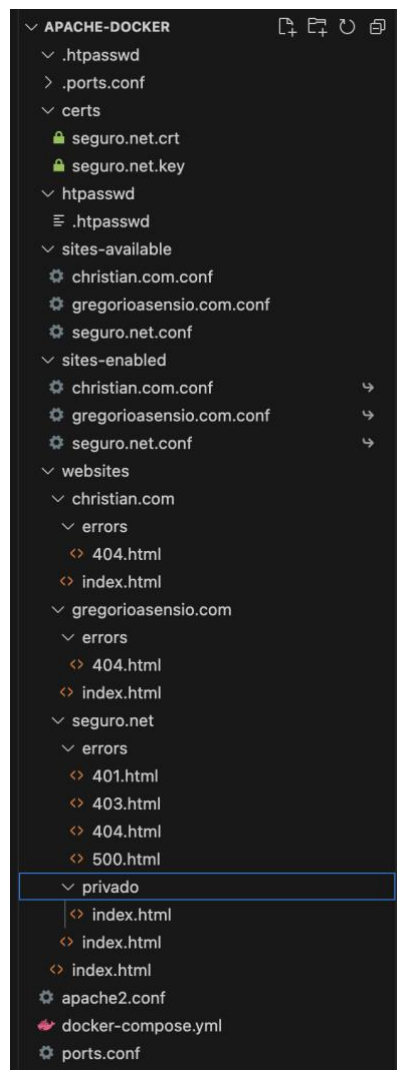


Servidor web usando Apache y Docker.

Inicialmente sabemos que la configuración de Apache está distribuida en varios archivos y directorios para facilitar su gestión y modularidad. La estructura es la siguiente:

```
/etc/apache2/  
|-- apache2.conf  
|  `-- ports.conf  
|-- mods-enabled  
|  |-- *.load  
|  |-- *.conf  
|-- conf-enabled  
|  `-- *.conf  
`-- sites-enabled  
    `-- *.conf
```

Nuestra estructura final de nuestro proyecto será:



Explicación de cada archivo y directorio:

- **apache2.conf:** Archivo de configuración principal de Apache donde se realizan cambios generales.
- **envvars:** Configuración de las variables de entorno.
- **ports.conf:** Configuración de los puertos en los que Apache escucha.
- **conf-available:** Ficheros de configuración adicionales para varios aspectos de Apache o aplicaciones web.
- **conf-enabled:** Enlaces simbólicos a los ficheros de configuración adicionales para activarlos. Se gestionan con los comandos `a2enconf` y `a2disconf`.
- **mods-available:** Módulos disponibles para usar con Apache.
- **mods-enabled:** Enlaces simbólicos a los módulos de Apache que están activados.
- **sites-available:** Ficheros de configuración de hosts virtuales disponibles. Se gestionan con los comandos `a2enmod` y `a2dismod`.
- **sites-enabled:** Enlaces simbólicos a los ficheros de configuración de hosts virtuales activos. Se gestionan con los comandos `a2ensite` y `a2dissite`.

Empezamos con la configuración principal:

Una vez creado el directorio en el que trabajaremos, en este caso:
`/Users/christian/apache-docker`

Vamos a añadir la configuración inicial con los archivos necesarios:

apache2.conf:

Archivo principal de configuración con directivas generales. Incluye configuraciones como:

```
<Directory /var/www/>
    Options Indexes FollowSymlinks
    AllowOverride None
    Require all granted
</Directory>
AccessFileName .htaccess # nombre del archivo de configuración
Include ports.conf # incluye el archivo de configuración de puertos
KeepAlive On # mantiene la conexión abierta
```

Aquí hemos incluido adicionalmente los sitios habilitados:

