

# Administración de servicios en red

Manual y servidor desarrollador por:

- Felipe Prieto de la Cruz
- Sanchez Ramírez Jonathan Leonardo
- Monroy García José Eduardo

## Configurar servidor en Ubuntu 18.04

### Índice

1. Requerimientos-----	1
2. Introducción-----	1-
3. Preparando el servidor -----	3-10
4. Habilitando servidor -----	13-14
5. Restricción de IP por segmento y usuario -----	14-19
6. Configurando puertos y certificados HTTPS -----	19-21
7. Personalización de páginas de error-----	23-24
8. Nivel de logs-----	24-25
9. Referencias-----	25

---

## Requerimientos

- Apache 2
  - Ubuntu 18.04
  - Conocimientos en editor nano, Vim o VI
  - OpenSSL 1.1.1
- 

# Introducción

Este manual está diseñado para poder configurar un servidor en Ubuntu 18.04 cumpliendo los siguientes requisitos:

- \* Contenedor virtual por IP y por dominio
- \* Restringir acceso al recurso por dirección IP del cliente
- \* Restringir acceso al recurso por segmento de red
- \* Restringir acceso al recurso por nombre de usuario (grupo de usuarios)/clave de acceso
- \* Configuración de puerto de operación
- \* Servidor de aplicación utilizando el protocolo HTTPS
- \* Definición de certificados /llaves de operación
- \* Certificados auto firmados
- \* Certificados firmados por un tercero (Autoridad certificadora)
- \* Personalización de páginas de error para todos los sitios
- \* Configurar al menos 3 Diferentes tipos de errores del sitio
- \* Configuración de archivos de bitácoras y mensajes de error.
- \* Resumen de operación de forma dinámica (Sitios, solicitudes, estado del sistema y recursos consumidos)
- \* Mostrar resumen de conexiones
- \* Mostrar resumen de consumo de recursos (Memoria/procesador/tiempo de ejecución)

## 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.

## Protocolo HTTPS

El Protocolo seguro de transferencia de hipertexto (en inglés: Hypertext Transfer Protocol Secure o HTTPS), es un protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de Hipertexto, es decir, es la versión segura de HTTP.

## OpenSSL

OpenSSL es un proyecto de software libre basado en SSLeay, desarrollado por Eric Young y Tim Hudson. Consiste en un robusto paquete de herramientas de administración y bibliotecas relacionadas con la criptografía, que suministran funciones criptográficas a otros paquetes como OpenSSH y navegadores web

## Virtual Host

La unidad básica que describe a un sitio o dominio es denominada virtual host (ó alojamiento virtual en español). Esta designación permite al administrador hacer uso de un servidor para alojar múltiples dominios o sitios en una única interfaz o IP utilizando un mecanismo de coincidencias.

---

# Administrador en la consola de Ubuntu

Antes de empezar es necesario ser root en la terminal, si estás usando Live CD de Ubuntu el comando para ser root es:

```
sudo -i
```

## 1 Parte la configuración del servidor

Lo primero va a ser instalar apache.

```
sudo apt-get install apache2 <-- Este comando nos sirve p  
ara ver si tiene instalados los repositorios.
```

## Iniciamos el servidor.

```
"service apache2 start" nos va a servir para iniciar el ser  
vicio
```

```
root@leonardo: /home/leonardo
Archivo Editar Ver Buscar Terminal Ayuda
root@leonardo:/home/leonardo# sudo apt-get update
Des:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88,7 kB]
Obj:2 http://es.archive.ubuntu.com/ubuntu bionic InRelease
Des:3 http://es.archive.ubuntu.com/ubuntu bionic-updates InRelease [88,7 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu bionic-backports InRelease [74,6 kB]
Des:5 http://es.archive.ubuntu.com/ubuntu bionic-updates/universe i386 Packages
[727 kB]
Des:6 http://es.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages
[736 kB]
Descargados 1.715 kB en 7s (254 kB/s)
Leyendo lista de paquetes... Hecho
root@leonardo:/home/leonardo# sudo apt-get install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
apache2 ya está en su versión más reciente (2.4.29-1ubuntu4.5).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
root@leonardo:/home/leonardo# service apache2 start
root@leonardo:/home/leonardo#
```

## Comprobamos que este corriendo el servidor.

```
systemctl status apache2.
```

```
root@leonardo: /home/leonardo
Archivo Editar Ver Buscar Terminal Ayuda
Creando árbol de dependencias
Leyendo la información de estado... Hecho
apache2 ya está en su versión más reciente (2.4.29-1ubuntu4.5).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
root@leonardo:/home/leonardo# service apache2 start
root@leonardo:/home/leonardo# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Wed 2019-02-20 19:33:19 CST; 9min ago
   Process: 989 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 1055 (apache2)
     Tasks: 55 (limit: 4424)
    CGroup: /system.slice/apache2.service
            └─1055 /usr/sbin/apache2 -k start
               1056 /usr/sbin/apache2 -k start
               1057 /usr/sbin/apache2 -k start

feb 20 19:33:17 leonardo systemd[1]: Starting The Apache HTTP Server...
feb 20 19:33:19 leonardo apachectl[989]: AH00112: Warning: DocumentRoot [/var/www]
feb 20 19:33:19 leonardo apachectl[989]: AH00558: apache2: Could not reliably de
feb 20 19:33:19 leonardo systemd[1]: Started The Apache HTTP Server.
lines 1-17/17 (END)
```

