

Práctica 4 Creación de dos sitios virtuales con SSL y autenticación básica



1. Reconocimiento de Parámetros de Administración (Apache):

1.1 MaxClients

1.2 ServerTokens

1.3 ServerSignature

1.4 KeepAlive

1.5 Timeout

2. Ampliación de Funcionalidad mediante Módulos (Apache)

2.1 Comprobación de los módulos

2.2 Instalación del módulo security2

2.3 Instalación del módulo rewrite

3. Creación y Configuración de Sitios Virtuales (Apache):

3.1 Creación del sitio principal www.miempresa.com

3.1.1 Crear un virtualhost puerto 80 ServerName www.miempresa.com en el archivo de configuración miempresa.com.conf

3.1.2 Creando los archivos del sitio web www.miempresa.com

3.1.3 Modificando el archivo hosts del sistema para acceder a www.miempresa.com

3.1.4 Obtener un certificado SSL

3.1.4.1 Pedir un certificado con certbot para despliegue real

3.1.4.2 Crear un certificado autofirmado para acceder de manera local y agregar el módulo SSL a apache2

3.1.5 Creación de un virtualhost puerto 443 para conexiones SSL en www.miempresa.com

3.1.6 Redirigir las conexiones del puerto 80 al puerto 443 (solo permitir HTTPS) en www.miempresa.com

3.1.7 Comprobando conexión HTTPS

3.2 Creación del sitio de administración admin.miempresa.com.

3.2.1 Configuración de un virtualhost puerto 80 con directiva ServerName www.admin.miempresa.com

3.2.2 Creando los archivos del sitio web www.admin.miempresa.com

3.2.3 Modificando el archivo hosts del sistema para acceder a www.admin.miempresa.com

3.2.3 Creación de un virtualhost puerto 443 para conexiones SSL en www.admin.miempresa.com.conf

3.2.4 Redirigir las conexiones del puerto 80 al puerto 443 (solo permitir HTTPS) en www.admin.miempresa.com

3.2.5 Añadiendo autenticación básica para garantizar el acceso restringido en www.admin.miempresa.com

1. Reconocimiento de Parámetros de Administración (Apache):

En este primer apartado, se recogen las diferentes directivas con cada uno de sus posibles valores que se pueden aplicar a nuestro servidor web apache2, especificando una breve descripción de cada uno de ellos y un ejemplo de código.

Todas estas directivas se aplican en el archivo de configuración general de apache2 ubicado en `/etc/apache2/apache2.conf`

1.1 MaxClients

Descripción: Determina el número máximo de solicitudes simultáneas que el servidor Apache puede manejar. Controla cuántos clientes pueden estar conectados al servidor al mismo tiempo

Ejemplo:
MaxClients 150

1.2 ServerTokens

Descripción: Controla la información que el servidor revela sobre sí mismo en las respuestas HTTP. Se configura en el archivo de configuración principal de Apache, generalmente **httpd.conf** o **apache2.conf**.

Esta directiva tiene diferentes posibles valores:

ServerTokens Full | ServerTokens OS | ServerTokens Minor | ServerTokens Major | ServerTokens Prod | ServerTokens Min

Full: Muestra la información completa sobre el servidor en el encabezado del servidor. Este es el valor predeterminado si no se especifica ningún otro.

OS: Muestra solo el nombre del sistema operativo en el encabezado del servidor.

Minor: Muestra la versión principal y menor del servidor en el encabezado del servidor.

Major: Muestra solo la versión principal del servidor en el encabezado del servidor.

Prod: Muestra solo "Apache" en el encabezado del servidor, sin información de versión.

Min: Muestra solo el nombre del servidor y oculta la información de la versión.

1.3 ServerSignature

Descripción: Determina si se incluye una firma del servidor en las páginas de error generadas por Apache.

Esta directiva tiene 2 valores posibles:

ServerSignature On | ServerSignature Off

On: Muestra la información detallada de la firma del servidor en las páginas de error generadas por Apache.

Off: No muestra información detallada de la firma del servidor en las páginas de error generadas por Apache. Este es el valor predeterminado si no se especifica ningún otro.

1.4 KeepAlive

Descripción: Se utiliza para controlar si se permite la conexión persistente (keep-alive) entre el servidor y el cliente. La conexión persistente permite que una única conexión TCP se utilice para varias solicitudes HTTP, lo que puede mejorar el rendimiento al evitar la necesidad de abrir y cerrar una conexión para cada solicitud.

Esta directiva tiene 2 valores posibles:

KeepAlive On | KeepAlive Off

On: Permite la conexión persistente. Este es el valor predeterminado si no se especifica ningún otro.

Off: Deshabilita la conexión persistente, lo que significa que se abrirá y cerrará una conexión para cada solicitud.

Además, hay otras **dos directivas relacionadas** que puedes usar para ajustar el comportamiento de la conexión persistente:

MaxKeepAliveRequests: Esta directiva establece el número máximo de solicitudes que se pueden enviar a través de una conexión persistente antes de cerrarla. Por ejemplo, si deseas cerrar la conexión después de 100 solicitudes, puedes configurar:

MaxKeepAliveRequests 100

KeepAliveTimeout: Esta directiva establece el tiempo máximo en segundos que una conexión persistente puede permanecer abierta sin recibir una solicitud. Después de este tiempo, la conexión se cerrará. Por ejemplo, para cerrar la conexión después de 5 segundos de inactividad, puedes configurar:

KeepAliveTimeout 5

1.5 Timeout

Descripción: Establece el tiempo máximo en segundos que el servidor esperará antes de recibir una solicitud.

Esta directiva tiene 5 posibles directivas:

Timeout | KeepAliveTimeout | ProxyTimeout | ProxyReceiveTimeout | ProxyPassTimeout

Timeout: Este es el valor predeterminado. Define el tiempo máximo en segundos que el servidor esperará para recibir un paquete en una conexión no segura antes de cerrar la conexión, puedes configurar:

Timeout: 300

KeepAliveTimeout: Esta directiva establece el tiempo máximo en segundos que una conexión persistente puede permanecer abierta sin recibir una solicitud. Después de este tiempo, la conexión se cerrará. Por ejemplo, para cerrar la conexión después de 5 segundos de inactividad, puedes configurar:

KeepAliveTimeout 5

ProxyTimeout: Define el tiempo máximo en segundos que el servidor esperará para recibir un paquete en una conexión segura con un servidor proxy antes de cerrar la conexión.

Puedes configurar:

ProxyTimeout 60

ProxyReceiveTimeout: Especifica el tiempo máximo en segundos que el servidor esperará para recibir una respuesta del servidor proxy después de haber enviado una solicitud.

Puedes configurar:

ProxyReceiveTimeout 60

ProxyPassTimeout: Define el tiempo máximo en segundos que el servidor esperará para recibir una respuesta del servidor backend después de haber enviado una solicitud. Puedes configurar:

ProxyPassTimeout 60

2. Ampliación de Funcionalidad mediante Módulos (Apache)

2.1 Comprobación de los módulos

Vamos a añadir dos módulos adicionales para la configuración que ya tenemos en nuestro servidor.

security2: Para añadir una capa adicional de seguridad mediante la detección y prevención de ataques.

rewrite.load: Para realizar reescrituras de URL y redirecciones de manera flexible y segura.

