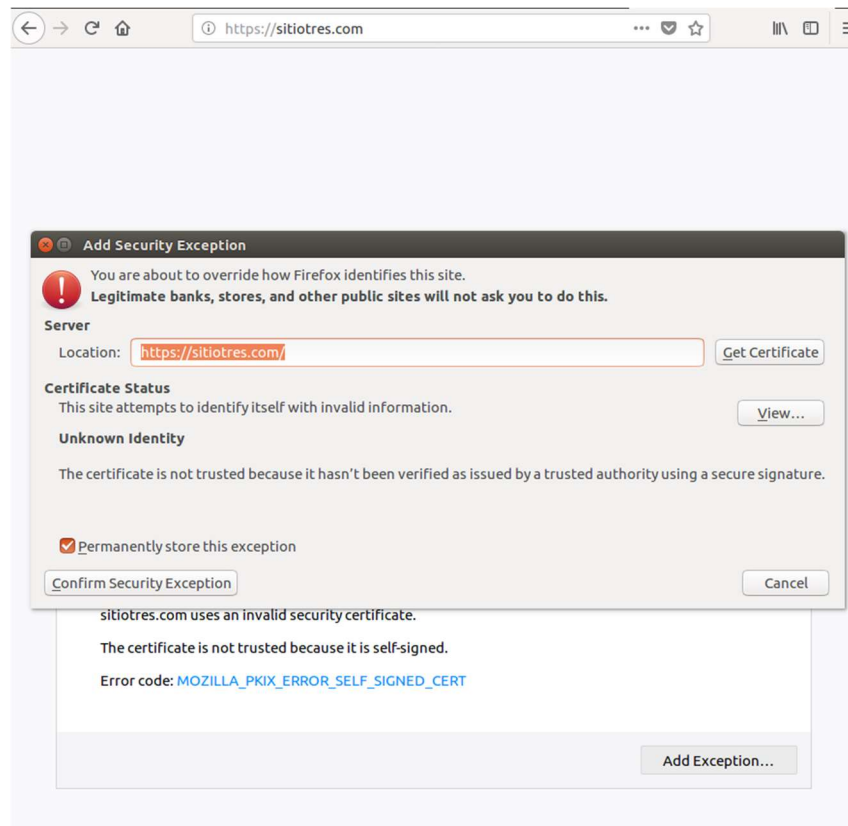
```
$sudo service apache2 reload
```

Para finalizar, abrimos una ventana de navegación y escribimos en la barra de direcciones `https://sitioCertificado.com`