## Creación del entorno para nuestro servidor.

En la consola vamos al directorio donde alojaremos nuestro sitio y crearemos el nombre de nuestra página web.

```
cd /var/www# mkdir primerhttp.com
```

```
root@leonardo: /var/www
Archivo Editar Ver Buscar Terminal Ayuda
root@leonardo:/home/leonardo# cd /var/www
root@leonardo:/var/www# ls
html
root@leonardo:/var/www# mkdir primerhttp.com
root@leonardo:/var/www# mkdir segundohttp.com
root@leonardo:/var/www# mkdir tercerhttp.com
root@leonardo:/var/www# ls
html primerhttp.com segundohttp.com tercerhttp.com
root@leonardo:/var/www#
```

Después agregamos nuestros archivos html para nuestro sitio.

```
cd /var/www/primerhttp.com# nano index.html
```

```
root@leonardo: /var/www/primerhttp.com
Archivo Editar Ver Buscar Terminal Ayuda
root@leonardo:/var/www/primerhttp.com# nano index.html
root@leonardo:/var/www/primerhttp.com# ls
index.html
root@leonardo:/var/www/primerhttp.com#
```

```
root@leonardo: /var/www/primerhttp.com
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 index.html
<html>
  <head>
    <title> Bienvenido al primer servidor http </title>
  </head>
  <body>
    <h1> El virtual host primerhttp.com funciona </h1>
  </body>
</html>

[ 8 líneas leídas ]
^G Ver ayuda  ^O Guardar    ^W Buscar     ^K Cortar Tex ^J Justificar  ^C Posición
^X Salir      ^R Leer fich. ^\ Reemplazar ^U Pegar txt   ^T Ortografía ^_ Ir a línea
```

**Configuraremos el host virtual.**



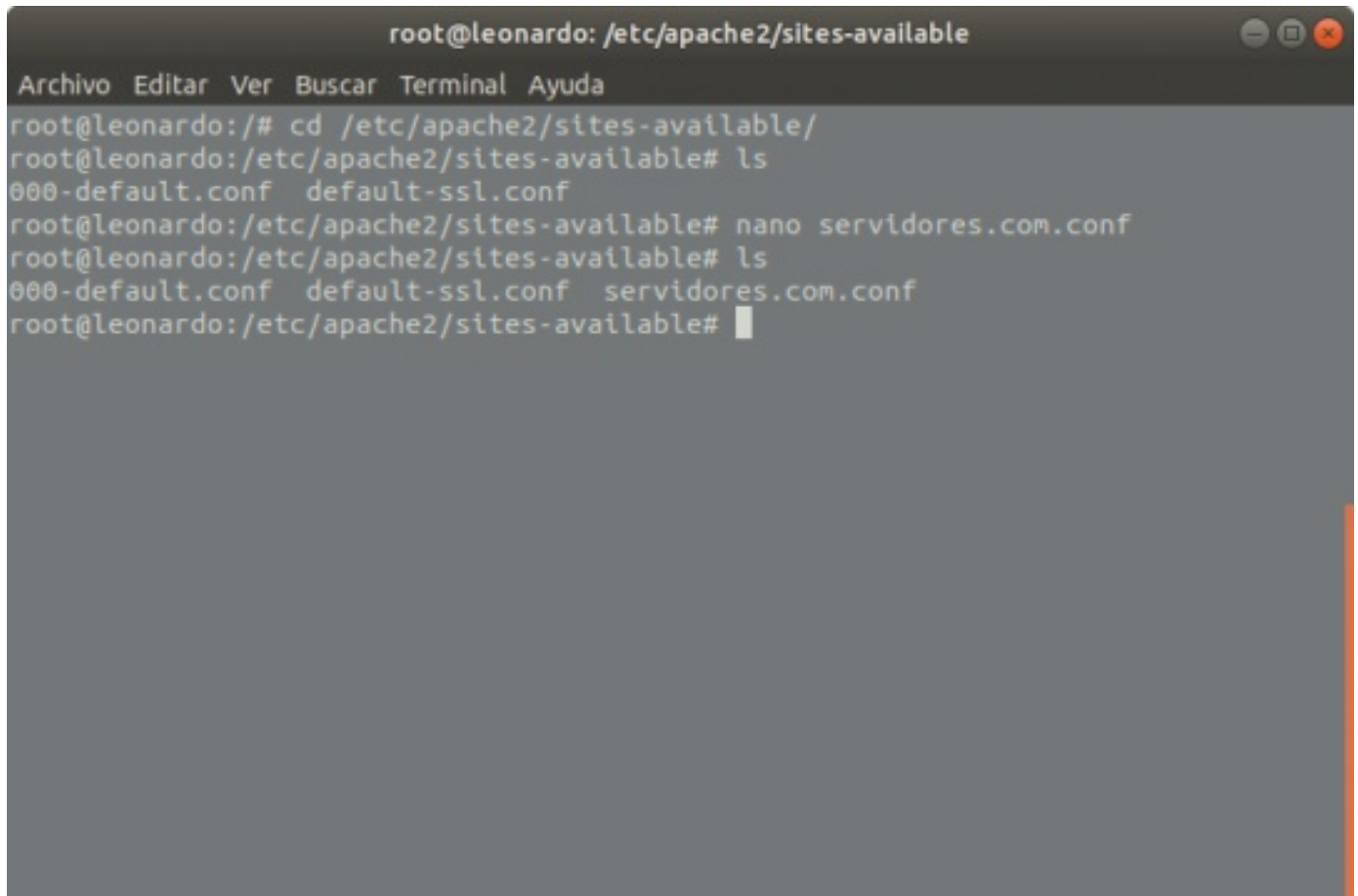
Nos vamos al directorio:

```
/etc/apache2/sites-available/
```

Creamos el archivo de configuración para nuestros host virtuales.