Para añadir estos mod , primero tenemos que acceder a la carpeta **etc/apache2/mods-available** desde la terminal para comprobar si tenemos el mod

cd /etc/apache2/mods-available

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996: /etc/apache2/mods-available
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$ cd /etc/apache2/mods-available/
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/etc/apache2/mods-available$
```

ls

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996: /etc/apache2/mods-available
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/etc/apache2/mods-available$ ls
access_compat.load      cgi.load                log_debug.load         ratelimit.load
actions5.conf           charset_lite.load       log_forensic.load      reflector.load
actions.load            data.load               lua.load               remoteip.load
alias.conf              dav_fs.conf             macro.load              reqtimeout.conf
alias.load              dav_fs.load             md.load                reqtimeout.load
allowmethods.load       dav.load                mime.conf              request.load
asis.load               dav_lock.load           mime.load              rewrite.load
auth_basic.load          dbd.load                mime_magic.conf        sed.load
auth_digest.load        deflate.conf            mime_magic.load        session_cookie.load
auth_form.load          deflate.load            mpm_event.conf         session_crypto.load
authn_anon.load         dialup.load             mpm_event.load         session_dbd.load
authn_core.load         dir.conf                mpm_prefork.conf       session.load
authn_dbd.load          dir.load                mpm_prefork.load       setenvif.conf
authn_dbm.load          dump_io.load            mpm_worker.conf        setenvif.load
authn_file.load         echo.load               mpm_worker.load        slotmem_plain.load
authn_socache.load       env.load                negotiation.conf        slotmem_shm.load
authnz_fcgi.load         expires.load            negotiation.load        socache_dbm.load
authnz_ldap.load        ext_filter.load         proxy_ajp.load          socache_memcache.load
authz_core.load          file_cache.load         proxy_balancer.conf     socache_redis.load
authz_dbd.load          filter.load              proxy_balancer.load     socache_shmcb.load
authz_dbm.load           headers.load            proxy.conf              spelling.load
authz_groupfile.load     heartbeat.load          proxy_connect.load      ssl.conf
authz_host.load          heartmonitor.load       proxy_express.load      ssl.load
authz_owner.load         http2.conf              proxy_fcgi.load         status.conf
authz_user.load          http2.load              proxy_fdpass.load       status.load
autoindex.conf           ident.load              proxy_ftp.conf          substitute.load
autoindex.load           imagemap.load           proxy_ftp.load          suexec.load
 Brotli.load             include.load            proxy_hcheck.load       unique_id.load
buffer.load              info.conf               proxy_html.conf         userdir.conf
cache_disk.conf          info.load               proxy_html.load         userdir.load
cache_disk.load          lbmethod_bybusyness.load proxy_html2.load         usertrack.load
cache.load               lbmethod_byrequests.load proxy_http.load          vhost_alias.load
cache_socache.load        lbmethod_bytraffic.load proxy_http.load          xml2enc.load
cern_meta.load           ldap.conf               proxy_scgi.load          proxy_uwsgi.load
cgid.conf                ldap.load               proxy_uwsgi.load        proxy_wstunnel.load
cgid.load
```

2.2 Instalación del módulo security2

Como podemos observar, no tenemos el mod security2, por lo que tenemos que hacer una instalación con el comando **sudo apt install libapache2-mod-security2**

```

joseangel@usuario-Standard-PC-i440FX-PIIX-1996: /etc/apache2/mods-available
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/etc/apache2/mods-available$ sudo apt install libapache2-mod-security2
[sudo] contraseña para joseangel:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  libflashrom1 libftdi1-2 liblvm13
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  liblua5.1-0 modsecurity-crs
Paquetes sugeridos:
  lua geopip-database-contrib ruby python
Se instalarán los siguientes paquetes NUEVOS:
  libapache2-mod-security2 liblua5.1-0 modsecurity-crs
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 39 no actualizados.
Se necesita descargar 504 kB de archivos.
Se utilizarán 2.376 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
Des:1 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 liblua5.1-0 amd64 5.1.5-8.1build4 [99,9 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 libapache2-mod-security2 amd64 2.9.5-1 [265 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 modsecurity-crs all 3.3.2-1 [139 kB]
Descargados 504 kB en 3s (163 kB/s)
Seleccionando el paquete liblua5.1-0:amd64 previamente no seleccionado.
(Leyendo la base de datos ... 222101 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../liblua5.1-0_5.1.5-8.1build4_amd64.deb ...
Desempaquetando liblua5.1-0:amd64 (5.1.5-8.1build4) ...
Seleccionando el paquete libapache2-mod-security2 previamente no seleccionado.
Preparando para desempaquetar .../libapache2-mod-security2_2.9.5-1_amd64.deb ...
Desempaquetando libapache2-mod-security2 (2.9.5-1) ...
Seleccionando el paquete modsecurity-crs previamente no seleccionado.
Preparando para desempaquetar .../modsecurity-crs_3.3.2-1_all.deb ...
Desempaquetando modsecurity-crs (3.3.2-1) ...
Configurando modsecurity-crs (3.3.2-1) ...
Configurando liblua5.1-0:amd64 (5.1.5-8.1build4) ...
Configurando libapache2-mod-security2 (2.9.5-1) ...
apache2_invoke: Enable module security2
Procesando disparadores para libc-bin (2.35-0ubuntu3.4) ...

```

Además, también tenemos que habilitar el mod headers con el comando **sudo a2enmod headers**

```

joseangel@usuario-Standard-PC-i440FX-PIIX-1996: /etc/apache2/mods-enabled
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/etc/apache2/mods-enabled$ sudo a2enmod headers
Enabling module headers.
To activate the new configuration, you need to run:
  systemctl restart apache2
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/etc/apache2/mods-enabled$

```

Una vez instalados los mod, vamos a la carpeta **etc/apache2/mods-enabled** para hacer un **ls** y verificar que tenemos habilitados los mods **security2** y **headers**

```

joseangel@usuario-Standard-PC-i440FX-PIIX-1996: /etc/apache2/mods-enabled
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/etc/apache2/mods-enabled$ ls
access_compat.load  authn_file.load  autoindex.load  env.load  mpm_event.conf  reqtimeout.load  status.conf
alias.conf          authz_core.load  deflate.conf     filter.load  mpm_event.load  security2.conf   status.load
alias.load          authz_host.load  deflate.load     headers.load  negotiation.conf security2.load    unique_id.load
auth_basic.load     authz_user.load  dir.conf        mime.conf    negotiation.load setenvif.conf    userdir.conf
authn_core.load     autoindex.conf  dir.load        mime.load    reqtimeout.conf  setenvif.load    userdir.load
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/etc/apache2/mods-enabled$

```

Una vez tenemos el mod en la carpeta enabled, procedemos a instalar el mod rewrite.

