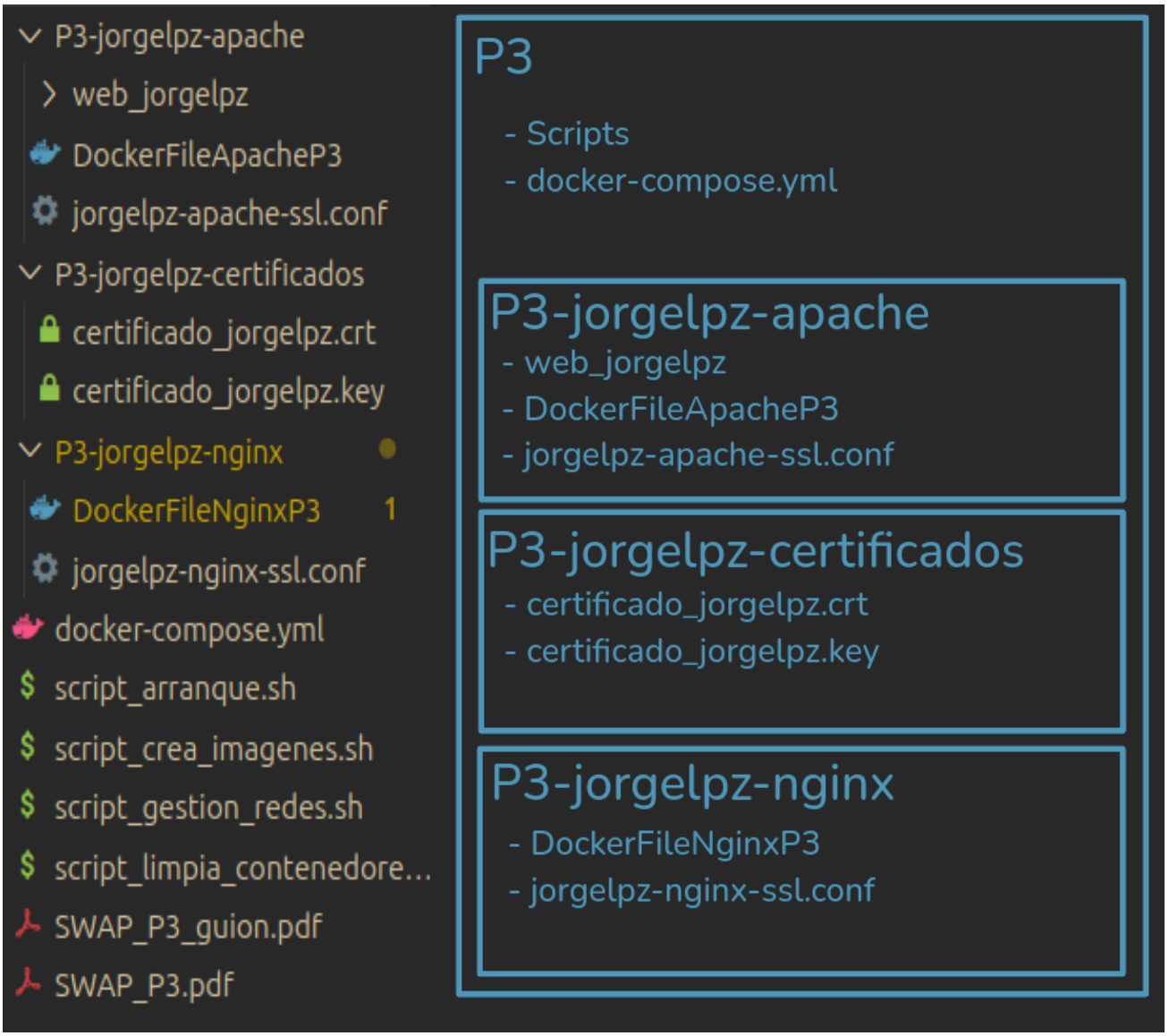
UNIVERSIDAD DE GRANADA

Memoria P3 SWAP
Jorge López Molina 3º TI grupo 3
Tiempo empleado: 13 horas 30 minutos
(Ejecutar scritp_arranque.sh)

Tareas Básicas - B1: Preparación del Entorno de Trabajo.

De cara a crear un entorno adecuado para la realización de la práctica se va a crear la siguiente estructura de directorios:



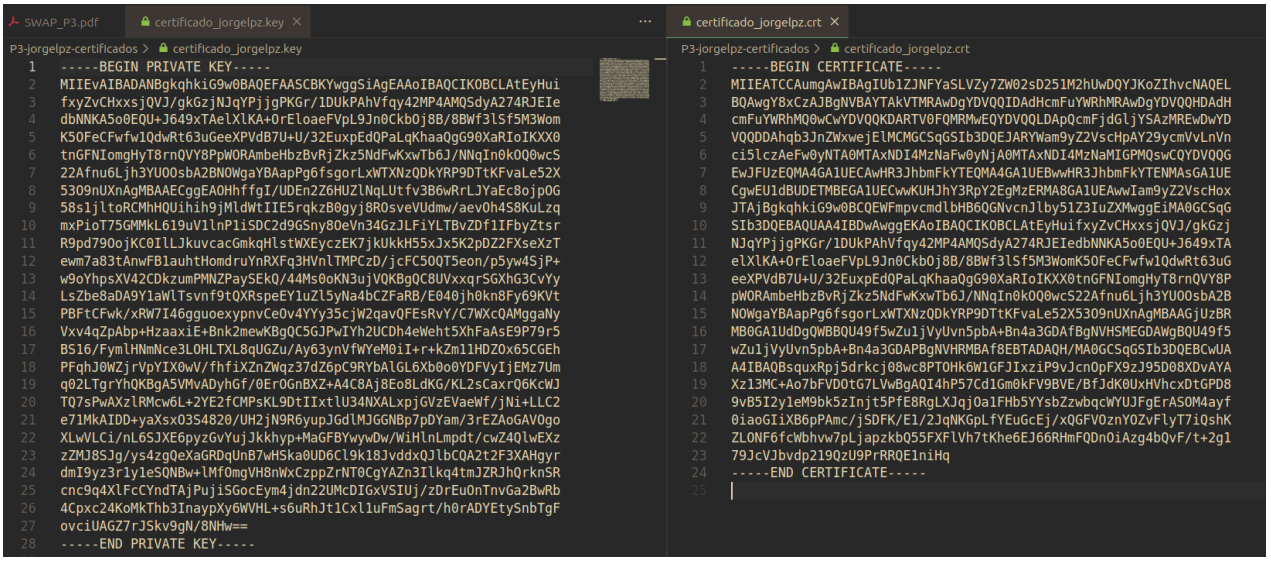
Tareas Básicas - B2: Creación de certificados SSL.