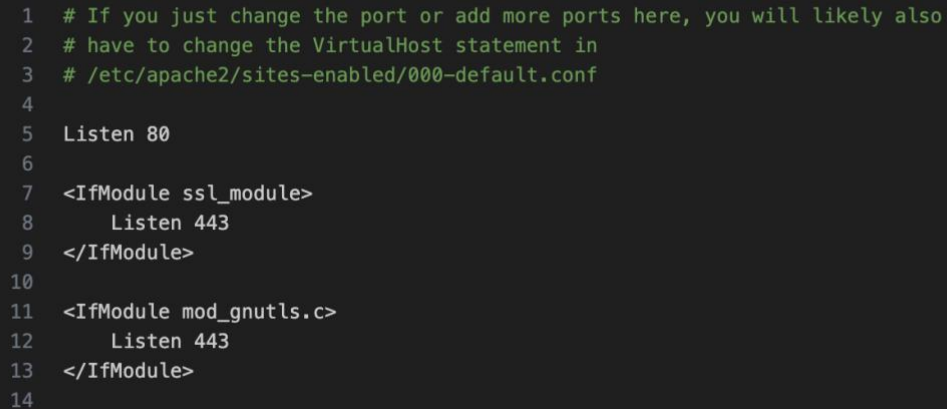
```
# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf
```

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

```
# Incluir los sitios habilitados
IncludeOptional /etc/apache2/sites-enabled/*.conf
```

ports.conf:

Configura los puertos en los que Apache escucha, que más tarde podemos modificar.



```
1  # If you just change the port or add more ports here, you will likely also
2  # have to change the VirtualHost statement in
3  # /etc/apache2/sites-enabled/000-default.conf
4
5  Listen 80
6
7  <IfModule ssl_module>
8      Listen 443
9  </IfModule>
10
11 <IfModule mod_gnutls.c>
12     Listen 443
13 </IfModule>
14
```

Host Virtuales

Los hosts virtuales permiten alojar múltiples sitios web en un solo servidor. Cada host virtual tiene su propio directorio raíz, configuración y archivos de registro. Por defecto, Apache tiene un host virtual configurado en `/etc/apache2/sites-available/000-default.conf`.

Puedes añadir nuevos hosts virtuales creando archivos de configuración en el directorio `/etc/apache2/sites-available/` y activándolos con los comandos `a2ensite` y `a2dissite`.

La estructura de un archivo de configuración de host virtual es la siguiente:

```
<VirtualHost *:80>
    ServerAdmin joseluis.gonzalez@iesluisvives.org # dirección de correo del administrador
    DocumentRoot /var/www/ dominio.com # directorio raíz del sitio web
    ServerName dominio.com # nombre del servidor
    ServerAlias www.dominio.com # alias del servidor

    #LogLevel info ssl:warn

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

    <Directory /var/www/ dominio.com>
        Options Indexes FollowSymLinks # opciones del directorio
        AllowOverride All # habilita la reescritura de URL
        Require all granted # permisos de acceso
    </Directory>
</VirtualHost>
```