2.3 Instalación del módulo rewrite

Este mod **ya lo tenemos** en la carpeta **/etc/apache2/mods-available** por lo que con el comando **sudo a2enmod rewrite** nos crea el enlace simbólico a la carpeta enabled y ya estaría instalado

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996: /etc/apache2/mods-enabled$ sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
systemctl restart apache2
```

Volvemos a hacer un **ls** en la carpeta **etc/apache2/mods-enabled** para comprobar que tenemos los 3 mods en la carpeta **/etc/apache2/mods-enabled**

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996: /etc/apache2/mods-enabled$ ls
access_compat.load  authn_file.load  autoindex.load  env.load  mpm_event.conf  reqtimeout.load  setenvif.load  userdir.load
alias.conf          authn_core.load  deflate.conf     filter.load  mpm_event.load  rewrite.load      status.conf
alias.load          authz_core.load  deflate.load     headers.load  negotiation.conf  security2.conf    status.load
auth_basic.load     authz_host.load  dir.conf        mime.load    negotiation.load  security2.load    unique_id.load
authn_core.load     authz_user.load  dir.load        mime.load    reqtimeout.conf  setenvif.conf     userdir.conf
```

Con los 3 mods instalados, hacer un restart de apache2 con el comando **sudo service apache2 restart**

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996: /etc/apache2/mods-enabled$ sudo service apache2 restart
joseangel@usuario-Standard-PC-i440FX-PIIX-1996: /etc/apache2/mods-enabled$
```


3. Creación y Configuración de Sitios Virtuales (Apache):

3.1 Creacion del sitio principal www.miempresa.com

■ Sitio Virtual 1 (Principal):

- Nombre del Sitio: www.miempresa.com.
- Directorio Raíz: `/var/www/html/ecommerce/shop`
- Configuración Adicional: Establecer reglas de acceso para permitir sólo conexiones seguras (HTTPS).

3.1.1 Crear un virtualhost puerto 80 ServerName www.miempresa.com en el archivo de configuracion `miempresa.com.conf`

Para configurar el primer sitio virtual , necesitamos crear un archivo `miempresa.com.conf` para nuestro sitio en la carpeta `/etc/apache2/sites-available`

Vamos a la carpeta `etc/apache2/sites-available`

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$ cd /etc/apache2/sites-available/  
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/etc/apache2/sites-available$
```

Creamos el archivo miempresa.com.conf con `sudo touch miempresa.com.conf`

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/etc/apache2/sites-available$ sudo touch miempresa.com.conf  
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/etc/apache2/sites-available$
```

Pasamos el archivo miempresa.com.conf a visual studio code y creamos un nuevo virtual host en el puerto 80 con la directiva `ServerName` www.miempresa.com para que cuando accedemos mediante este nombre de dominio , apache2 use la configuración de este virtualhost para este sitio.

En la directiva `DocumentRoot` añadimos la ruta de donde se encontraran los archivos del sitio, que es `/var/www/html/ecommerce/shop`

En la directiva `ServerName` añadimos el nombre de dominio `www.miempresa.com`

En la directiva `DirectoryIndex` ponemos el nombre del archivo que queremos que se muestre cuando accedemos a nuestra web, en mi caso pondré un archivo `index.html`

En la directiva `ErrorDocument 404` ponemos el mensaje de error al producirse un código de error HTTP 404 , al no encontrar un recurso en nuestro directorio web.

Así quedaría nuestro virtualhost puerto 80 ServerName www.miempresa.com del archivo [miempresa.com.conf](#)

```
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com
ServerName www.miempresa.com
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/ecommerce/shop
DirectoryIndex index.html
ErrorDocument 404 "no se ha encontrado la página en nuestro servidor, practica ecommerce, pagina principal"

# 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

# 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>
```

Tenemos que activar el sitio, para ello ejecutamos el comando **sudo a2ensite miempresa.com**, para que se cree el **enlace simbólico** desde el archivo **etc/apache2/sites-available/miempresa.com.conf** a **etc/apache2/sites-enabled/**

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/etc/apache2/sites-available$ sudo a2ensite miempresa.com
Enabling site miempresa.com.
To activate the new configuration, you need to run:
    systemctl reload apache2
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/etc/apache2/sites-available$
```

3.1.2 Creando los archivos del sitio web www.miempresa.com

Para crear el directorio con los archivos de nuestra web en la ruta especificada en el archivo **000-default.conf** Accedemos a la ruta **/var/www/html/**

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$ cd /var/www/html
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html$
```

Desde aquí ponemos el comando **sudo mkdir ecommerce**

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html$ sudo mkdir ecommerce
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html$
```

Ahora creamos la carpeta shop con **sudo mkdir ecommerce/shop**

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html$ sudo mkdir ecommerce/shop
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html$
```

Accedemos a la carpeta con **cd ecommerce/shop**

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996: /va
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html$ cd ecommerce/shop/
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html/ecommerce/shop$
```

Creamos un archivo con **sudo touch index.html**

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html/ecommerce/shop$ sudo touch index.html
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html/ecommerce/shop$
```

Pasamos el archivo a nuestro Visual Studio Code y creamos una web que identifique qué es la web principal

```
<> index.html x 000-default.conf ports.conf apache2.conf pepe.es.conf file
var > www > html > ecommerce > shop > <> index.html > html > body > p
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Ecommerce Principal</title>
7 </head>
8 <body>
9   <h1>Esta es la web principal de Ecommerce</h1>
10  <p>probando la web PRINCIPAL</p>
11 </body>
12 </html>
```

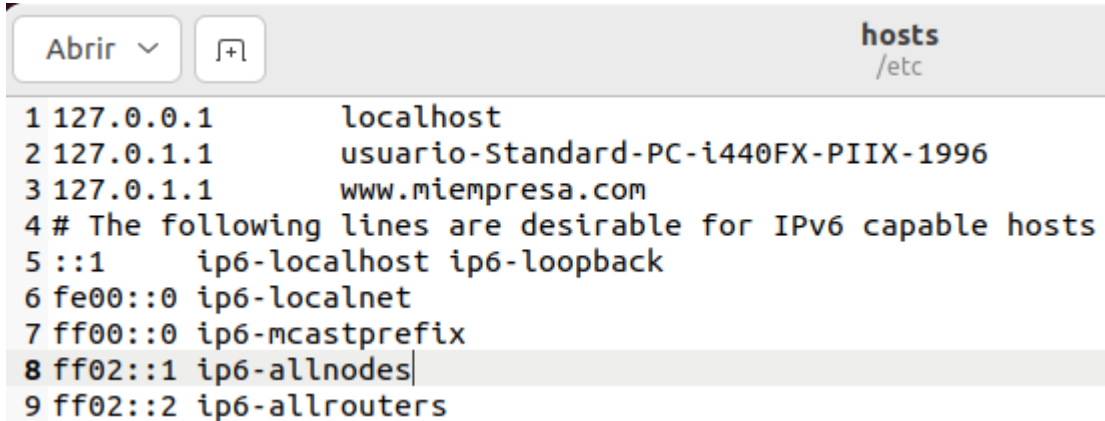
Guardamos los cambios en el archivo y reiniciamos de nuevo apache2 con **sudo service apache2 restart**

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html/ecommerce/shop$ sudo service apache2 restart
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html/ecommerce/shop$
```

3.1.3 Modificando el archivo hosts del sistema para acceder a www.miempresa.com

Para poder acceder al sitio mediante el nombre www.miempresa.com, tenemos que modificar el archivo hosts de nuestro sistema, para acceder a este archivo, ejecutamos el comando **sudo gedit /etc/hosts**