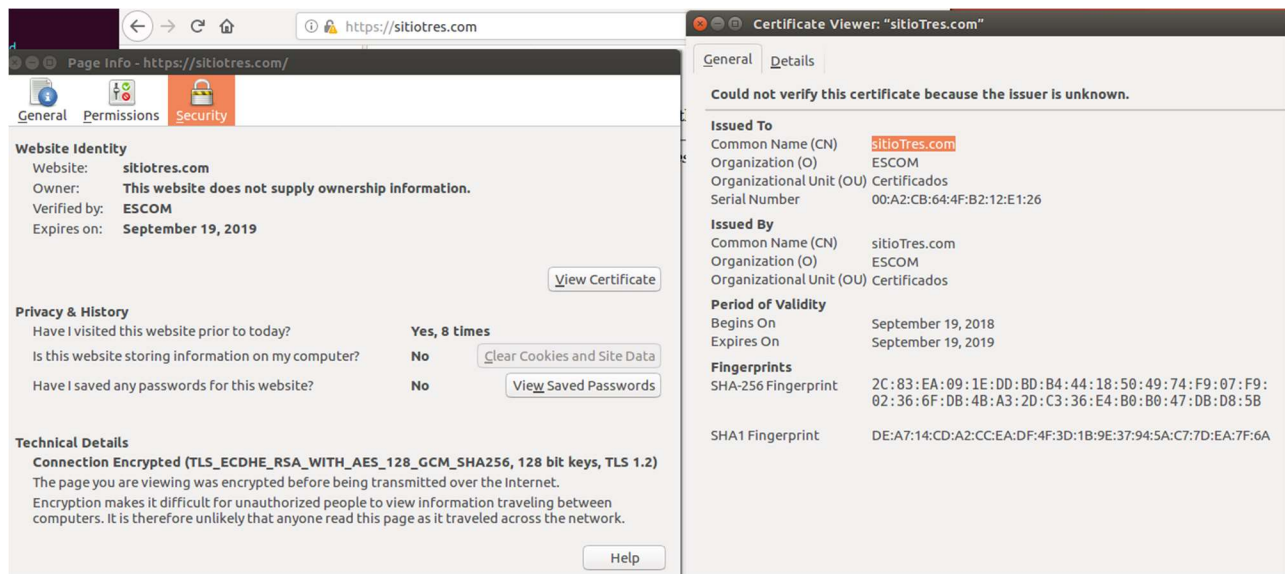


Nos avisará que nuestra conexión no es segura, para aceptar el certificado que acabamos de crear, seleccionamos el botón "Advanced" y después "Add Exception"





Confirmamos la excepción de seguridad y listo, ahora podemos ver que todo lo que hagamos dentro del sitio estará cifrado.



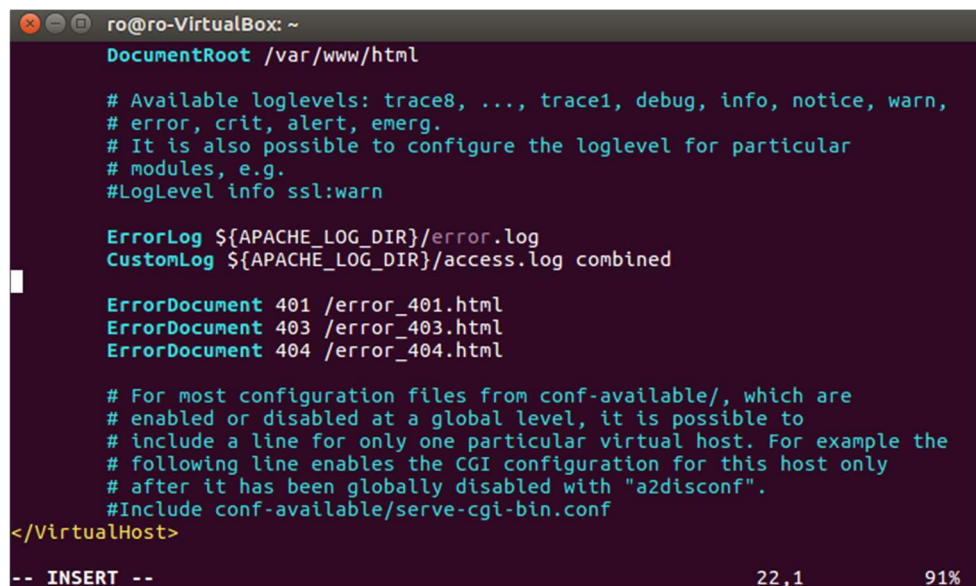
Los certificados proporcionados por terceros garantizan algunas características a la hora de tener conexión en una red, brindan un alto nivel de seguridad. Los certificados de terceros tienen generalmente un costo para poder utilizarlos, aunque existen patrocinios que dadas ciertas condiciones te permiten utilizar sus certificados de manera gratuita.

4. Personalización de páginas de error para todos los sitios

1. Configurar al menos 3 Diferentes tipos de errores del sitio

Los html de los errores configurados deben estar en la carpeta /var/www/html, lo siguiente es modificar el archivo de configuración para agregar los errores personalizados, para ello agregamos la siguiente línea en el archivo conf por default /etc/apache2/sites-available/000-default.conf :

```
ErrorDocument xxx /nombreArchivoError.html
```



```
ro@ro-VirtualBox: ~
DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

ErrorDocument 401 /error_401.html
ErrorDocument 403 /error_403.html
ErrorDocument 404 /error_404.html

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>

-- INSERT --                22,1          91%
```



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

Apache tiene varias opciones de bandera que podemos utilizar para obtener información que nos interese ver como administradores, dicha configuración se realiza en el archivo `/etc/apache2/apache2.conf`:

```
#vim /etc/apache2/apache2.conf
#LogFormat banderas formatoSeleccionado
```

```
ro@ro-VirtualBox: /
#
# These deviate from the Common Log Format definitions in that they use %O
# (the actual bytes sent including headers) instead of %b (the size of the
# requested file), because the latter makes it impossible to detect partial
# requests.
#
# Note that the use of %{X-Forwarded-For}i instead of %h is not recommended.
# Use mod_remoteip instead.
#
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined
LogFormat "%h %l %u %t \"%r\" %>s %O \"%{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.

# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf

# Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf
```