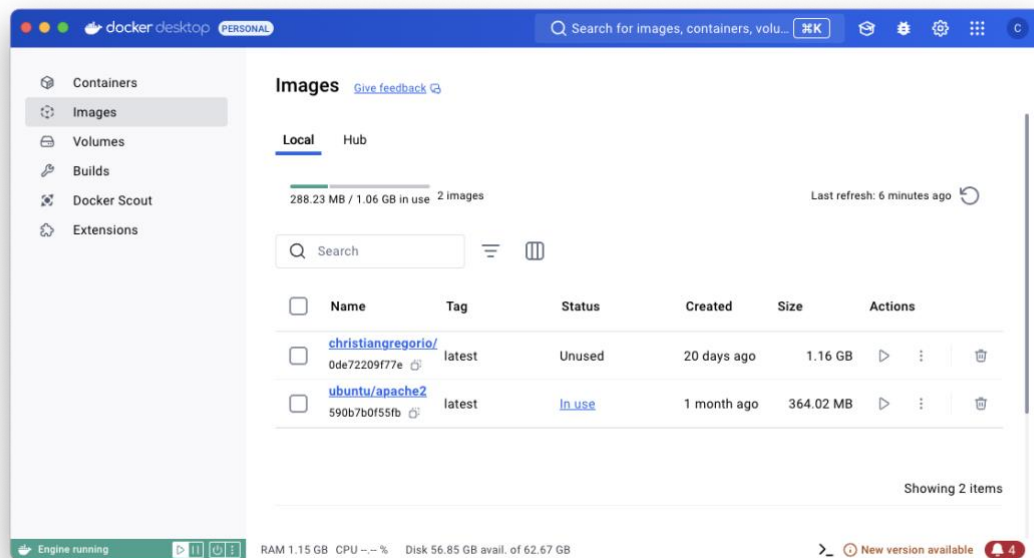
Sites Available/Enables

- sites-available: Directorio que contiene los archivos de configuración de los hosts virtuales disponibles.
- sites-enabled: Directorio que contiene enlaces simbólicos a los archivos de configuración de los hosts virtuales activos.
-

Una vez ya hecho todo, pasamos a ejecutarlo en Docker:

Ejecutamos en nuestro ordenador Docker desktop para que corra el servicio:

```
apache-docker — -zsh — 67x15
Last login: Wed Oct 23 12:32:55 on ttys001
christian@Mac apache-docker % docker-compose up -d
[+] Running 4/4
  ✓ web Pulled                                4.5s
    ✓ 5efb307521c7 Download complete          0.7s
    ✓ 5cce572b52b6 Download complete          1.9s
    ✓ b8142b74ea46 Download complete          2.1s
[+] Running 2/2
  ✓ Network apache-docker_default Created      0.0s
  ✓ Container apache_server Started           0.4s
christian@Mac apache-docker %
```



Nuestro Docker-compose.yml final:

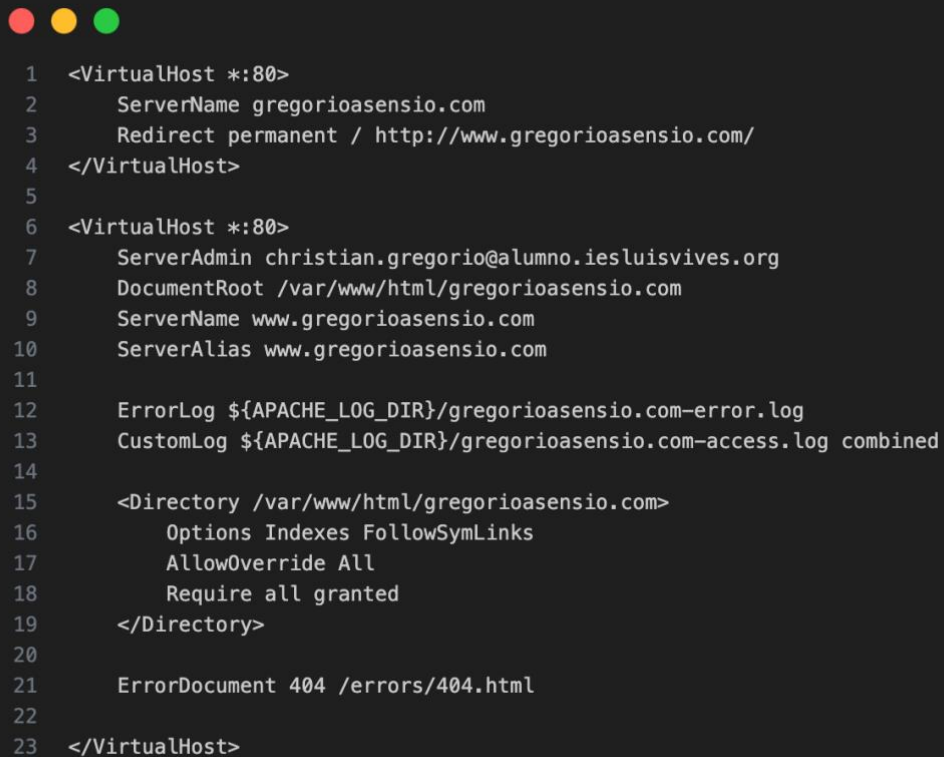
```
1 services:
2   web:
3     image: ubuntu/apache2 # imagen de Apache
4     container_name: apache_server # nombre del contenedor
5     ports:
6       - "8080:80" # mapeo de puertos
7       - "8443:443" # para HTTPS
8     volumes:
9       - ./apache2.conf:/etc/apache2/apache2.conf # archivo de configuración principal
10      - ./sites-available:/etc/apache2/sites-available # archivos de configuración de hosts virtuales
11      - ./sites-enabled:/etc/apache2/sites-enabled
12      - ./websites:/var/www/html/ # directorio de los sitios web
13      - ./ports.conf:/etc/apache2/ports.conf # archivo de configuración de puertos
14      - ./htpasswd:/etc/apache2/.htpasswd # archivo de contraseñas
15      - ./certs:/etc/apache2/certs # directorio de certificados (hechos con openssl)
16    restart: always # reinicio automático
17    # comando para activar los hosts virtuales y arrancar Apache
18
```