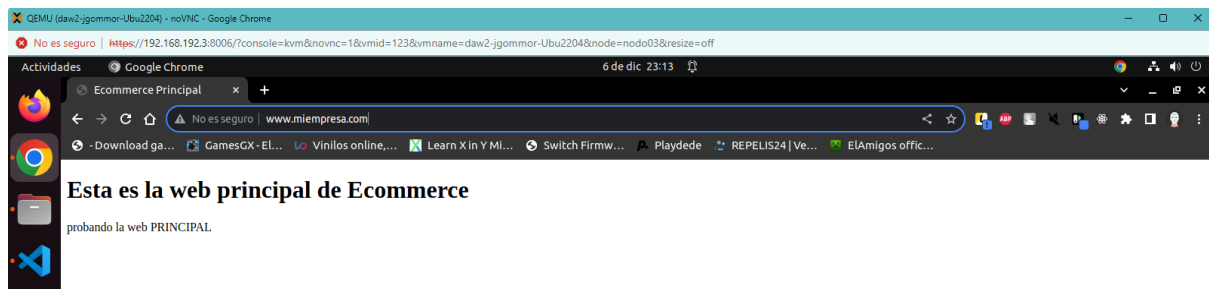
En este archivo, añadimos la ip reservada 127.0.0.1 que nos redirecciona a nuestro localhost seguido del nombre de dominio www.miempresa.com



```
1 127.0.0.1      localhost
2 127.0.1.1      usuario-Standard-PC-i440FX-PIIX-1996
3 127.0.1.1      www.miempresa.com
4 # The following lines are desirable for IPv6 capable hosts
5 ::1            ip6-localhost ip6-loopback
6 fe00::0        ip6-localnet
7 ff00::0        ip6-mcastprefix
8 ff02::1        ip6-allnodes
9 ff02::2        ip6-allrouters
```

Con esto, hacemos que al acceder a www.miempresa.com desde nuestro navegador, accedamos a nuestro localhost por el puerto predeterminado de apache2 que es el 80

Si probamos a acceder, debemos de ver la web que hemos creado



3.1.4 Obtener un certificado SSL

3.1.4.1 Pedir un certificado con certbot para despliegue real

Para añadir **SSL a nuestro servidor apache2** necesitamos un certificado que podemos obtener de manera gratuita con certbot, **nosotros no haremos la petición del certificado** ya que para poder hacerlo, **tenemos que tener acceso al dominio**, para demostrar que es nuestro y pedirlo con certbot desde la máquina que tenemos el hosting, a continuación se muestra como se hace, siguiendo los pasos de la web de certbot.

<https://certbot.eff.org/instructions?ws=apache&os=ubuntu&tab=standard>

Para nuestro entorno local, estos pasos no son necesarios hacerlos

Para ello lo primero que tenemos que hacer es instalar snapd con el comando **sudo apt install snapd**

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$ sudo apt install snapd
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
snapd ya está en su versión más reciente (2.58+22.04.1).
fijado snapd como instalado manualmente.
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
  libflashrom1 libftdi1-2 libllvm13
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 39 no actualizados.
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$
```

ahora, con el comando **snap install --classic certbot** instalamos certbot

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$ snap install --classic certbot
Buscar y comproBar yBuscMontMontarMontaConfigurar los perfiles de seguridad del snap "cer
tbot" (3566)
Configurar los perfi
les de seguridad del snap "certbot" (3566)
Conexión
automática de enchufes y ranuras apropiados del «snap» "EjecuEjecutar el enganche de config
uración deSe ha instalado certbot 2.8.0 por Certbot Project (certbot-eff✓)
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$
```

ejecutamos el comando **sudo ln -s /snap/bin/certbot /usr/bin/certbot** para comprobar que podemos ejecutar certbot

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$ sudo ln -s /snap/bin/certbot /usr/bin/certbot
```

añadimos certbot a apache2 **sudo certbot --apache**

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$ sudo certbot --apache
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security notices)
(Enter 'c' to cancel): jgommor167@g.educaand.es

-----
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.3-September-21-2022.pdf. You must
agree in order to register with the ACME server. Do you agree?
-----
(Y)es/(N)o: Y

-----
Would you be willing, once your first certificate is successfully issued, to
share your email address with the Electronic Frontier Foundation, a founding
partner of the Let's Encrypt project and the non-profit organization that
develops Certbot? We'd like to send you email about our work encrypting the web,
EFF news, campaigns, and ways to support digital freedom.
-----
(Y)es/(N)o: Y
Account registered.

Which names would you like to activate HTTPS for?
We recommend selecting either all domains, or all domains in a VirtualHost/server block.
-----
1: www.coches.com
2: www.filemon.es
3: www.miempresa.com
4: www.pepe.es
-----
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel): 3
Requesting a certificate for www.miempresa.com
```

ejecutamos el comando **sudo certbot renew --dry-run** para que certbot renueve automáticamente los certificados de seguridad

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$ sudo certbot renew --dry-run
Saving debug log to /var/log/letsencrypt/letsencrypt.log

-----
No simulated renewals were attempted.
-----
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$
```

3.1.4.2 Crear un certificado autofirmado para acceder de manera local y agregar el módulo SSL a apache2

Nosotros haremos la configuración SSL para nuestro virtualhost de manera local de la siguiente manera

Instalar el módulo ssl en apache2 con el comando **sudo a2enmod ssl**

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$ 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:
  systemctl restart apache2
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$
```

Creamos un certificado que auto firmamos nosotros mismos con el comando

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/apache-clavemiempresa.com.key -out
/etc/ssl/certs/apache-certificadomiempresa.com.crt
```

Durante este proceso, se te harán algunas preguntas, incluyendo la solicitud de información sobre el certificado. Puedes proporcionar información ficticia ya que este certificado es solo para propósitos de desarrollo.

3.1.5 Creación de un virtualhost puerto 443 para conexiones SSL en www.miempresa.com

Ahora tenemos que configurar un nuevo **virtualhost** para el puerto reservado **443** , que es el puerto estándar para conexiones SSL en nuestro archivo **miempresa.com.conf** ubicado en **/etc/apache2/sites-available**

Añadimos la directiva **ServerName** <https://www.miempresa.com> para que cuando se haga la petición HTTPS utilice la configuración de este VirtualHost

```
ServerName https://www.miempresa.com
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/ecommerce/shop
DirectoryIndex index.html
ErrorDocument 404 "no se ha encontrado la página en nuestro servidor,
práctica ecommerce, página principal"
```

Añadir las directivas de configuración SSL

```
# Configuración para SSL
SSLEngine on
SSLCertificateFile /etc/ssl/certs/certificadomiempresa.com.crt
SSLCertificateKeyFile /etc/ssl/private/clavemiempresa.com.key
```

SSLEngine on: Activa SSL en nuestro virtual host

SSLCertificateFile: Especifica la ruta donde está el archivo con el certificado creado anteriormente

SSLCertificateKeyFile: Especifica la ruta donde está el archivo con la clave creada anteriormente

Nuestro virtualhost para el puerto 443 en el archivo `miempresa.com.conf` queda de la siguiente manera