El comando usado para generar un certificado y su clave privada es:

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

Se nos pide explícitamente que las claves tengan unas características concretas:
- x509 para Generar un certificado.
- days 365 para que tenga Validez de 1 año.
- newkey rsa:2048 para usar la Encriptación RSA de 2048 bits.
- keyout para indicarle que No sea necesaria una passphrase.
- certificado_jorgelpz.key almacenará la Clave privada.
- certificado_jorgelpz.crt será el archivo Certificado autofirmado.

Al ejecutar esta sentencia y rellenar el formulario que nos pide para poder crear el certificado de dominio. Obtenemos lo siguiente:

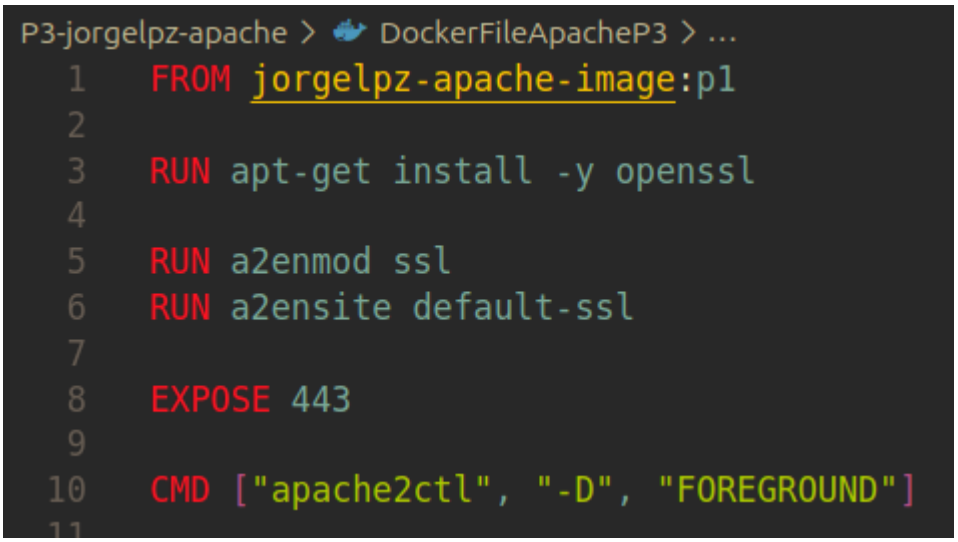


Como podemos observar el certificado y la clave han sido creados correctamente.

Tareas Básicas - B3: Configuración de Servidores Web Apache con SSL.

Si bien en el guión se nos recomienda una estructura para el Dockerfile, yo he decidido optar por un planteamiento distinto. En lugar de colocar los archivos de configuración en el Dockerfile mediante COPY lo hago usando volumes: en el docker-compose. De esta forma se mejora la flexibilidad de la imagen. Además me permite probar los cambios sin necesidad de recrear la imagen cada vez.

Mi Dockerfile para Apache con ssl es el siguiente:



La imagen se crea a partir de jorgelpz-apache-image:p1. Procedo a instalar openssl (línea 3). Habilito el módulo de Apache ssl usando a2enmod (línea 5) y el sitio ssl (línea 6).

Por otra parte, para configurar el archivo de host virtual de Apache para atender peticiones HTTPS partí de una plantilla básica sobre la que añadí los cambios especificados en el guión:

P3-jorgelpz-apache > jorgelpz-apache-ssl.conf

```
<VirtualHost *:443>
    ServerAdmin webmaster@localhost

    DocumentRoot /var/www/html

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

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

    # 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

    #   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/apache2/ssl/certificado_jorgelpz.crt
    SSLCertificateKeyFile   /etc/apache2/ssl/certificado_jorgelpz.key
```

Habilitamos SSLEngine e indicamos la dirección del certificado y de la key dentro de los contenedores.

Tareas Básicas - B4: Configuración del Balanceador de Carga Nginx con SSL.

El Dockerfile para crear la imagen del contenedor que hace de balanceador de carga Nginx es el siguiente:

```
1   FROM nginx:latest
2
3   EXPOSE 443
4
5   CMD ["nginx", "-g", "daemon off;"]
6
```

Por lo explicado anteriormente he dejado toda la parte de copia de archivos para realizarla en el docker-compose. Por ello en este Dockerfile lo único que hago es habilitar el puerto 443.

Para la construcción del archivo de configuración parto del nginx.conf de la práctica 2. He sustituido la parte de la etiqueta server donde antes le especificaba que atendiera peticiones HTTP por lo siguiente:

```
listen 443 ssl;
ssl_certificate /etc/nginx/ssl/certificado_jorgelpz.crt;
ssl_certificate_key /etc/nginx/ssl/certificado_jorgelpz.key;

location / {
    proxy_pass http://backend_jorgelpz;
    proxy_set_header Cookie $http_cookie;
    proxy_hide_header Set-Cookie;
}

location /estadisticas_jorgelpz {
    stub_status on;
}
```

Para atender peticiones HTTPS le fijo que escuche en el puerto 443. Además, le especifico la dirección del certificado público que autentica la conexión. Por último, le indico la ruta a la clave privada asociada al certificado SSL. El resto del archivo permanece sin modificaciones.

Tareas Básicas - B5: Docker Compose para la Granja Web con SSL.

Para el desarrollo del docker-compose he optado por unificar todo el despliegue en un único archivo a diferencia de como lo planteé para la práctica 2.

Los contenedores webX se despliegan de la siguiente forma:

```
▷ Run Service
web1:
    image: jorgelpz-apache-image:p3
    container_name: web1
    volumes:
    - ./P3-jorgelpz-apache/web_jorgelpz:/var/www/html
    - ./P3-jorgelpz-certificados:/etc/apache2/ssl
    - ./P3-jorgelpz-apache/jorgelpz-apache-ssl.conf:/etc/apache2/sites-available/jorgelpz-apache-ssl.conf
    command:
        /bin/sh -c "chmod 600 /etc/apache2/ssl/certificado_jorgelpz.crt && apachectl -D FOREGROUND"
    networks:
        red_web:
            ipv4_address: 192.168.10.2
        red_servicios:
            ipv4_address: 192.168.20.2
```

Son creados a partir de la imagen de Apache para la práctica 3 generada con el Dockerfile de la tarea básica 3. Es aquí donde monto los archivos de configuración necesarios y el propio directorio web_jorgelpz que contiene la web. Hacerlo de esta forma me permite comprobar las modificaciones en las configuraciones solamente ejecutando de nuevo el docker-compose sin necesidad de recargar las imágenes.