Archivos .conf de sites-enabled:

Del dominio: christian.com

```
1 <VirtualHost *:80>
2   ServerAdmin christian.gregorio@alumno.iesluisvives.org
3   DocumentRoot /var/www/html/christian.com
4   ServerName christian.com
5   ServerAlias www.christian.com
6
7   ErrorLog ${APACHE_LOG_DIR}/christian.com-error.log
8   CustomLog ${APACHE_LOG_DIR}/christian.com-access.log combined
9
10  <Directory /var/www/html/christian.com>
11    Options Indexes FollowSymLinks
12    AllowOverride All
13    Require all granted
14  </Directory>
15
16  ErrorDocument 404 /errors/404.html
17
18 </VirtualHost>
```

Dominio: gregorioasensio.com ó www.gregorioasensio.com



```
1 <VirtualHost *:80>
2     ServerName gregorioasensio.com
3     Redirect permanent / http://www.gregorioasensio.com/
4 </VirtualHost>
5
6 <VirtualHost *:80>
7     ServerAdmin christian.gregorio@alumno.iesluisvives.org
8     DocumentRoot /var/www/html/gregorioasensio.com
9     ServerName www.gregorioasensio.com
10    ServerAlias www.gregorioasensio.com
11
12    ErrorLog ${APACHE_LOG_DIR}/gregorioasensio.com-error.log
13    CustomLog ${APACHE_LOG_DIR}/gregorioasensio.com-access.log combined
14
15    <Directory /var/www/html/gregorioasensio.com>
16        Options Indexes FollowSymLinks
17        AllowOverride All
18        Require all granted
19    </Directory>
20
21    ErrorDocument 404 /errors/404.html
22
23 </VirtualHost>
```