```
etc > apache2 > sites-available > miempresa.com.conf
32 | #Include conf-available/serve-cgi-bin.conf
33 | </VirtualHost>
34 |
35 | <VirtualHost *:443>
36 |     # The ServerName directive sets the request scheme, hostname and port that
37 |     # the server uses to identify itself. This is used when creating
38 |     # redirection URLs. In the context of virtual hosts, the ServerName
39 |     # specifies what hostname must appear in the request's Host: header to
40 |     # match this virtual host. For the default virtual host (this file) this
41 |     # value is not decisive as it is used as a last resort host regardless.
42 |     # However, you must set it for any further virtual host explicitly.
43 |     #ServerName www.example.com
44 |     ServerName https://www.miempresa.com
45 |     ServerAdmin webmaster@localhost
46 |     DocumentRoot /var/www/html/ecommerce/shop
47 |     DirectoryIndex index.html
48 |     ErrorDocument 404 "no se ha encontrado la página en nuestro servidor, practica ecommerce, pagina principal"
49 |
50 |     # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
51 |     # error, crit, alert, emerg.
52 |     # It is also possible to configure the loglevel for particular
53 |     # modules, e.g.
54 |     #LogLevel info ssl:warn
55 |
56 |     ErrorLog ${APACHE_LOG_DIR}/error.log
57 |     CustomLog ${APACHE_LOG_DIR}/access.log combined
58 |
59 |     #Configuración SSL
60 |     SSLEngine on
61 |     SSLCertificateFile /etc/ssl/certs/apache-certificadomiempresa.com.crt
62 |     SSLCertificateKeyFile /etc/ssl/private/apache-clavemiempresa.com.key
63 |
64 |     # For most configuration files from conf-available/, which are
65 |     # enabled or disabled at a global level, it is possible to
66 |     # include a line for only one particular virtual host. For example the
67 |     # following line enables the CGI configuration for this host only
68 |     # after it has been globally disabled with "a2disconf".
69 |     #Include conf-available/serve-cgi-bin.conf
70 | </VirtualHost>
```

3.1.6 Redirigir las conexiones del puerto 80 al puerto 443 (solo permitir HTTPS) en `www.miempresa.com`

Configuramos que solo se pueda acceder a nuestro servidor mediante peticiones **HTTPS** redirigiendo las peticiones **HTTP** del **virtualhost con puerto 80** **ServerName** www.miempresa.com a **HTTPS** que es nuestro **virtualhost 443** **ServerName** <https://www.miempresa.com> con la siguiente directiva en el archivo ubicado en `etc/apache2/sites-available/miempresa.com.conf`

La siguiente directiva se aplica al **virtualhost puerto 80** para que redirija el puerto 80 a HTTPS

Redirect permanent / https://www.miempresa.com/

Quedándonos nuestro virtualhost puerto 80 de la siguiente manera del archivo `miempresa.com.conf`

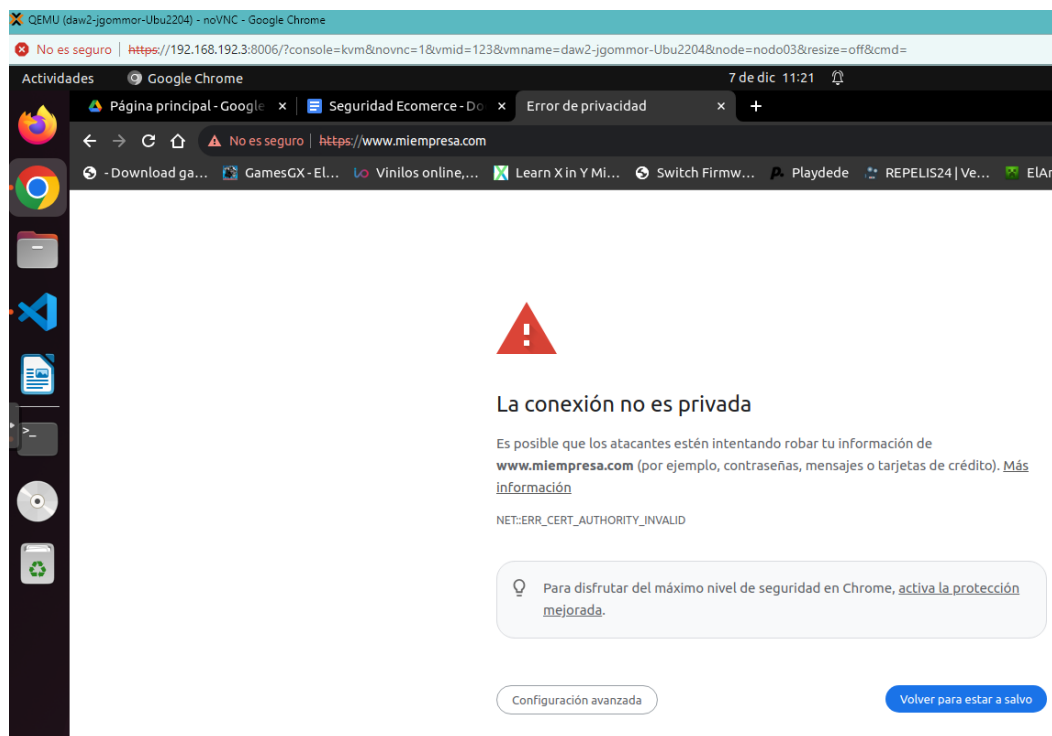
```
etc > apache2 > sites-available > miempresa.com.conf
1  <VirtualHost *:80>
2      # The ServerName directive sets the request scheme, hostname and port that
3      # the server uses to identify itself. This is used when creating
4      # redirection URLs. In the context of virtual hosts, the ServerName
5      # specifies what hostname must appear in the request's Host: header to
6      # match this virtual host. For the default virtual host (this file) this
7      # value is not decisive as it is used as a last resort host regardless.
8      # However, you must set it for any further virtual host explicitly.
9      #ServerName www.example.com
10     ServerName www.miempresa.com
11     ServerAdmin webmaster@localhost
12     DocumentRoot /var/www/html/ecommerce/shop
13     DirectoryIndex index.html
14     ErrorDocument 404 "no se ha encontrado la página en nuestro servidor, practica ecommerce, pagina principal"
15
16     # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
17     # error, crit, alert, emerg.
18     # It is also possible to configure the loglevel for particular
19     # modules, e.g.
20     #LogLevel info ssl:warn
21
22     ErrorLog ${APACHE_LOG_DIR}/error.log
23     CustomLog ${APACHE_LOG_DIR}/access.log combined
24
25     Redirect permanent / https://www.miempresa.com/
26
27     # For most configuration files from conf-available/, which are
28     # enabled or disabled at a global level, it is possible to
29     # include a line for only one particular virtual host. For example the
30     # following line enables the CGI configuration for this host only
31     # after it has been globally disabled with "a2disconf".
32     #Include conf-available/serve-cgi-bin.conf
33 </VirtualHost>
```

Reiniciamos el servicio apache2 con **sudo service apache2 restart**