Un archivo guía de nuestro sitio es el: "000-default.conf", pero nosotros podemos crear el nuestro. Escribiendo servidores.com.conf

Como siguiente paso configuramos la información de nuestro archivo donde indicaremos la ubicación de los archivos fuente de cada sitio, donde se almacenan los logs y el nivel.

A terminal window titled 'root@leonardo: /etc/apache2/sites-available' with a menu bar 'Archivo Editar Ver Buscar Terminal Ayuda'. The terminal shows the following commands and output:

```
root@leonardo:/# cd /etc/apache2/sites-available/  
root@leonardo:/etc/apache2/sites-available# ls  
000-default.conf  default-ssl.conf  
root@leonardo:/etc/apache2/sites-available# nano servidores.com.conf  
root@leonardo:/etc/apache2/sites-available# ls  
000-default.conf  default-ssl.conf  servidores.com.conf  
root@leonardo:/etc/apache2/sites-available#
```

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

```
ServerName primerhttp.com
```

```
ServerAlias www.primerhttp.com
```

ServerAdmin webmaster@primer\_http.com

DocumentRoot /var/www/primerhttp.com

Customlog /var/log/apache2/primerhttp.com/acces.log common

ErrorLog /var/log/apache2/primerhttp.com/error.log

</VirtualHost>

<VirtualHost \*:80>

ServerName segundohttp.com

ServerAlias www.segundohttp.com

ServerAdmin webmaster@segundo\_http.com

DocumentRoot /var/www/segundohttp.com

Customlog /var/log/apache2/segundohttp.com/acces.log common

ErrorLog /var/log/apache2/segundohttp.com/error.log

</VirtualHost>

```
root@leonardo: /etc/apache2/sites-available
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 servidores.com.conf Modificado

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

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

<VirtualHost *:80>

^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Text ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

**Creamos las carpetas donde alojaremos los logs de cada sitio.**

```
/var/log/apache2# mkdir primerhttp.com
/var/log/apache2# mkdir segundohttp.com
/var/log/apache2# mkdir tercerhttp.com
```

```
root@leonardo: /var/log/apache2
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@leonardo:/var/log/apache2# mkdir primerhttp.com
root@leonardo:/var/log/apache2# mkdir segundohttp.com
root@leonardo:/var/log/apache2# mkdir tercerhttp.com
root@leonardo:/var/log/apache2# ls
access.log  other_vhosts_access.log  segundohttp.com
error.log   primerhttp.com            tercerhttp.com
root@leonardo:/var/log/apache2#
```

## Modificamos los permisos de nuestros sitios.

```
chown -R www-data:www-data /var/www/primerhttp.com/
chown -R www-data:www-data /var/www/segundohttp.com/
chown -R www-data:www-data /var/www/tercerhttp.com/
```

```
root@leonardo: /etc/apache2/sites-available
Archivo Editar Ver Buscar Terminal Ayuda
root@leonardo:/etc/apache2/sites-available# chown -R www-data:www-data /var/www/
primerhttp.com/
root@leonardo:/etc/apache2/sites-available# chown -R www-data:www-data /var/www/
segundohttp.com/
root@leonardo:/etc/apache2/sites-available# chown -R www-data:www-data /var/www/
tercerhttp.com/
root@leonardo:/etc/apache2/sites-available#
```

**Debemos agregar las direcciones de los hosts a nuestro archivo.**

/etc/hosts/

Con sus respectivos nombres de dominio.

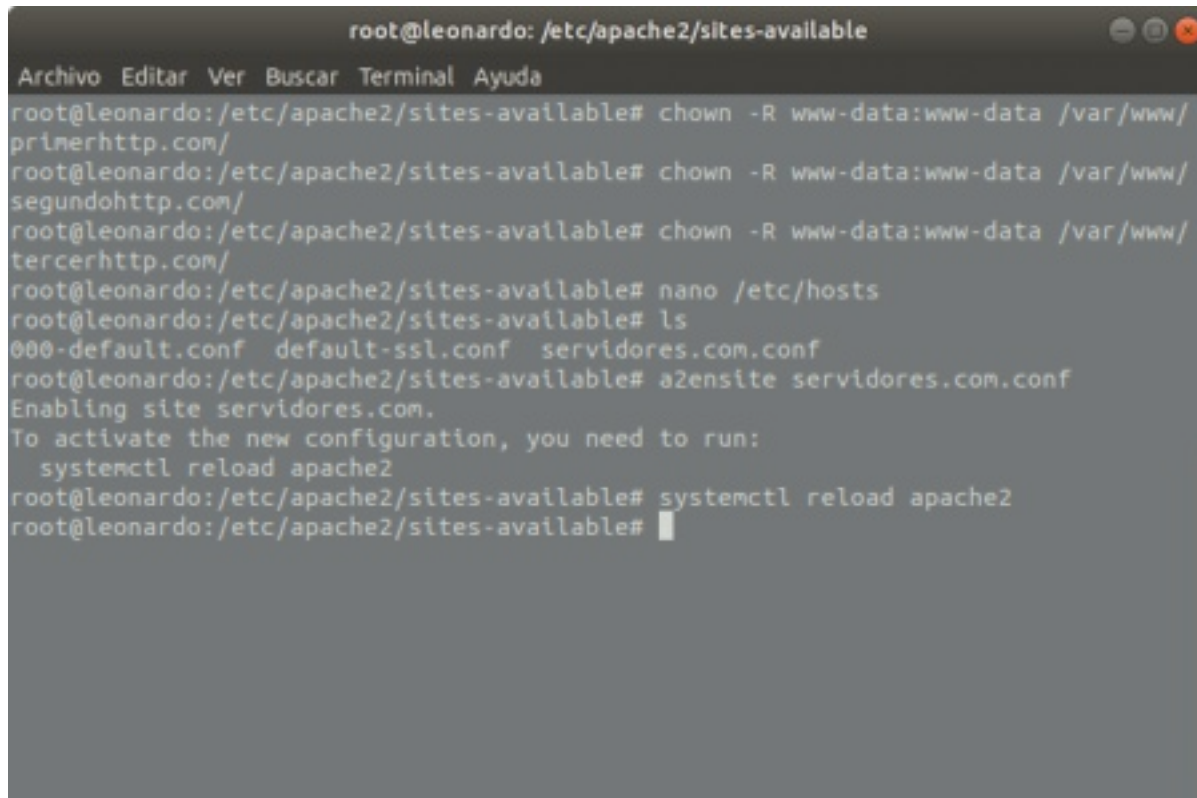
```
root@leonardo: /etc/apache2/sites-available
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 /etc/hosts
127.0.0.1 localhost
127.0.1.1 leonardo
127.0.1.1 primerhttp.com
127.0.1.1 segundohttp.com
127.0.1.1 tercerhttp.com
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