En el ejemplo de Dockerfile para Apache del guión usa RUN para ejecutar chmod dentro de los contenedores. Todo esto con el fin de darle permisos al usuario y denegarlos para el resto de usuarios del sistema al archivo certificado_jorgelpz.crt. Con ayuda de la IA conseguí replicar este comportamiento dentro del docker-compose usando la directiva command:. De esta forma ejecuto dentro de cada contenedor al momento de arrancarlos chmod 600 para el archivo especificado y apachectl -D FOREGROUND para que el contenedor no se cierre de inmediato al terminar. Por último le asigno las IPs correspondientes dentro de las subredes red_web y red_servicios como en las prácticas anteriores.

Como se puede observar no asigno ningún mapeo de puertos a estos contenedores para que no sean accesibles si no es a través del balanceador de carga.

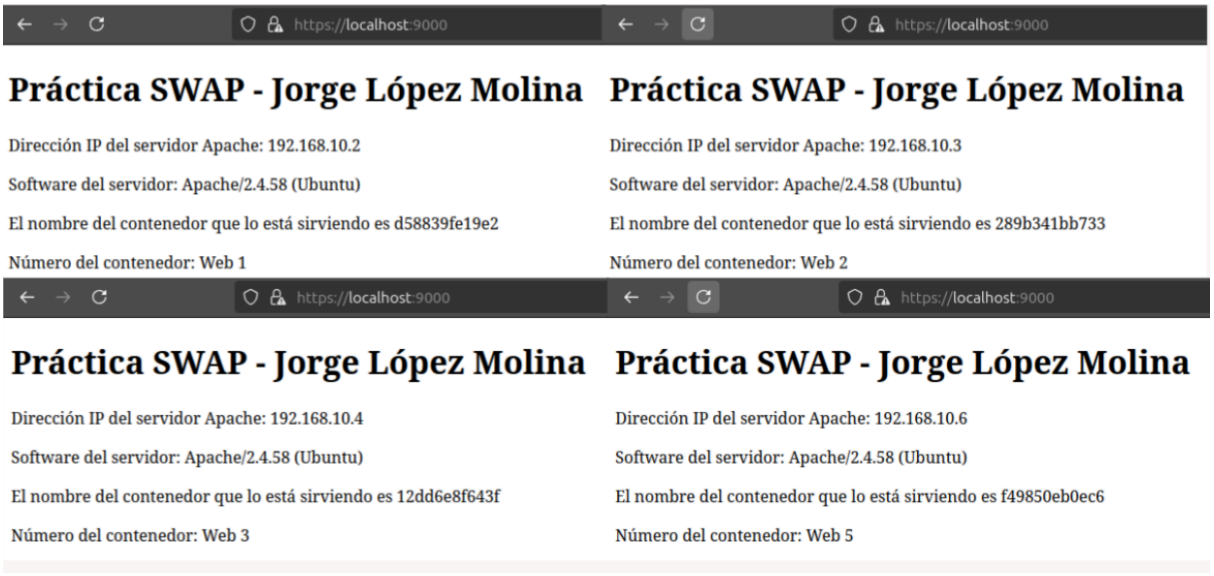
Para lanzar el contenedor balanceador de carga con Nginx la configuración es la siguiente:

```
balanceador-nginx-ssl:
    image: jorgelpz-nginx-image:p3
    container_name: balanceador-nginx-ssl
    ports:
    - "9000:443"
    volumes:
    - ./P3-jorgelpz-nginx/jorgelpz-nginx-ssl.conf:/etc/nginx/nginx.conf
    - ./P3-jorgelpz-certificados:/etc/nginx/ssl
    networks:
        red_web:
            ipv4_address: 192.168.10.50
    depends_on:
        - web1
        - web2
        - web3
        - web4
        - web5
        - web6
        - web7
        - web8
```

El contenedor se lanza con la imagen de nginx de la práctica 3. En las prácticas anteriores el mapeo de puertos se hacía con el puerto 80 del contenedor. Monto el que será el nuevo nginx.conf y los certificados dentro de la ruta /etc/nginx/ssl. Asigno la ip indicada en el guión para el repartidor de carga. Por último usando la clave depends_on: le especifico que en el orden de arranque el balanceador de carga debe ser el último.

Tareas Básicas - B6: Verificación y Pruebas del Escenario con SSL.

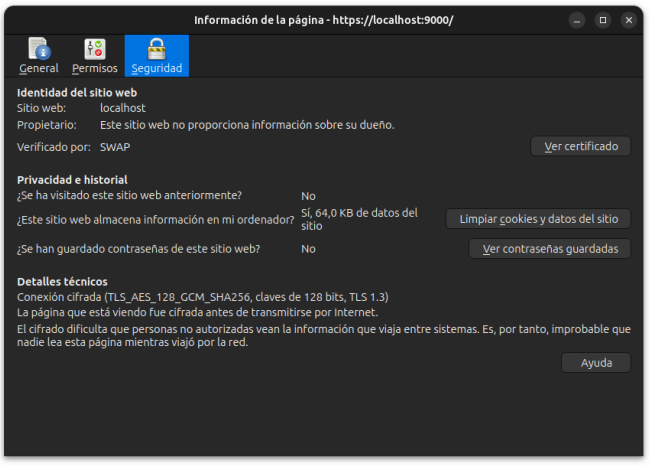
Para probar la nueva configuración tenemos que acceder a https://localhost:9000. Comprobamos que el balanceador de carga funciona correctamente:



Se ajusta a la siguiente estrategia de balanceo:

```
least_conn;
server 192.168.10.2 max_fails=1 fail_timeout=60s;
server 192.168.10.3 max_fails=1 fail_timeout=60s;
server 192.168.10.4 max_fails=1 fail_timeout=60s;
server 192.168.10.5 backup max_fails=1 fail_timeout=60s;
server 192.168.10.6 max_fails=1 fail_timeout=60s;
server 192.168.10.7 max_fails=1 fail_timeout=60s;
server 192.168.10.8 max_fails=1 fail_timeout=60s;
server 192.168.10.9 max_fails=1 fail_timeout=60s;
```

Si comprobamos el certificado podemos observar que está activo:



Certificado

jorgelpz

Nombre del asunto

País	ES
Estado/Provincia	Granada
Localidad	Granada
Organización	SWAP
Unidad organizativa	Practica 3
Nombre común	jorgelpz
Dirección de correo electrónico	jorgelpz@correo.ugr.es