Hecho esto, procedemos a indicar en el archivo `.conf` el nuevo formato de log:

```
CustomLog rutaParaGuardarLog formatoSeleciconado
```



```
ro@ro-VirtualBox: /

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
CustomLog /var/log/apache2/sitios/access.log vhost_combined
CustomLog /var/log/apache2/my_log.log common
ErrorDocument 401 /error_401.html
ErrorDocument 403 /error_403.html
ErrorDocument 404 /error_404.html

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>

-- INSERT --
```

```
#service apache2 reload
```

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

1. Mostrar resumen de conexiones
2. Mostrar resumen de consumo de recursos (Memoria/procesador/tiempo de ejecución)

Apache tiene una herramienta para mostrar el estado del servidor para cada sitio, podemos acceder a este recurso simplemente abriendo una nueva ventana de navegación y escribiendo en la barra de direcciones `nombredeSitio/server-status`:

```
Current Time: Wednesday, 19-Sep-2018 12:23:10 CDT
Restart Time: Wednesday, 19-Sep-2018 11:00:46 CDT
Parent Server Config. Generation: 4
Parent Server MPM Generation: 3
Server uptime: 1 hour 22 minutes 23 seconds
Server load: 0.65 0.56 0.30
Total accesses: 37 - Total Traffic: 48 kB
CPU Usage: u.49 s.45 cu0 cs0 - .019% CPU load
.00749 requests/sec - 9 B/second - 1328 B/request
1 requests currently being processed, 49 idle workers
```

PID	Connections		Threads		Async connections		
	total	accepting	busy	idle	writing	keep-alive	closing
2956	0	yes	1	24	0	0	0
2927	0	yes	0	25	0	0	0
Sum	0		1	49	0	0	0

Scoreboard Key:
 " " Waiting for Connection, "s" Starting up, "R" Reading Request,
 "W" Sending Reply, "k" Keepalive (read), "n" 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
0-3	2956	0/1/2	0.44	36	0	0.0	0.00	0.00	0.00	127.0.0.1	127.0.1.1:80	GET /favicon.ico HTTP/1.1
0-3	2956	0/0/1	0.12	843	0	0.0	0.00	0.00	0.00	192.168.1.81		
0-3	2956	0/1/1	0.03	989	0	0.0	0.00	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioTres/css/component.css HTTP/1.1
0-3	2956	0/1/1	0.03	989	0	0.0	0.00	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioTres/css/default.css HTTP/1.1
0-3	2956	0/1/1	0.06	939	1	0.0	0.00	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioDos/ HTTP/1.1
0-3	2956	0/1/1	0.06	939	1	0.0	0.00	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioDos/p2.css HTTP/1.1

Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	VHost	Request
0-3	2956	0/1/2	_	0.44	36	0	0.0	0.00	0.00	127.0.0.1	127.0.1.1:80	GET /favicon.ico HTTP/1.1
0-3	2956	0/0/1	_	0.12	843	0	0.0	0.00	0.00	192.168.1.81		
0-3	2956	0/1/1	_	0.03	989	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioTres/css/component.css HTTP/1.1
0-3	2956	0/1/1	_	0.03	989	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioTres/css/default.css HTTP/1.1
0-3	2956	0/1/1	_	0.06	939	1	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioDos/ HTTP/1.1
0-3	2956	0/1/1	_	0.06	939	1	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioDos/p2.css HTTP/1.1
0-3	2956	0/1/1	_	0.44	36	0	0.0	0.00	0.00	127.0.0.1	127.0.1.1:80	GET /service-status HTTP/1.1
0-3	2956	0/0/0	W	0.00	0	0	0.0	0.00	0.00	127.0.0.1	127.0.1.1:80	GET /server-status HTTP/1.1
1-1	-	0/0/1	_	0.14	1165	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioDos/ HTTP/1.1
1-1	-	0/0/1	_	0.13	1165	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioUno/ HTTP/1.1
1-1	-	0/0/1	_	0.04	1165	1	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioTres/ HTTP/1.1
1-1	-	0/0/1	_	0.04	1165	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /favicon.ico HTTP/1.1
1-1	-	0/0/1	_	0.12	1165	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioUno/ HTTP/1.1
1-1	-	0/0/1	_	0.13	1165	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /favicon.ico HTTP/1.1
1-1	-	0/0/1	_	0.13	1165	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /favicon.ico HTTP/1.1
1-1	-	0/0/1	_	0.14	1165	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioDos/ HTTP/1.1
1-1	-	0/0/1	_	0.14	1165	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /favicon.ico HTTP/1.1
1-1	-	0/0/1	_	0.14	1165	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioDos/ HTTP/1.1
1-1	-	0/0/1	_	0.14	1165	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /favicon.ico HTTP/1.1
1-1	-	0/0/1	_	0.14	1165	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioDos/ HTTP/1.1
1-1	-	0/0/1	_	0.14	1165	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /favicon.ico HTTP/1.1
1-1	-	0/0/1	_	0.14	1165	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioDos/ HTTP/1.1
1-1	-	0/0/1	_	0.14	1165	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioDos/ HTTP/1.1
1-1	-	0/0/1	_	0.14	1165	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /favicon.ico HTTP/1.1
2-2	-	0/0/1	_	0.01	1060	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioUno/ HTTP/1.1
2-2	-	0/0/1	_	0.02	1060	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioTres/ HTTP/1.1
3-3	2927	0/1/1	_	0.00	1060	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioTres/css/component.css HTTP/1.1
3-3	2927	0/1/2	_	0.00	1060	1	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioTres/CSS/estiloi.css HTTP/1.1
3-3	2927	0/1/2	_	0.00	1060	0	0.0	0.00	0.00	192.168.1.81	127.0.1.1:80	GET /sitioTres/css/default.css HTTP/1.1
3-3	2927	0/1/2	_	0.00	1060	4	0.0	0.00	0.01	192.168.1.81	127.0.1.1:80	GET /sitioTres/ HTTP/1.1
3-3	2927	0/1/1	_	0.00	1060	4	0.0	0.00	0.00	192.168.1.81		
3-3	2927	0/1/1	_	0.00	1060	2	0.0	0.00	0.00	192.168.1.81		
3-3	2927	0/1/1	_	0.34	252	4	0.0	0.00	0.00	127.0.0.1	sitioUno.com:80	GET /server-status HTTP/1.1
3-3	2927	0/1/1	_	0.34	252	0	0.0	0.00	0.00	127.0.0.1	sitioUno.com:80	GET /favicon.ico HTTP/1.1

7. Resumen de operación y de error del servidor

Para la generación de resúmenes se utilizó el lenguaje de programación python en su versión 2.7. Se eligió este lenguaje debido a la sencillez en que se puede realizar el manejo de ficheros (en este caso, los archivos log), además de que cuenta con diversos paquetes que permiten dar un formato presentable a los datos de salida.

Para instalar python en nuestro sistema operativo Ubuntu necesitamos ejecutar las siguientes líneas en la consola:

```
$sudo apt update
```

```
$sudo apt install python2.7 python-pip
```

```

bryanddlr@bryanddlr-HP-14-Notebook-PC:~$ sudo apt install python2.7 python-pip
[sudo] password for bryanddlr:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
python2.7 ya está en su versión más reciente (2.7.12-lubuntu0~16.04.3).
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-pkg-resources 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-pkg-resources
  python-setuptools python-wheel python2.7-dev
0 actualizados, 13 nuevos se instalarán, 0 para eliminar y 41 no actualizados.
Se necesita descargar 29.8 MB de archivos.
Se utilizarán 45.2 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]