```
oseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$ sudo service apache2 restart
oseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$
```

3.1.7 Comprobando conexión HTTPS

Comprobamos si podemos acceder a nuestra web mediante SSL con la url <https://www.miempresa.com> o con www.miempresa.com ya que siempre accederemos mediante **HTTPS**



Como podemos observar, conectamos por **HTTPS** pero el navegador nos advierte que la **conexión no es privada**, ya que tenemos un **certificado autofirmado**, pero si hacemos clic en configuración avanzada y pulsamos la url acceder a www.miempresa.com (sitio no seguro) entraremos a la web y podremos visualizarla



Una vez en la web, si hacemos **click en la advertencia de no es seguro**, nos da el motivo por el cual esta web no es segura, y vemos que nos dice que el **certificado no es válido**, si hacemos click en El certificado no es válido, podremos ver la información del certificado



Podemos observar que en la información del certificado contiene la **información que pusimos anteriormente** al crear el certificado **autofirmado**



En la pestaña detalles encontramos el email, y los demás datos que añadimos al crear el certificado

✕

Visor de certificados: emailAddress=prueba,O=practica ecommerce,L=Estepona,ST=Malaga,C=ES

General

Detalles

Jerarquía de certificados

emailAddress=prueba,O=practica ecommerce,L=Estepona,ST=Malaga,C=ES

Campos de certificado

▼ emailAddress=prueba,O=practica ecommerce,L=Estepona,ST=Malaga,C=ES

▼ Certificado

Versión

Número de serie

Algoritmo de firma de certificado

Emisor

▼ Validez

Posterior a

Valor de campo

Exportar...

3.2 Creación del sitio de administración admin.miempresa.com.

- Sitio Virtual 2 (Panel de Administración):
 - Nombre del Sitio: admin.miempresa.com.
 - Directorio Raíz: /var/www/html/ecommerce/admin
 - Configuración Adicional:
 - Implementar autenticación básica para garantizar el acceso restringido.
 - Establecer reglas de acceso para permitir sólo conexiones seguras (HTTPS).
 - NOTA: Para comprobar este sitio virtual, puedes descargarte una plantilla o realizar un html simple, que se diferente a la tienda online.

3.2.1 Configuración de un virtualhost puerto 80 con directiva ServerName www.admin.miempresa.com

Para configurar el segundo sitio virtual , necesitamos crear **otro archivo de configuracion** **admin.miempresa.com.conf** para nuestro sitio en la carpeta **/etc/apache2/sites-available**

Vamos a la carpeta **etc/apache2/sites-available**

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$ cd /etc/apache2/sites-available/  
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/etc/apache2/sites-available$
```

Creamos el archivo admin.[miempresa.com.conf](http://www.admin.miempresa.com) con **sudo touch**
admin.miempresa.com.conf

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/etc/apache2/sites-available$ sudo touch admin.miempresa.com.conf  
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/etc/apache2/sites-available$
```

Pasamos el archivo admin.[miempresa.com.conf](http://www.admin.miempresa.com) a visual studio code y creamos un nuevo virtual host en el **puerto 80** con la directiva **ServerName** www.admin.miempresa.com para que cuando accedemos mediante este nombre de dominio , apache2 use la configuración de este virtualhost para este sitio.

En la directiva DocumentRoot añadimos la ruta de donde se encontraran los archivos del sitio, que es **/var/www/html/ecommerce/admin**

En la directiva ServerName añadimos el nombre de dominio **www.admin.miempresa.com**

En la directiva DirectoryIndex ponemos el nombre del archivo que queremos que se muestre cuando accedemos a nuestra web, en mi caso pondré un archivo **index.html**

En la directiva **ErrorDocument 404** ponemos el mensaje de error al producirse un código de error HTTP 404 , al no encontrar un recurso en nuestro directorio web.

Así quedaría nuestro virtualhost puerto 80 **ServerName** www.admin.miempresa.com del archivo `admin.miempresa.com.conf`

```
etc > apache2 > sites-available > admin.miempresa.com.conf
1 <VirtualHost *:80>
2     # The ServerName directive sets the request scheme, hostname and port that
3     # the server uses to identify itself. This is used when creating
4     # redirection URLs. In the context of virtual hosts, the ServerName
5     # specifies what hostname must appear in the request's Host: header to
6     # match this virtual host. For the default virtual host (this file) this
7     # value is not decisive as it is used as a last resort host regardless.
8     # However, you must set it for any further virtual host explicitly.
9     #ServerName www.example.com
10    ServerName www.admin.miempresa.com
11    ServerAdmin webmaster@localhost
12    DocumentRoot /var/www/html/ecommerce/admin
13    DirectoryIndex index.html
14    ErrorDocument 404 "no se ha encontrado la página en nuestro servidor, practica ecommerce, pagina principal"
15
16    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
17    # error, crit, alert, emerg.
18    # It is also possible to configure the loglevel for particular
19    # modules, e.g.
20    #LogLevel info ssl:warn
21
22    ErrorLog ${APACHE_LOG_DIR}/error.log
23    CustomLog ${APACHE_LOG_DIR}/access.log combined
24
25    # For most configuration files from conf-available/, which are
26    # enabled or disabled at a global level, it is possible to
27    # include a line for only one particular virtual host. For example the
28    # following line enables the CGI configuration for this host only
29    # after it has been globally disabled with "a2disconf".
30    #Include conf-available/serve-cgi-bin.conf
31 </VirtualHost>
```

Tenemos que activar el sitio, para ello ejecutamos el comando **sudo a2ensite** `admin.miempresa.com`, para que se cree el **enlace simbólico** desde el archivo `etc/apache2/sites-available/admin.miempresa.com` a `etc/apache2/sites-enabled/`

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996: ~
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$ sudo a2ensite admin.miempresa.com
[sudo] contraseña para joseangel:
Enabling site admin.miempresa.com.
To activate the new configuration, you need to run:
    systemctl reload apache2
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$
```


3.2.2 Creando los archivos del sitio web www.admin.miempresa.com

Para crear el directorio con los archivos de nuestra web en la ruta especificada en el archivo `admin.miempresa.com.conf` Accedemos a la ruta `/var/www/html/ecommerce`

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996: /var/
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$ cd /var/www/html/ecommerce
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html/ecommerce$
```

Desde aquí ponemos el comando **sudo mkdir admin**

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996: /var/www/html/
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html/ecommerce$ sudo mkdir admin
[sudo] contraseña para joseangel:
Lo siento, pruebe otra vez.
[sudo] contraseña para joseangel:
Lo siento, pruebe otra vez.
[sudo] contraseña para joseangel:
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html/ecommerce$ ls
admin  shop
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html/ecommerce$
```

Accedemos a la carpeta con **cd admin**

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html/ecommerce$ cd admin
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html/ecommerce/admin$
```

Creamos un archivo con **sudo touch index.html**

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html/ecommerce/admin$ sudo touch index.html
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html/ecommerce/admin$
```

Pasamos el archivo **index.html** a nuestro Visual Studio Code y creamos una web que identifique qué es la web de administración

```
000-default.conf  ports.conf  apache2.conf  pepe.es.conf  filemon.es.conf

var > www > html > ecommerce > admin > index.html > html > body > p
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9      <h1>Esta es la web de ADMINISTRACION de la practica 4 ECOMMERCE</h1>
10     <p>Probando el acceso a la web de administración</p>
11 </body>
12 </html>
```

Guardamos los cambios en el archivo y reiniciamos de nuevo apache2 con **sudo service apache2 restart**