[ 11 lineas escritas ]
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Text ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

## Habilitando nuestros servidores.

Para poder habilitar nuestros servidores utilizaremos el comando *a2ensite*

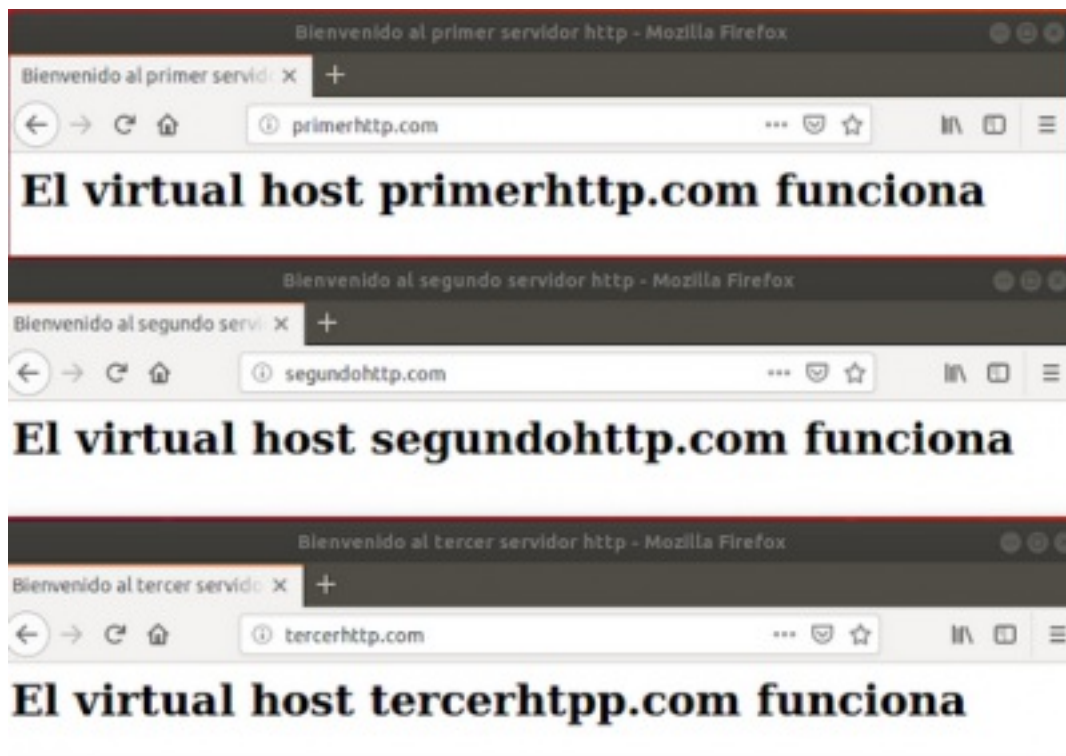
```
a2ensite servidores.com.conf
```

A terminal window titled 'root@leonardo: /etc/apache2/sites-available' with a menu bar 'Archivo Editar Ver Buscar Terminal Ayuda'. The terminal shows the following commands and output:

```
root@leonardo:/etc/apache2/sites-available# chown -R www-data:www-data /var/www/
primerhttp.com/
root@leonardo:/etc/apache2/sites-available# chown -R www-data:www-data /var/www/
segundohhttp.com/
root@leonardo:/etc/apache2/sites-available# chown -R www-data:www-data /var/www/
tercerhttp.com/
root@leonardo:/etc/apache2/sites-available# nano /etc/hosts
root@leonardo:/etc/apache2/sites-available# ls
000-default.conf default-ssl.conf servidores.com.conf
root@leonardo:/etc/apache2/sites-available# a2ensite servidores.com.conf
Enabling site servidores.com.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@leonardo:/etc/apache2/sites-available# systemctl reload apache2
root@leonardo:/etc/apache2/sites-available#
```

Reiniciamos el servicio y comprobamos que tengamos acceso el comando para poder reiniciar nuestro servidor es:

```
systemctl reload apache2
```



## Restricción por IP del cliente.

Para poder restringir el acceso por una IP específica, necesitamos modificar el archivo “apache2.conf” ubicado en el directorio /etc/apache2/