Dominio SSL: seguro.net

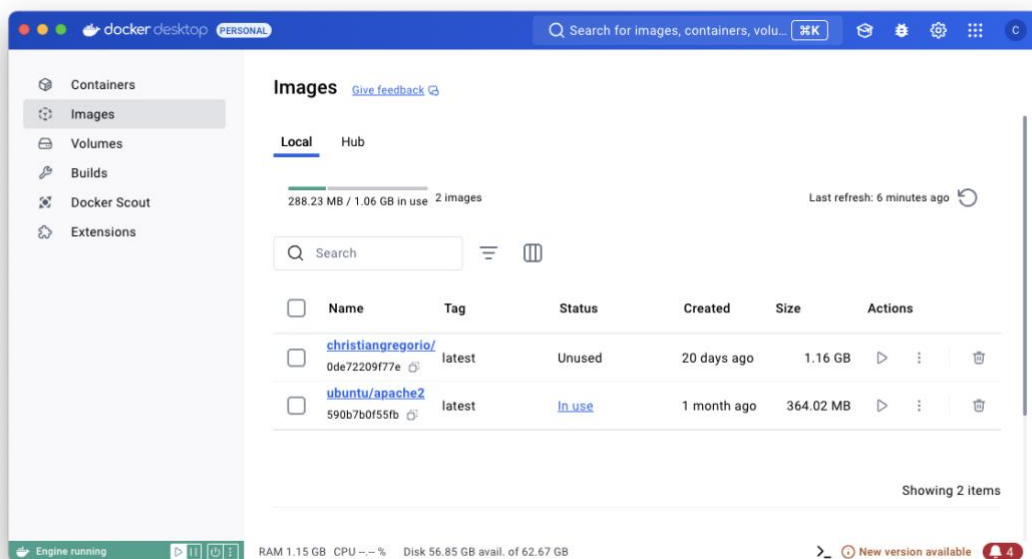


```
1 <VirtualHost *:80>
2     ServerName seguro.net
3     ServerAlias www.seguro.net
4     Redirect permanent / https://seguro.net/
5 </VirtualHost>
6
7 <VirtualHost *:443>
8     ServerAdmin christian.gregorio@alumno.iesluisvives.org
9     DocumentRoot /var/www/html/seguro.net
10    ServerName seguro.net
11    ServerAlias www.seguro.net
12
13    # Configuración SSL
14    SSLEngine On
15    SSLCertificateFile /etc/apache2/certs/seguro.net.crt
16    SSLCertificateKeyFile /etc/apache2/certs/seguro.net.key
17
18    # Habilitar protocolos seguros
19    SSLProtocol All -SSLv3
20
21    # Protección de directorio
22    <Directory "/var/www/html/seguro.net/privado">
23        AuthType Basic
24        AuthName "Acceso Restringido a Usuarios"
25        AuthUserFile /etc/apache2/.htpasswd
26        Require valid-user
27        Options -Indexes
28    </Directory>
29
30    ErrorDocument 401 /errors/401.html
31    ErrorDocument 403 /errors/403.html
32    ErrorDocument 404 /errors/404.html
33    ErrorDocument 500 /errors/500.html
34
35 </VirtualHost>
```

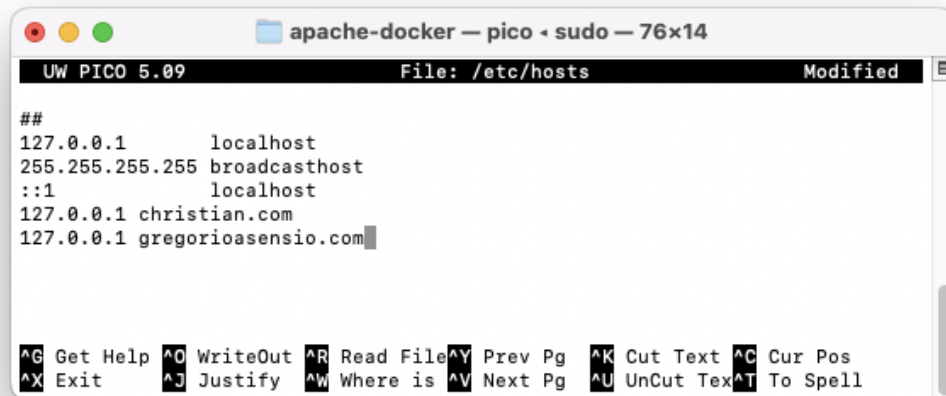
Ahora solo tendríamos que ejecutar el comando `docker-compose up -d` para desplegar el servidor web Apache en un contenedor Docker.