Nombre del emisor

País	ES
Estado/Provincia	Granada
Localidad	Granada
Organización	SWAP
Unidad organizativa	Practica 3
Nombre común	jorgelpz
Dirección de correo electrónico	jorgelpz@correo.ugr.es

Validez

No antes	Thu, 10 Apr 2025 14:28:33 GMT
No después	Fri, 10 Apr 2026 14:28:33 GMT

Información de clave pública

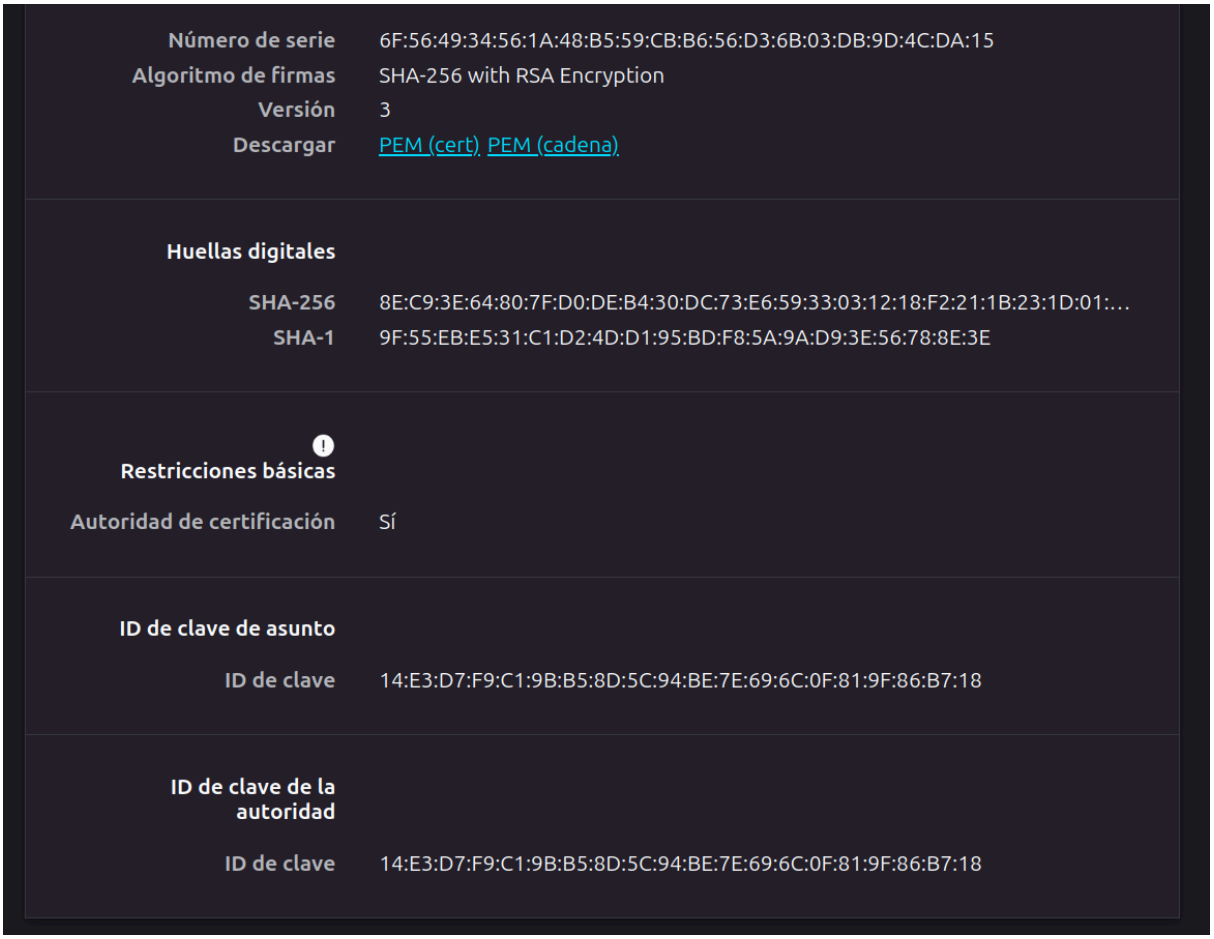
Algoritmo	RSA
Tamaño de la clave	2048
Exponente	65537
Módulo	88:28:E0:42:2C:0B:44:C8:7B:A2:7F:1C:99:BC:21:F1:C6:C8:D0:54:9F:E0:90:6C:…

Misceláneo

Número de serie	6F:56:49:34:56:1A:48:B5:59:CB:B6:56:D3:6B:03:DB:9D:4C:DA:15
Algoritmo de firmas	SHA-256 with RSA Encryption
Versión	3
Descargar	PEM (cert) PEM (cadena)

Huellas digitales

SHA-256	8E:C9:3E:64:80:7F:D0:DE:B4:30:DC:73:E6:59:33:03:12:18:F2:21:1B:23:1D:01:…
SHA-1	9F:55:EB:E5:31:C1:D2:4D:D1:95:BD:F8:5A:9A:D9:3E:56:78:8E:3E



La información y características se corresponden con la del certificado que creamos en la tarea básica 2.

Tareas Avanzadas - A1: Exploraciones Avanzadas de creación de certificados SSL.

Para crear una cadena de confianza es necesario seguir una serie de pasos:

1. Crear una clave privada (caRaiz.key) para con ella crear un certificado raíz (caRaiz.crt). Con esa clave privada, el certificado raíz se firma a sí mismo para posteriormente poder firmar los certificados intermedios. Significa el inicio de la cadena de confianza y no depende de otra entidad para validarse. Al ser el elemento más importante de la cadena toda la seguridad se sustenta sobre él y debe ser secreto.

```
jorge@smart-refrigerator:~/Escritorio/3ro_de_carrera/segundo_cuatri/SWAP/practic
as/p3/P3-jorgelpz-A1$ openssl req -x509 -new -nodes -key caRaiz.key -sha256 -day
s 365 -out caRaiz.crt
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Granada
Locality Name (eg, city) []:Granada
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SWAP
Organizational Unit Name (eg, section) []:Practica 3
Common Name (e.g. server FQDN or YOUR name) []:jorgelpz
Email Address []:jorgelpz@correo.ugr.es
```

```
jorge@smart-refrigerator:~/Escritorio/3ro_de_carrera/segundo_cuatri/SWAP/practic
as/p3/P3-jorgelpz-A1$ ls
caRaiz.crt  caRaiz.key
```