```

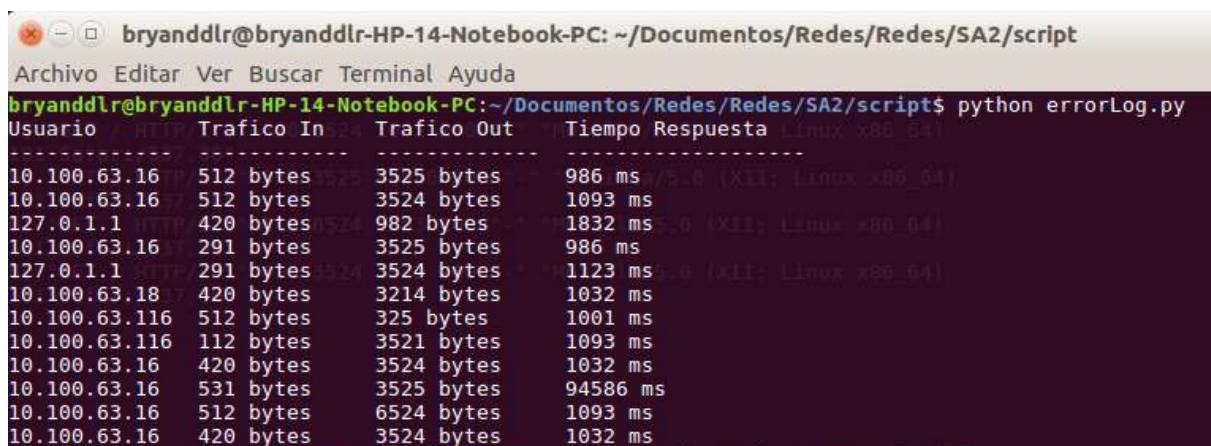
Una vez teniendo python en nuestro sistema operativo, necesitamos descargar un paquete que nos permitirá dar formato de tabla a nuestros datos de salida. El paquete mencionado se llama tabulate y se descarga con las siguientes líneas:

```
$pip install tabulate
```

Una vez instalado todo, ejecutamos los scripts.

Para el resumen de operación ejecutamos el script opLog.py con las líneas:

```
$python opLog.py
```



```

bryanddlr@bryanddlr-HP-14-Notebook-PC: ~/Documentos/Redes/Redes/SA2/script
Archivo Editar Ver Buscar Terminal Ayuda
bryanddlr@bryanddlr-HP-14-Notebook-PC:~/Documentos/Redes/Redes/SA2/script$ python errorLog.py
Usuario HTTP Trafico In Trafico Out Tiempo Respuesta Linux x86_64
-----
10.100.63.16 512 bytes 3525 bytes 986 ms
10.100.63.16 512 bytes 3524 bytes 1093 ms
127.0.1.1 420 bytes 982 bytes 1832 ms
10.100.63.16 291 bytes 3525 bytes 986 ms
127.0.1.1 291 bytes 3524 bytes 1123 ms
10.100.63.18 420 bytes 3214 bytes 1032 ms
10.100.63.116 512 bytes 325 bytes 1001 ms
10.100.63.116 112 bytes 3521 bytes 1093 ms
10.100.63.16 420 bytes 3524 bytes 1032 ms
10.100.63.16 531 bytes 3525 bytes 94586 ms
10.100.63.16 512 bytes 6524 bytes 1093 ms
10.100.63.16 420 bytes 3524 bytes 1032 ms

```

Para el resumen de errores ejecutamos el script opLog.py con las líneas:

```
$python errorLog.py
```

```

Archivo  Editar  Ver  Buscar  Terminal  Ayuda
bryanddlr@bryanddlr-HP-14-Notebook-PC:~/Documentos/Redes/Redes/SA2/script$ python readLog.py

```

Usuario	Recurso	Codigo	# Accesos
10.100.74.233	GET /	200	15
10.100.74.233	GET /icons/ubuntu-logo.png	200	15
10.100.74.233	GET /favicon.ico	404	15
10.100.74.233	GET /example.com	301	15
10.100.74.233	GET /example.com/	200	15
10.100.74.233	GET /secondexample.com/	200	15
10.100.74.233	GET /secondexample.com/	200	15
10.100.74.233	GET /favicon.ico	404	15
10.100.67.52	GET /secondexample.com	301	8
10.100.67.52	GET /secondexample.com/	200	8
10.100.67.52	GET /favicon.ico	404	8
10.100.70.182	GET /secondexample.com	301	4
10.100.70.182	GET /secondexample.com/	200	4
10.100.70.182	GET /favicon.ico	404	4
10.100.70.182	GET /favicon.ico	404	4
10.100.67.52	-	408	8
10.100.67.52	-	408	8
10.100.74.233	GET /secondexample.com/	200	15
10.100.74.233	GET /secondexample.com/	200	15
10.100.74.233	GET /secondexample.com/	200	15
10.100.74.233	GET /secondexample.com/	200	15
10.100.74.233	GET /secondexample.com/	200	15

Referencias

- [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>