`docker-compose up -d`

```
apache-docker — -zsh — 67x15
Last login: Wed Oct 23 12:32:55 on ttys001
christian@Mac apache-docker % docker-compose up -d
[+] Running 4/4
  ✓ web Pulled                                4.5s
  ✓ 5efb307521c7 Download complete           0.7s
  ✓ 5cce572b52b6 Download complete           1.9s
  ✓ b8142b74ea46 Download complete           2.1s
[+] Running 2/2
  ✓ Network apache-docker_default Created     0.0s
  ✓ Container apache_server Started           0.4s
christian@Mac apache-docker %
```



Mediante la terminal añadiremos al archivo /etc/hosts las entradas:



```
UW PICO 5.09 File: /etc/hosts Modified
##
127.0.0.1 localhost
255.255.255.255 broadcasthost
::1 localhost
127.0.0.1 christian.com
127.0.0.1 gregorioasensio.com
```

```
127.0.0.1 christian.com
127.0.0.1 gregorioasensio.com
127.0.0.1 www.gregorioasensio.com
127.0.0.1 seguro.net
127.0.0.1 www.seguro.net
```

Accedemos con el commando:
`sudo nano /etc/hosts`

Y guardamos los cambios presionando Ctrl + O y Enter para confirmar.
Comprobamos los cambios con:
`cat /etc/hosts`

Pasamos a activar nuestros host virtuales:

Para activar un host virtual, una vez creados los archivos e introducidos en sites-available, los activamos con el comando `a2ensite`:

Ahora entramos al contenedor y activamos los dominios virtuales y reiniciamos apache

```
docker exec -it apache_server bash
```

Activamos los siguientes dominios:

```
a2ensite christian.com.conf
a2ensite gregorioasensio.com.conf
a2ensite seguro.net.conf
```

Y reiniciamos:
`service apache2 reload`

```
(( Para desactivar un host virtual, puedes usar el comando a2dissite. Por ejemplo:  
sudo a2dissite mi-sitio.conf  
sudo service apache2 reload  
))
```

Modificación para que se acceda mediante www.gregorioasensio.com y que se muestre www.gregorioasensio.com:

(Ver en captura correspondiente de este dominio)

De este modo hará que:

- Si accedes a gregorioasensio.com, serás redirigido automáticamente a www.gregorioasensio.com, y la URL cambiará en la barra del navegador.
- Si accedes directamente a www.gregorioasensio.com, se mostrará esa URL sin cambios.

Explicación de la configuración:

a) **Primer bloque <VirtualHost>:**

- o **ServerName gregorioasensio.com:** Este bloque captura las solicitudes que llegan a gregorioasensio.com.
- o **Redirect permanent / http://www.gregorioasensio.com/:** Redirige todas las solicitudes a www.gregorioasensio.com. Esta redirección es permanente y cambiará la URL en el navegador.

b) **Segundo bloque <VirtualHost>:**

- o **ServerName www.gregorioasensio.com:** Este bloque maneja las solicitudes que llegan a www.gregorioasensio.com.
- o **DocumentRoot /var/www/html/gregorioasensio.com:** Este es el directorio donde se sirve el contenido para www.gregorioasensio.com.
- o **ServerAlias www.gregorioasensio.com:** Aunque este alias no es estrictamente necesario aquí, se puede incluir por claridad.

Error 404

Dentro de `/websites/christian.com`, he añadido la carpeta **errors** con su correspondiente `404.html` para que lo lance.

También se debe añadir al archivo `.conf` de cada dominio la línea:

```
ErrorDocument 404 /errors/404.html
```

Ahora tanto el dominio `christian.com` tanto como www.gregorioasensio.com lanzarán el error 404.