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html/ecommerce/admin$ sudo service apache2 restart
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:/var/www/html/ecommerce/admin$
```

3.2.3 Modificando el archivo hosts del sistema para acceder a www.admin.miempresa.com

Para poder acceder al sitio mediante el nombre www.admin.miempresa.com, tenemos que modificar el archivo hosts de nuestro sistema, para acceder a este archivo, ejecutamos el comando **sudo gedit /etc/hosts**

En este archivo, añadimos la ip reservada 127.0.0.1 que nos redirecciona a nuestro localhost seguido del nombre de dominio **www.admin.miempresa.com**



```
1 127.0.0.1    localhost
2 127.0.1.1    usuario-Standard-PC-i440FX-PIIX-1996
3 127.0.1.1    www.miempresa.com
4 127.0.1.1    www.admin.miempresa.com
5 # The following lines are desirable for IPv6 capable hosts
6 ::1         ip6-localhost ip6-loopback
7 fe00::0     ip6-localnet
8 ff00::0     ip6-mcastprefix
9 ff02::1     ip6-allnodes
10 ff02::2     ip6-allrouters
```

Con esto, hacemos que al acceder a <http://www.admin.miempresa.com> desde nuestro navegador, accedamos a nuestro localhost por el puerto predeterminado de apache2 que es el 80

Si probamos a acceder, debemos de ver la web de **administración** que hemos creado



3.2.3 Creación de un virtualhost puerto 443 para conexiones SSL en `www.admin.miempresa.com.conf`

Como ya disponemos de un certificado SSL , vamos a crear un virtualhost para el puerto 443 en nuestro archivo `/etc/apache2/sites-available/admin.miempresa.com.conf`

Añadimos la directiva **ServerName** <https://www.admin.miempresa.com> para que cuando se haga la petición HTTPS utilice la configuración de este VirtualHost

```
ServerName https://www.admin.miempresa.com
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/ecommerce/admin
DirectoryIndex index.html
ErrorDocument 404 "no se ha encontrado la página en nuestro servidor,
práctica ecommerce, página de administracion"
```

Añadir las directivas de configuración SSL

```
# Configuración para SSL
SSLEngine on
SSLCertificateFile /etc/ssl/certs/certificadomiempresa.com.crt
SSLCertificateKeyFile /etc/ssl/private/clavemiempresa.com.key
```

SSLEngine on: Activa SSL en nuestro virtual host

SSLCertificateFile: Especifica la ruta donde está el archivo con el certificado creado anteriormente

SSLCertificateKeyFile: Especifica la ruta donde está el archivo con la clave creada anteriormente

Nuestro virtualhost para el puerto 443 en el archivo `admin.miempresa.com.conf` queda de la siguiente manera

```
etc > apache2 > sites-available > admin.miempresa.com.conf
31 <!-- VirtualHost
32
33 <VirtualHost *:443>
34     # The ServerName directive sets the request scheme, hostname and port that
35     # the server uses to identify itself. This is used when creating
36     # redirection URLs. In the context of virtual hosts, the ServerName
37     # specifies what hostname must appear in the request's Host: header to
38     # match this virtual host. For the default virtual host (this file) this
39     # value is not decisive as it is used as a last resort host regardless.
40     # However, you must set it for any further virtual host explicitly.
41     #ServerName www.example.com
42     ServerName https://www.miempresa.com
43     ServerAdmin webmaster@localhost
44     DocumentRoot /var/www/html/ecommerce/admin
45     DirectoryIndex index.html
46     ErrorDocument 404 "no se ha encontrado la página en nuestro servidor, practica ecommerce, pagina de administracion"
47
48     # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
49     # error, crit, alert, emerg.
50     # It is also possible to configure the loglevel for particular
51     # modules, e.g.
52     #LogLevel info ssl:warn
53
54     ErrorLog ${APACHE_LOG_DIR}/error.log
55     CustomLog ${APACHE_LOG_DIR}/access.log combined
56
57     #Configuracion SSL
58     SSLEngine on
59     SSLCertificateFile /etc/ssl/certs/apache-certificadomiempresa.com.crt
60     SSLCertificateKeyFile /etc/ssl/private/apache-clavemiempresa.com.key
61
62     # For most configuration files from conf-available/, which are
63     # enabled or disabled at a global level, it is possible to
64     # include a line for only one particular virtual host. For example the
65     # following line enables the CGI configuration for this host only
66     # after it has been globally disabled with "a2disconf".
67     #Include conf-available/serve-cgi-bin.conf
68 </VirtualHost>
```

3.2.4 Redirigir las conexiones del puerto 80 al puerto 443 (solo permitir HTTPS) en `www.admin.miempresa.com`

Configuramos que solo se pueda acceder a nuestra web mediante peticiones **HTTPS** redirigiendo las peticiones **HTTP** del **virtualhost con puerto 80** **ServerName** www.admin.miempresa.com a **HTTPS** que es nuestro **virtualhost 443** **ServerName** <https://www.admin.miempresa.com> con la siguiente directiva en el archivo ubicado en `etc/apache2/sites-available/admin.miempresa.com.conf`

La siguiente directiva se aplica al **virtualhost puerto 80** para que redirija el puerto 80 a HTTPS

Redirect permanent / https://www.admin.miempresa.com/

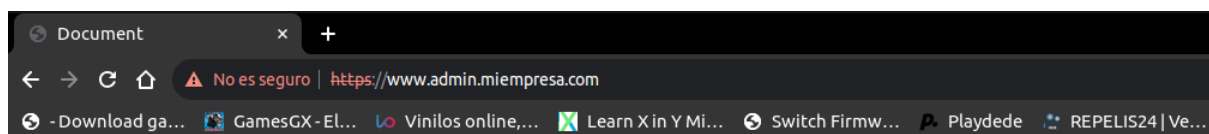
Quedándonos nuestro virtualhost puerto 80 de la siguiente manera del archivo `admin.miempresa.com.conf`

```
etc > apache2 > sites-available > admin.miempresa.com.conf
1 <VirtualHost *:80>
2     # The ServerName directive sets the request scheme, hostname and port that
3     # the server uses to identify itself. This is used when creating
4     # redirection URLs. In the context of virtual hosts, the ServerName
5     # specifies what hostname must appear in the request's Host: header to
6     # match this virtual host. For the default virtual host (this file) this
7     # value is not decisive as it is used as a last resort host regardless.
8     # However, you must set it for any further virtual host explicitly.
9     #ServerName www.example.com
10    ServerName www.admin.miempresa.com
11    ServerAdmin webmaster@localhost
12    DocumentRoot /var/www/html/ecommerce/admin
13    DirectoryIndex index.html
14    ErrorDocument 404 "no se ha encontrado la página en nuestro servidor, practica ecommerce, pagina de administracion"
15
16    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
17    # error, crit, alert, emerg.
18    # It is also possible to configure the loglevel for particular
19    # modules, e.g.
20    #LogLevel info ssl:warn
21
22    ErrorLog ${APACHE_LOG_DIR}/error.log
23    CustomLog ${APACHE_LOG_DIR}/access.log combined
24
25    Redirect permanent / https://admin.miempresa.com/
26    # For most configuration files from conf-available/, which are
27    # enabled or disabled at a global level, it is possible to
28    # include a line for only one particular virtual host. For example the
29    # following line enables the CGI configuration for this host only
30    # after it has been globally disabled with "a2disconf".
31    #Include conf-available/serve-cgi-bin.conf
32 </VirtualHost>
```

Reiniciamos el servicio apache2 con `sudo service apache2 restart`