Para esta parte se va a restringir el acceso mediante el directorio.

```
<Directory /var/www/primerhttp.com>  
    Options ALL  
    AllowOverride ALL  
    <RequireALL>  
        Require all granted  
        Require ip 10.100.96.38  
    </RequireALL>  
</Directory>
```



```
root@leonardo: /etc/apache2
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 apache2.conf

Options Indexes FollowSymLinks
AllowOverride None
Require all granted
</Directory>

#Restringir por IP

<Directory /var/www/primerhttp.com>
Options ALL
AllowOverride All
<RequireAll>
Require all granted
Require ip 10.100.96.38
</RequireAll>
</Directory>

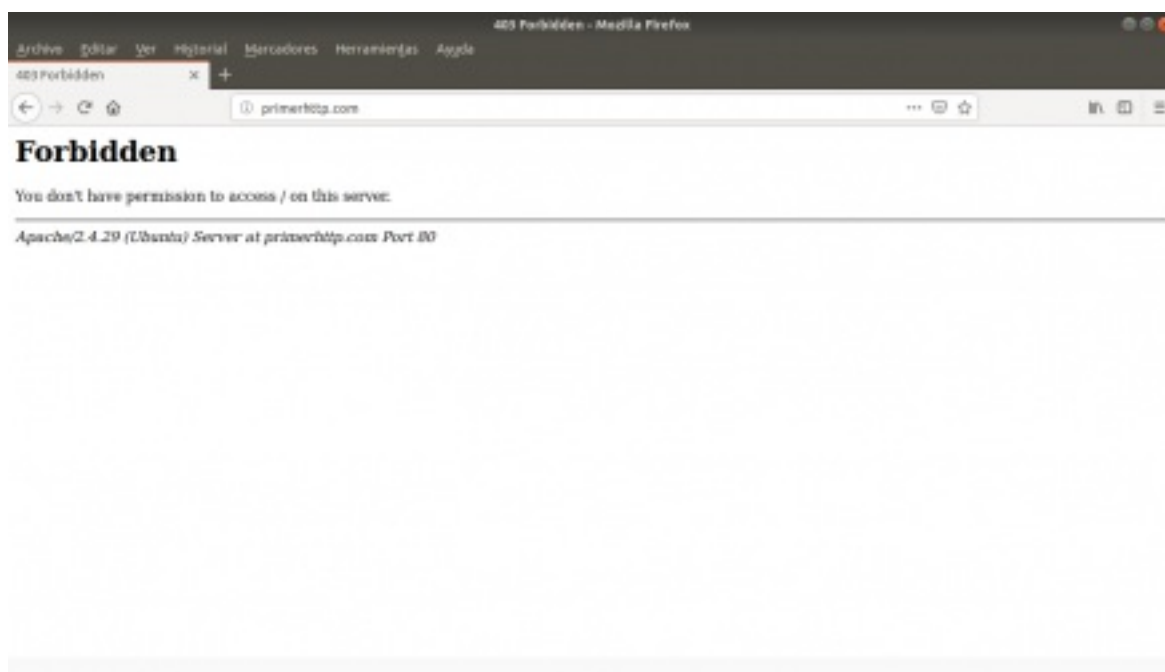
#<Directory /srv/>
#Options Indexes FollowSymLinks
#AllowOverride None
[ 238 líneas escritas ]
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Tex ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

En otro caso podemos determinar que la IP que asignamos no pueda acceder con un simple cambio:

```
<Directory /var/www/primerhttp.com>
Options ALL
AllowOverride ALL
<RequireALL>
Require all granted
Require not ip 10.100.96.38
</RequireALL>
</Directory>
```

El resultado debe salir como la siguiente imagen:





## Restricción por segmento de red.

Volvemos a abrir nuestro archivo “apache2.conf que se ubica en /etc/apache2”.

En las líneas “Require not ip agregamos las ip que deseemos bloquear”