2. Crear una nueva clave privada para la subcadena (subCa.key) y una solicitud de firma (subCa.csr) que contiene principalmente la clave pública de la subcadena junto a otros datos para que la raíz pueda firmar. La raíz firma la solicitud de la subcadena y obtenemos un certificado de la subcadena (subCa.crt).

```
jorge@smart-refrigerator:~/Escritorio/3ro_de_carrera/segundo_cuatri/SWAP/practic
as/p3/P3-jorgelpz-A1$ openssl req -new -key subCa.key -out subCa.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Granada
Locality Name (eg, city) []:Granada
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SWAP
Organizational Unit Name (eg, section) []:Practica 3
Common Name (e.g. server FQDN or YOUR name) []:jorgelpz
Email Address []:jorgelpz@correo.ugr.es

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:SWAP1234
An optional company name []:
```

```
jorge@smart-refrigerator:~/Escritorio/3ro_de_carrera/segundo_cuatri/SWAP/practic
as/p3/P3-jorgelpz-A1$ openssl x509 -req -in subCa.csr -CA caRaiz.crt -CAkey caRa
iz.key -CAcreateserial -out subCa.crt -days 365 -sha256
Certificate request self-signature ok
subject=C = ES, ST = Granada, L = Granada, O = SWAP, OU = Practica 3, CN = jorge
lpz, emailAddress = jorgelpz@correo.ugr.es
```

3. Ahora con esa subcadena certificada por la raíz podemos firmar nuevas solicitudes sin necesidad de exponer la clave raíz. Generamos una nueva key para autenticar a los servidores apache (server.key) y con ella una nueva CSR (server.csr).

```
jorge@smart-refrigerator:~/Escritorio/3ro_de_carrera/segundo_cuatri/SWAP/practic
as/p3/P3-jorgelpz-A1$ openssl genrsa -out server.key 2048
jorge@smart-refrigerator:~/Escritorio/3ro_de_carrera/segundo_cuatri/SWAP/practic
as/p3/P3-jorgelpz-A1$ openssl req -new -key server.key -out server.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Granada
Locality Name (eg, city) []:Granada
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SWAP
Organizational Unit Name (eg, section) []:Practica 3
Common Name (e.g. server FQDN or YOUR name) []:jorgelpz
Email Address []:jorgelpz@correo.ugr.es

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:SWAP1234
An optional company name []:
```

4. Procedemos a firmar la solicitud con el certificado de la subcadena (subCa.crt). De esta forma conseguimos generar el certificado del servidor (server.crt) sin necesidad de exponer la raíz.

```
jorge@smart-refrigerator:~/Escritorio/3ro_de_carrera/segundo_cuatri/SWAP/practic
as/p3/P3-jorgelpz-A1$ openssl x509 -req -in server.csr -CA subCa.crt -CAkey subC
a.key -CAcreateserial -out server.crt -days 365 -sha256
Certificate request self-signature ok
subject=C = ES, ST = Granada, L = Granada, O = SWAP, OU = Practica 3, CN = jorge
lpz, emailAddress = jorgelpz@correo.ugr.es
```

5. Volvemos a repetir este proceso para generar un nuevo certificado para el balanceador de carga. Cuanto menos reutilicemos los certificados más segura será la estructura y más fácil será detectar los fallos. Creamos nginx.key, generamos la solicitud y la firmamos con el certificado de la subcadena.

```
jorge@smart-refrigerator:~/Escritorio/3ro_de_carrera/segundo_cuatri/SWAP/practic
as/p3/P3-jorgelpz-apache$ openssl genrsa -out nginx.key 2048
jorge@smart-refrigerator:~/Escritorio/3ro_de_carrera/segundo_cuatri/SWAP/practic
as/p3/P3-jorgelpz-apache$ openssl req -new -key nginx.key -out nginx.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Granada
Locality Name (eg, city) []:Granada
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SWAP
Organizational Unit Name (eg, section) []:Practica 3
Common Name (e.g. server FQDN or YOUR name) []:jorgelpz
Email Address []:jorgelpz@correo.ugr.es

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:SWAP1234
An optional company name []:
```

```
jorge@smart-refrigerator:~/Escritorio/3ro_de_carrera/segundo_cuatri/SWAP/practic
as/p3/P3-jorgelpz-A1$ openssl x509 -req -in nginx.csr -CA subCa.crt -CAkey subCa
.key -CAcreateserial -out nginx.crt -days 365 -sha256
Certificate request self-signature ok
subject=C = ES, ST = Granada, L = Granada, O = SWAP, OU = Practica 3, CN = jorge
lpz, emailAddress = jorgelpz@correo.ugr.es
```

6. Una vez creados todos los certificados que vamos a usar procedemos a concatenarlos de la siguiente forma para formar la propia cadena de confianza.

```
1$ cat nginx.crt subCa.crt > chain_nginx.crt
1$ cat server.crt subCa.crt > chain_apache.crt
```

7. Modificamos los archivos de configuración de apache y nginx especificando cuales son ahora sus nuevos certificados (la cadena).

```
listen 443 ssl;
ssl_certificate /etc/nginx/ssl/chain_nginx.crt;
ssl_certificate_key /etc/nginx/ssl/nginx.key;
                                                    tored in the same
#   SSLCertificateFile directive is needed.
SSLCertificateFile /etc/apache2/ssl/chain_apache.crt
SSLCertificateKeyFile /etc/apache2/ssl/server.key
```