Creando una web de tipo <https://seguro.net>

Instalamos en el equipo openssl:

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

En el directorio donde estoy trabajando, creo la carpeta certs, donde openssl me generará lo necesario:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout seguro.net.key -out  
seguro.net.crt
```

Introducimos de usuario “Christian” y email:
“christian.gregorio@alumno.iesluisvives.com”

Nos generará los certificados en la carpeta certs

Ajustar la seguridad de nuestro Virtual Host añadiendo las siguientes líneas al archivo de configuración del host virtual:

Dentro de Docker, instalamos el módulo:

```
docker exec -it apache_server bash
```

```
a2enmod ssl
```

Añadimos el port 443 en el **Docker-compose.yml**

ports:

- "8080:80" # para HTTP
- "443:443" # para HTTPS

Como anteriormente, añadido al hosts del sistema el dominio que estoy usando como privado.

Añado:

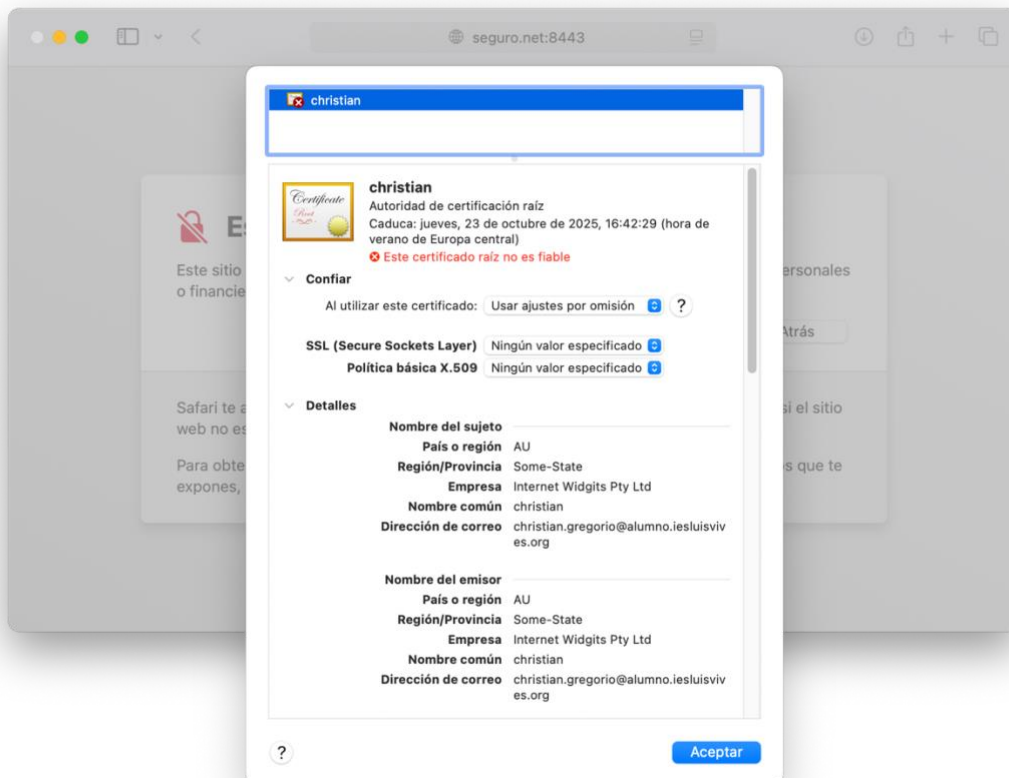
```
127.0.0.1 seguro.net  
127.0.0.1 www.seguro.net
```

Habilito el n2site correspondiente con la nueva web segura metiéndonos en el contenedor con:

```
docker exec -it apache_server bash
```

Accediendo a un directorio o dominio protegido:

Podemos proteger un dominio o directorio de un dominio gracias a Apache.



Accediendo a la web privada:



Para crear el usuario y contraseña, introducimos en la terminal:

Creamos el directorio htpasswd.

Dentro de la terminal del contenedor, debemos ejecutar el comando:

```
docker exec -it apache_server bash
```

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

Siguiendo los pasos que nos indica, configuramos usuario y contraseña, al introducirla nos mostrará el html que hemos creado dentro del directorio Privado y que antes no era visible.