```
<Directory /var/www/primerhttp.com>
    Options ALL
    AllowOverride ALL
    <RequireAll>
        Require all granted
        Require not ip 10.100.96.38 192.168.1.67
    </RequireAll>
</Directory>
```

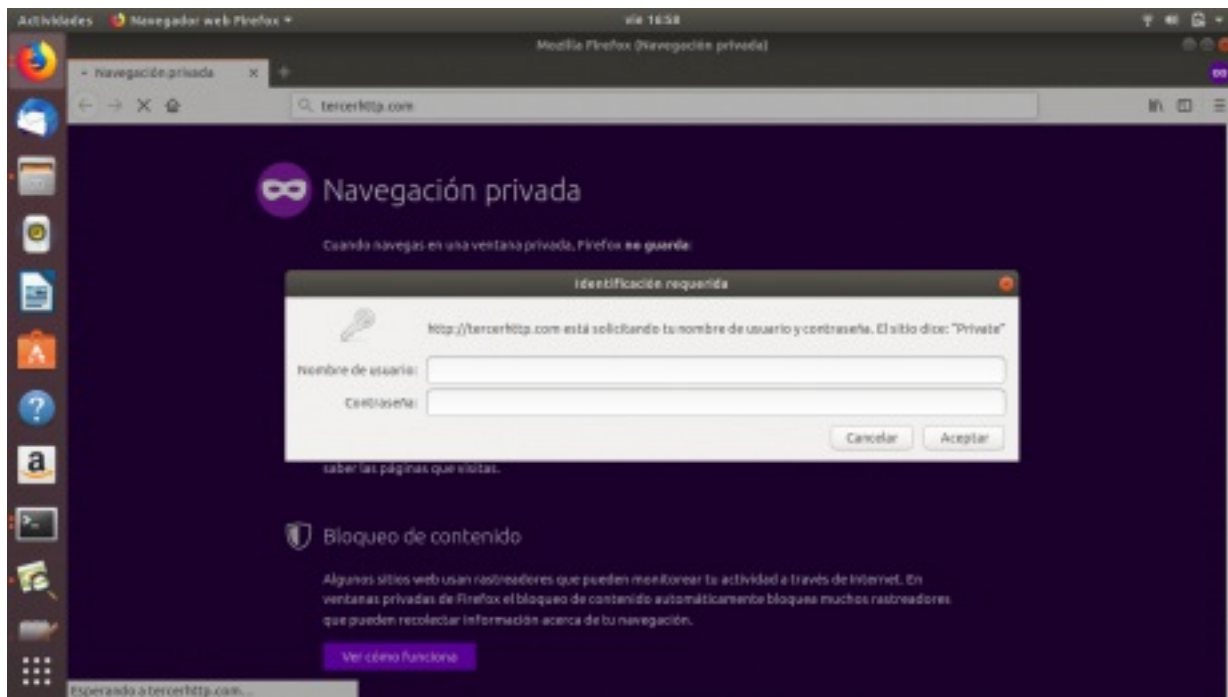
Insertar imagen aquí.

## Restricción por usuario.

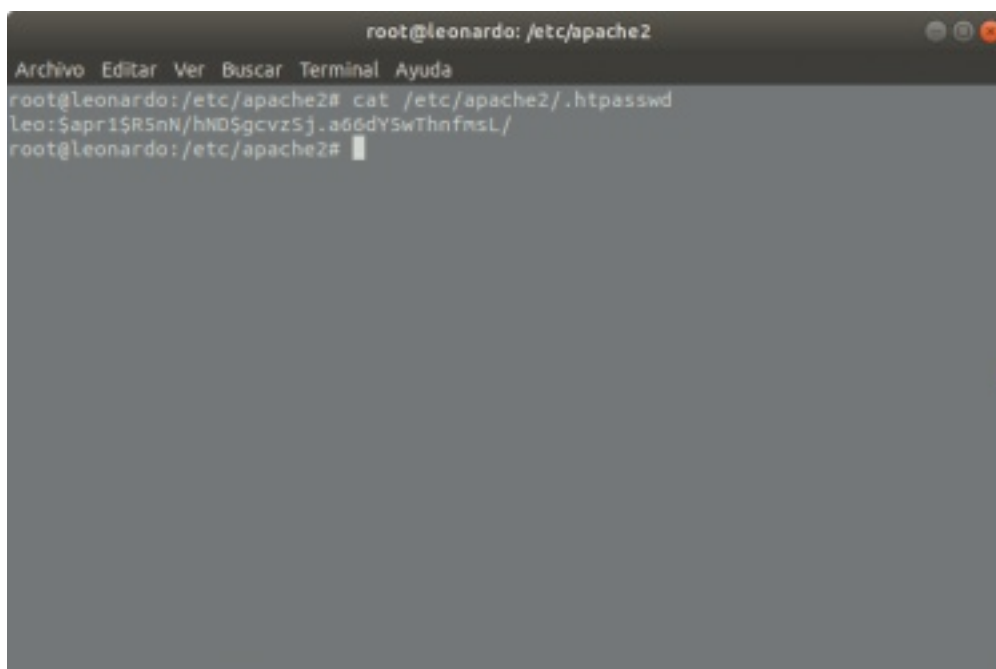
Para esta restricción es necesario que creamos un archivo donde agregaremos los usuarios, su respectiva contraseña.

El respectivo comando que usaremos para generar la contraseña es:

```
htpasswd -c /etc/apache2/.htpasswd leo
```

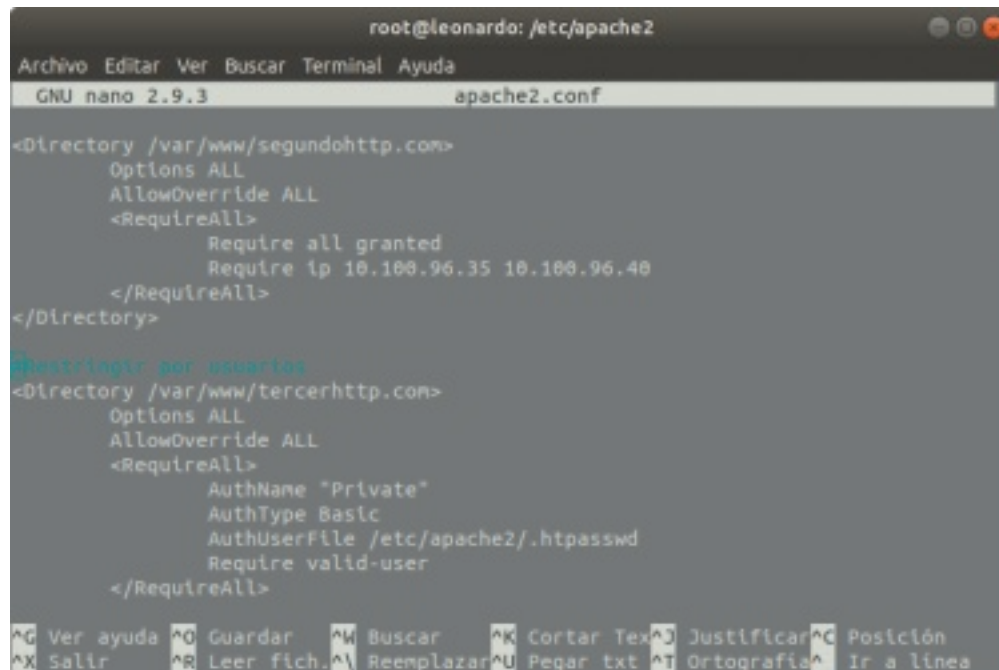


Procedemos a ver lo que contiene el archivo



El siguiente procedimiento que debemos realizar es agregar nuestro

archivo a nuestro directorio para indicar que solo esos usuarios van a poder acceder.



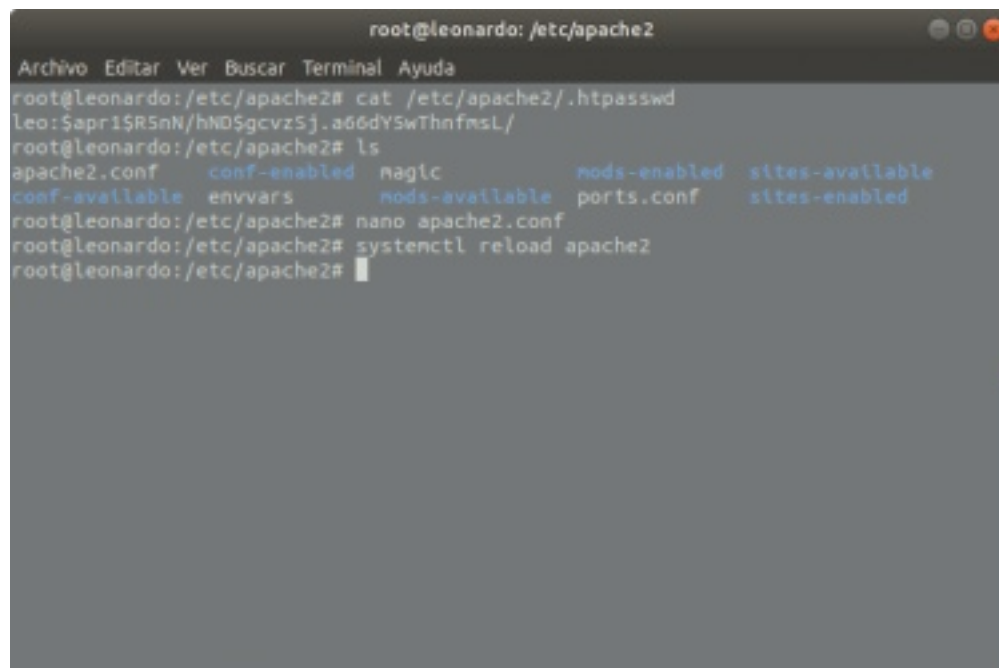
```
root@leonardo: /etc/apache2
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 apache2.conf