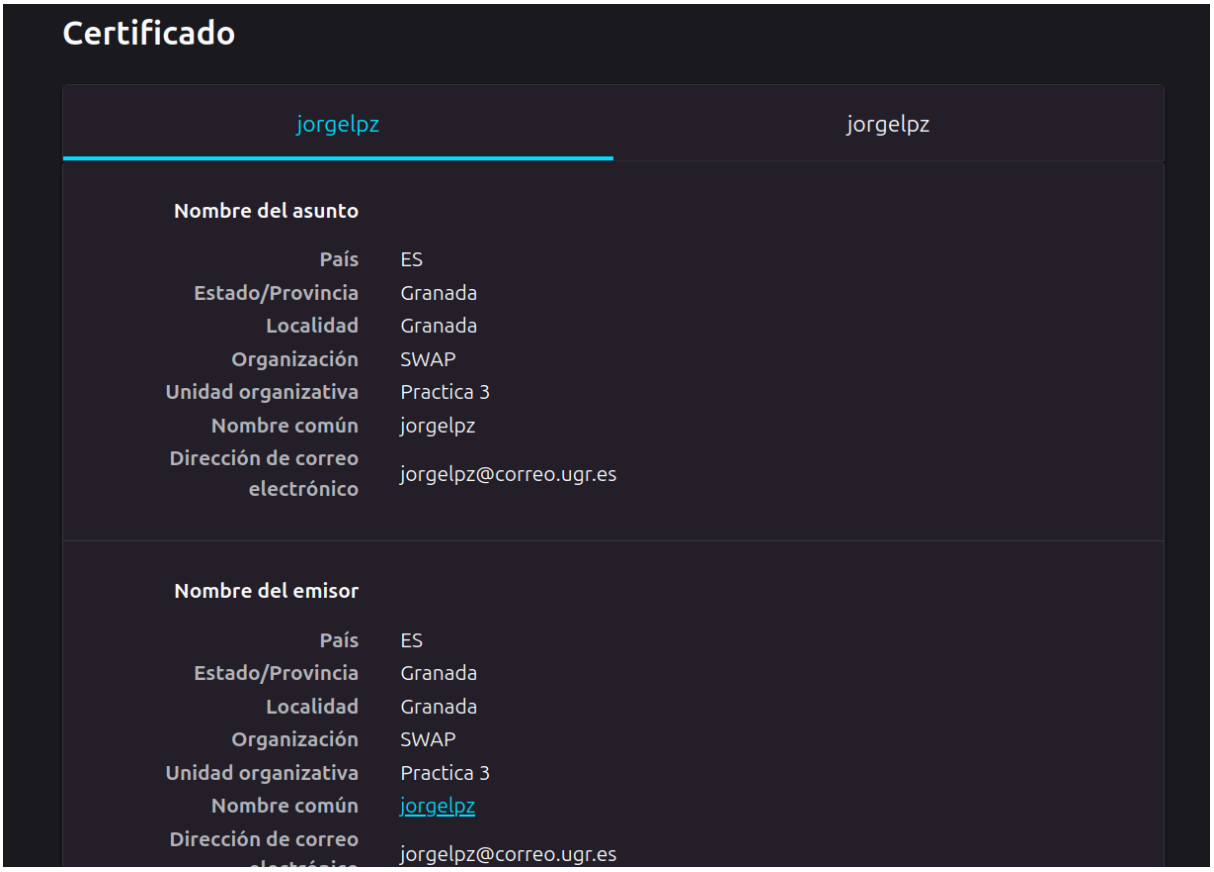
8. Modificamos el docker-compose.yml para indicarle los nuevos directorios y permisos que debemos asignar.

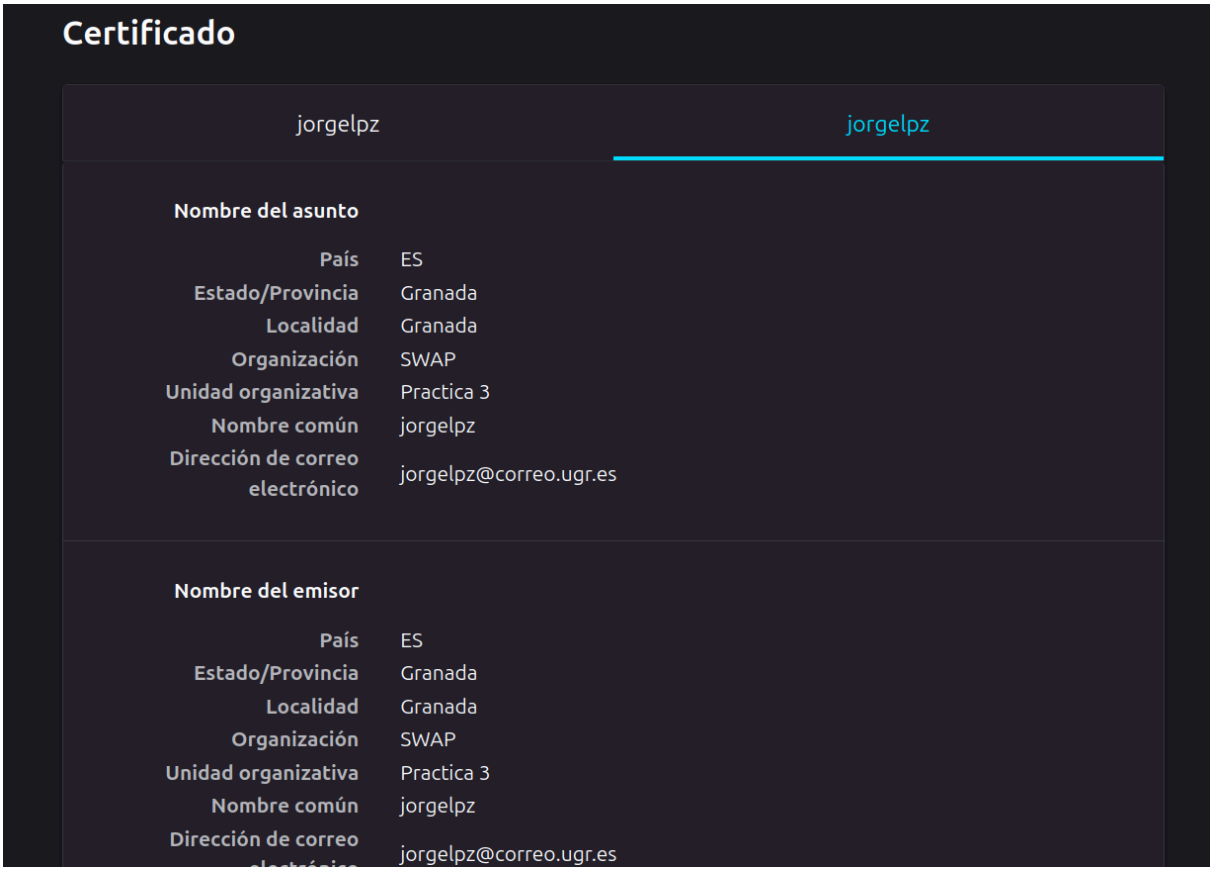
```
web1:
    image: jorgelpz-apache-image:p3
    container_name: web1-A1
    volumes:
    - ./P3-jorgelpz-apache/web_jorgelpz:/var/www/html
    - ./P3-jorgelpz-A1:/etc/apache2/ssl
    - ./P3-jorgelpz-apache/jorgelpz-apache-sslA1.conf:/etc/apache2/sites-available/jorgelpz-apache-ssl.conf
    command:
        /bin/sh -c "chmod 600 /etc/apache2/ssl/server.crt && apachectl -D FOREGROUND"
    networks:
        red_web:
            ipv4_address: 192.168.10.2
        red_servicios:
            ipv4_address: 192.168.20.2
```