```
oseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$ sudo service apache2 restart
oseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$
```

Si ahora probamos a acceder a nuestra web www.admin.miempresa.com nos debe redireccionar a **HTTPS**



Esta es la web de ADMINISTRACION de la practica 4 ECOMMERCE

Probando el acceso a la web de administración

Como podemos observar tenemos el mismo problema de que no es seguro porque tenemos un certificado autofirmado

3.2.5 Añadiendo autenticación básica para garantizar el acceso restringido en www.admin.miempresa.com

Primero vamos a crear un archivo llamado **usuarios** que contendrá los **usuarios y las contraseñas** que se permitirán para acceder al sitio web

este archivo lo creamos con **htpasswd** en **/etc/apache2/** con el comando **htpasswd /etc/apache2/usuarios joseangel**

“joseangel” se puede sustituir por cualquier nombre de usuario

hacemos un **sudo htpasswd -c /etc/apache2/usuarios joseangel**

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$ sudo htpasswd -c /etc/apache2/usuarios joseangel
New password: 
```

Al ejecutar este comando, nos pedirá que establezcamos una contraseña para nuestro usuario joseangel

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$ sudo htpasswd -c /etc/apache2/usuarios joseangel
New password:
Re-type new password:
Adding password for user joseangel
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$ 
```

Una vez introducida la contraseña y haberla verificado, ya tenemos nuestro fichero usuarios creado y si observamos , el fichero está oculto al haberlo creado con .usuarios

Para configurar un acceso con autenticación básica tenemos que modificar el archivo de configuración del sitio **/etc/apache2/sites-available/admin.miempresa.com.conf**

En el virtualhost 443 tenemos que añadir la directiva **Directory** para indicar que la carpeta **/var/www/html/ecommerce/admin** va a tener una serie de restricciones

#autenticación básica

```
<Directory /var/www/html/ecommerce/admin>
    AuthType Basic
    AuthName "esto es confidencial"
    AuthUserFile "etc/apache2/usuarios"
    <RequireALL>
        Require user joseangel
    </RequireALL>
</Directory>
```

Quedandonos el fichero de configuración del sitio **admin.miempresa.com.conf** de la siguiente manera:

```
34 <VirtualHost *:443>
35     # The ServerName directive sets the request scheme, hostname and port that
36     # the server uses to identify itself. This is used when creating
37     # redirection URLs. In the context of virtual hosts, the ServerName
38     # specifies what hostname must appear in the request's Host: header to
39     # match this virtual host. For the default virtual host (this file) this
40     # value is not decisive as it is used as a last resort host regardless.
41     # However, you must set it for any further virtual host explicitly.
42     #ServerName www.example.com
43     ServerName https://admin.www.miempresa.com
44     ServerAdmin webmaster@localhost
45     DocumentRoot /var/www/html/ecommerce/admin
46     DirectoryIndex index.html
47     ErrorDocument 404 "no se ha encontrado la página en nuestro servidor, practica ecommerce, pagina de administracion"
48
49     #Autenticacion Basica
50     <Directory /var/www/html/ecommerce/admin>
51     AuthType Basic
52     AuthName "esto es confidencial"
53     AuthUserFile "/etc/apache2/.usuarios"
54     <RequireALL>
55         Require user joseangel
56     </RequireALL>
57     </Directory>
58
59     # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
60     # error, crit, alert, emerg.
61     # It is also possible to configure the loglevel for particular
62     # modules, e.g.
63     #LogLevel info ssl:warn
64
65     ErrorLog ${APACHE_LOG_DIR}/error.log
66     CustomLog ${APACHE_LOG_DIR}/access.log combined
67
68     #Configuracion SSL
69     SSLEngine on
70     SSLCertificateFile /etc/ssl/certs/apache-certificadomiempresa.com.crt
71     SSLCertificateKeyFile /etc/ssl/private/apache-clavemiempresa.com.key
72
73     # For most configuration files from conf-available/, which are
74     # enabled or disabled at a global level, it is possible to
75     # include a line for only one particular virtual host. For example the
76     # following line enables the CGI configuration for this host only
77     # after it has been globally disabled with "a2disconf".
78     #Include conf-available/serve-cgi-bin.conf
79 </VirtualHost>
```

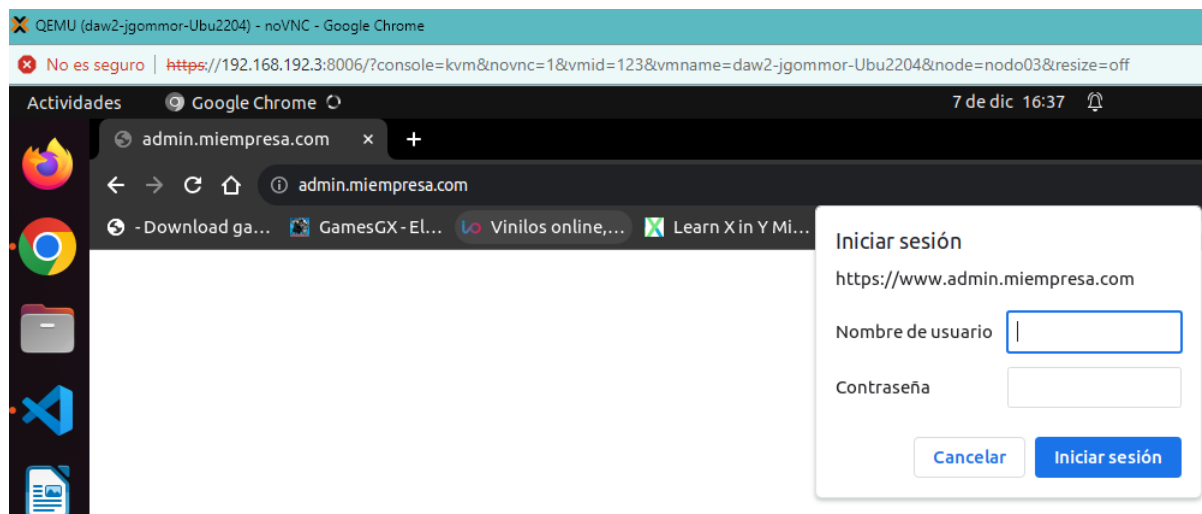
Tenemos que darle permisos de lectura para otros al archivo oculto **.usuarios** con el comando **sudo chmod o+r etc/apache2/.usuarios**

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$ sudo chmod o+r /etc/apache2/.usuarios
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$
```

Reiniciamos el servicio apache2 con el comando **sudo service apache2 restart**

```
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$ sudo service apache2 restart
joseangel@usuario-Standard-PC-i440FX-PIIX-1996:~$
```


Si ahora intentamos acceder a la web <https://www.admin.miempresa.com> nos pedirá la **autenticación básica**



Introducimos el usuario **joseangel** y la contraseña **usuario** que es la definida en el archivo oculto **.usuarios** que hemos creado