<Directory /var/www/segundohttp.com>
    Options ALL
    AllowOverride ALL
    <RequireAll>
        Require all granted
        Require ip 10.100.96.35 10.100.96.40
    </RequireAll>
</Directory>

# Restringir por usuarios
<Directory /var/www/tercerhttp.com>
    Options ALL
    AllowOverride ALL
    <RequireAll>
        AuthName "Private"
        AuthType Basic
        AuthUserFile /etc/apache2/.htpasswd
        Require valid-user
    </RequireAll>

^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Texto ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

Reiniciamos el servidor para poder ver los cambios que se han aplicado y ahora al momento de ingresar nos pedirá el usuario y contraseña.



```
root@leonardo: /etc/apache2
Archivo Editar Ver Buscar Terminal Ayuda
root@leonardo:/etc/apache2# cat /etc/apache2/.htpasswd
leo:$apr1$R5nN/hND$gcVzSj.a66dYSwThnfmsL/
root@leonardo:/etc/apache2# ls
apache2.conf  conf-enabled  magic          mods-enabled  sites-available
conf-available  envvars      mods-available  ports.conf    sites-enabled
root@leonardo:/etc/apache2# nano apache2.conf
root@leonardo:/etc/apache2# systemctl reload apache2
root@leonardo:/etc/apache2#
```



## Configuración de puertos de operación

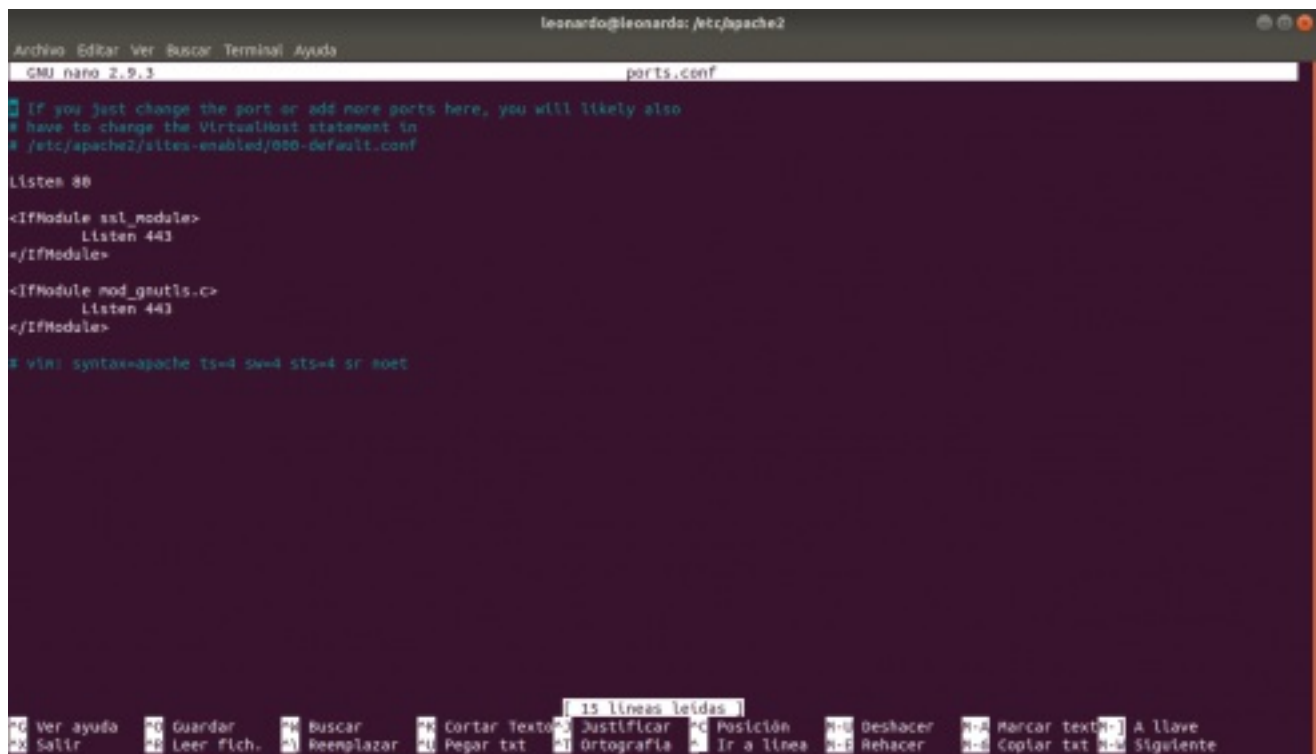
Cuando httpd se ejecuta, se mapea a una dirección y un puerto en la máquina local, y espera a recibir peticiones. Por defecto, escucha en todas las direcciones de la máquina. Ahora bien, se le puede especificar que escuche en un determinado puerto, o en una sola dirección IP específica, o una combinación de ambos. A menudo esto se combina con la característica de los Hosts virtuales, que determina como responde el httpd a diferentes direcciones IP, nombres de máquinas y puertos.