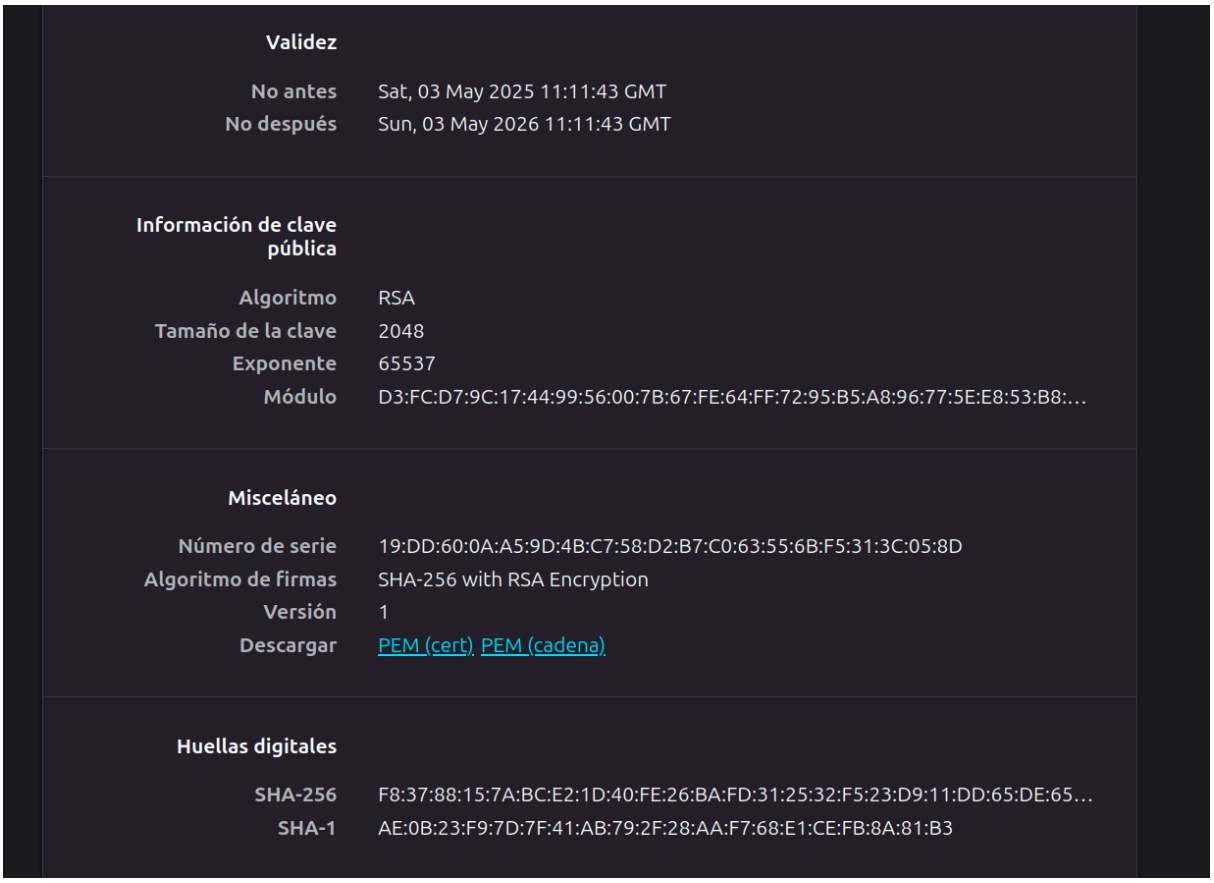
```
balanceador-nginx-ssl:
    image: jorgelpz-nginx-image:p3
    container_name: balanceador-nginx-ssl-A1
    ports:
    - "9000:443"
    volumes:
    - ./P3-jorgelpz-nginx/jorgelpz-nginx-sslA1.conf:/etc/nginx/nginx.conf
    - ./P3-jorgelpz-A1:/etc/nginx/ssl
    networks:
        red_web:
            ipv4_address: 192.168.10.50
    depends_on:
```

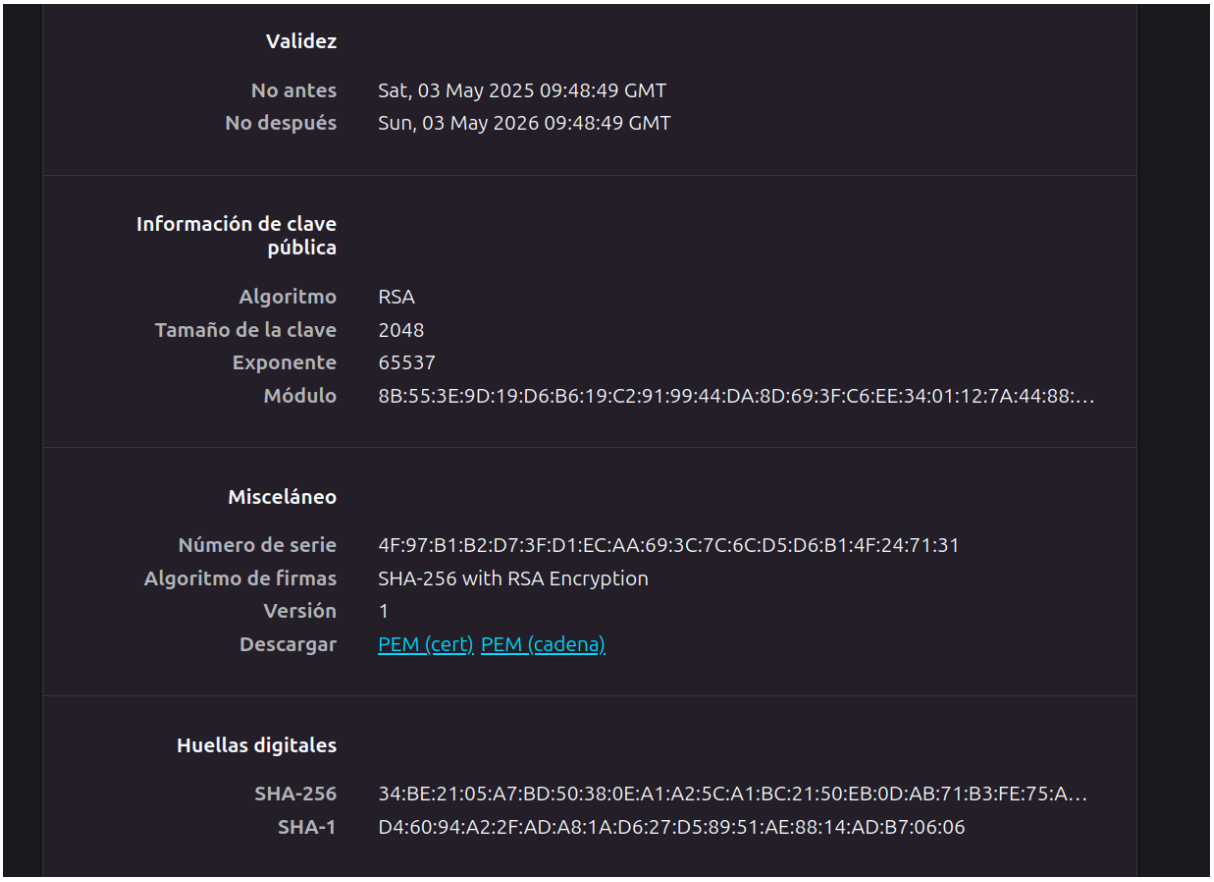
Podemos comprobar que el balanceador funciona correctamente y si vamos a comprobar el certificado nos saldrá, por una parte server.crt y por otra subCa.crt. Esto se debe a que hemos concatenado ambos para los servicios apache y de forma similar ocurriría en nginx.