Vamos a configurar nuestros puertos escucha y el puerto que estará escuchando para nuestro protocolo HTTPS.

Para este caso utilizaremos el puerto 80 para el protocolo de HTTP y el 443 para HTTPS.

Modificamos el archivo: ports que se encuentra ubicado.

```
/etc/apache2/
```



```
leonardo@leonardo: /etc/apache2
GNU nano 2.9.3 ports.conf
# 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
```

## Configuración HTTPS Certificados

Procederemos a crear nuestro certificado para esto nos vamos a la ruta: `/etc/apache2#` y escribimos el siguiente comando:

```
openssl req -new -x509 -days 365 -keyout sitios.key -out sitios.crt -nodes -subj '/O=Company/OU=ESCOM/CN=www.primerhttp.com'
```

```
root@leonardo: /etc/apache2
Archivo Editor Ver Buscar Terminal Ayuda
root@leonardo:/etc/apache2# openssl req -new -x509 -days 365 -keyout sitios.key
-out sitios.crt -nodes -subj '/O=Company/OU=ESCOM/CN=www.primerhttp.com'
Generating a 2048 bit RSA private key
.....
.....
.....+++
.....+++
writing new private key to 'sitios.key'
-----
root@leonardo:/etc/apache2#
```

Dependiendo del puerto que se haya configurado para que pueda recibir las peticiones HTTPS, vamos a modificar el archivo `servidores.com.conf`

```
<VirtualHost *:80>
ServerName primerhttp.com
ServerAlias www.primerhttp.com
ServerAdmin webmaster@primer_http.com
DocumentRoot /var/www/primerhttp.com
CustomLog /var/log/apache2/primerhttp.com/acces.log common
ErrorLog /var/log/apache2/primerhttp.com/error.log
SSLEngine On
SSLCertificateFile /etc/apache2/sitios.crt
SSLCertificateKeyFile /etc/apache2/sitios.key
</VirtualHost>
```

```
leonardo@leonardo: /etc/apache2/sites-available
GNU nano 2.9.3 default-ssl.conf

# 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

#
# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

#
# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/certs/sites.crt
SSLCertificateKeyFile /etc/ssl/private/sites.key

#
# Server Certificate Chain:
# Point SSLCertificateChainFile at a file containing the
# concatenation of PEM encoded CA certificates which form the
# certificate chain for the server certificate. Alternatively
# the referenced file can be the same as SSLCertificateFile
# when the CA certificates are directly appended to the server
# certificate for convenience.
#SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

#
# Certificate Authority (CA):
# Set the CA certificate verification path where to find CA
# certificates for client authentication or alternatively one
# huge file containing all of them (file must be PEM encoded)
# Note: Inside SSLCACertificatePath you need hash symlinks
# to point to the certificate files. Use the provided
# Makefile to update the hash symlinks after changes.
```

Ingresamos a nuestro sitio para comprobar que está funcionando el certificado, así debe aparecernos:



## Personalización de páginas de error.

Para personalización de las páginas de error necesitamos acudir a la siguiente ruta:

```
/etc/apache2/sites-available/
```

Agregamos las siguientes líneas.

```
ErrorDocument 401 /401.html
```

```
ErrorDocument 403/403.html
```

Así respectivamente con las páginas de error que desee agregar.

Error 401



Contacte con el administrador en sitio.

## Nivel de logs

El servidor Apache nos permite activar un módulo llamado `mod_log_config`, una de las ventajas de activarlo es que nos permite modificar el nivel de logs y hacerlos personalizados indicando las directivas que nos proporciona la documentación oficial de apache [1]. En general este módulo nos permite activar tres directivas, la de `TransferLog`, `LogFormat` y `CustomLog`. Para este caso hemos activado las siguientes banderas cuya descripción se anexa

Podemos consultar la siguiente tabla.

### Configuración Logs



# Referencias

[¿Cómo configurar Virtual Hosts de Apache en Ubuntu 16.04?](#)

[SSL en Ubuntu](#)

[Directrices de configuración en httpd.conf - MIT](#)

[Restringir acceso por IP en Apache](#)

[mod\\_log\\_config - Apache HTTP Server Version 2.4](#)

[Mapeo de Direcciones y Puertos.](#)

[Configurar página de error 404 en Apache](#)