https://localhost:9000
Práctica SWAP - Jorge López Molina
Dirección IP del servidor Apache: 192.168.10.7
Software del servidor: Apache/2.4.58 (Ubuntu)
El nombre del contenedor que lo está sirviendo es bd47b8d77601
Número del contenedor: Web 6
https://localhost:9000
Práctica SWAP - Jorge López Molina
Dirección IP del servidor Apache: 192.168.10.8
Software del servidor: Apache/2.4.58 (Ubuntu)
El nombre del contenedor que lo está sirviendo es 23c8740cbfce
Número del contenedor: Web 7
https://localhost:9000
Práctica SWAP - Jorge López Molina
Dirección IP del servidor Apache: 192.168.10.9
Software del servidor: Apache/2.4.58 (Ubuntu)
El nombre del contenedor que lo está sirviendo es 5edc3b8e8745
Número del contenedor: Web 8


Certificado
jorgelpz
jorgelpz
Nombre del asunto
País ES
Estado/Provincia Granada
Localidad Granada
Organización SWAP
Unidad organizativa Practica 3
Nombre común jorgelpz
Dirección de correo electrónico jorgelpz@correo.ugr.es
Nombre del emisor
País ES
Estado/Provincia Granada
Localidad Granada
Organización SWAP
Unidad organizativa Practica 3
Nombre común jorgelpz
Dirección de correo jorgelpz@correo.ugr.es


Certificado
jorgelpz
jorgelpz
Nombre del asunto
País ES
Estado/Provincia Granada
Localidad Granada
Organización SWAP
Unidad organizativa Practica 3
Nombre común jorgelpz
Dirección de correo electrónico jorgelpz@correo.ugr.es
Nombre del emisor
País ES
Estado/Provincia Granada
Localidad Granada
Organización SWAP
Unidad organizativa Practica 3
Nombre común jorgelpz
Dirección de correo jorgelpz@correo.ugr.es


Validez
No antes Sat, 03 May 2025 11:11:43 GMT
No después Sun, 03 May 2026 11:11:43 GMT
Información de clave pública
Algoritmo RSA
Tamaño de la clave 2048
Exponente 65537
Módulo D3:FC:D7:9C:17:44:99:56:00:7B:67:FE:64:FF:72:95:B5:A8:96:77:5E:E8:53:B8:…
Misceláneo
Número de serie 19:DD:60:0A:A5:9D:4B:C7:58:D2:B7:C0:63:55:6B:F5:31:3C:05:8D
Algoritmo de firmas SHA-256 with RSA Encryption
Versión 1
Descargar PEM (cert) PEM (cadena)
Huellas digitales
SHA-256 F8:37:88:15:7A:BC:E2:1D:40:FE:26:BA:FD:31:25:32:F5:23:D9:11:DD:65:DE:65…
SHA-1 AE:0B:23:F9:7D:7F:41:AB:79:2F:28:AA:F7:68:E1:CE:FB:8A:81:B3



Validez	
No antes	Sat, 03 May 2025 09:48:49 GMT
No después	Sun, 03 May 2026 09:48:49 GMT
Información de clave pública	
Algoritmo	RSA
Tamaño de la clave	2048
Exponente	65537
Módulo	8B:55:3E:9D:19:D6:B6:19:C2:91:99:44:DA:8D:69:3F:C6:EE:34:01:12:7A:44:88:…
Misceláneo	
Número de serie	4F:97:B1:B2:D7:3F:D1:EC:AA:69:3C:7C:6C:D5:D6:B1:4F:24:71:31
Algoritmo de firmas	SHA-256 with RSA Encryption
Versión	1
Descargar	PEM (cert) PEM (cadena)
Huellas digitales	
SHA-256	34:BE:21:05:A7:BD:50:38:0E:A1:A2:5C:A1:BC:21:50:EB:0D:AB:71:B3:FE:75:A…
SHA-1	D4:60:94:A2:2F:AD:A8:1A:D6:27:D5:89:51:AE:88:14:AD:B7:06:06

La salida del comando $openssl s_client -connect localhost:900 a modo de comprobación de lo configurado hasta ahora:

```
Certificate chain
 0 s:C = ES, ST = Granada, L = Granada, O = SWAP, OU = Practica 3, CN = jorgelpz, emailAddress = jorgelpz@correo.ugr.es
   i:C = ES, ST = Granada, L = Granada, O = SWAP, OU = Practica 3, CN = jorgelpz, emailAddress = jorgelpz@correo.ugr.es
   a:PKEY: rsaEncryption, 2048 (bit); sigalg: RSA-SHA256
   v:NotBefore: May  3 11:11:43 2025 GMT; NotAfter: May  3 11:11:43 2026 GMT
 1 s:C = ES, ST = Granada, L = Granada, O = SWAP, OU = Practica 3, CN = jorgelpz, emailAddress = jorgelpz@correo.ugr.es
   i:C = ES, ST = Granada, L = Granada, O = SWAP, OU = Practica 3, CN = jorgelpz, emailAddress = jorgelpz@correo.ugr.es
   a:PKEY: rsaEncryption, 2048 (bit); sigalg: RSA-SHA256
   v:NotBefore: May  3 09:48:49 2025 GMT; NotAfter: May  3 09:48:49 2026 GMT
```

Tareas Avanzadas - A2: Optimización de la configuración SSL en los servidores web.

De cara a optimizar la seguridad y el rendimiento de la configuración SSL en apache he deshabilitado protocolos y cifrados inseguros para permitir la conexión sólo a través de los medios más fiables. La IA me ha ayudado a saber cual es la configuración más fiable para mi servidor. En este caso he usado la función de búsqueda por internet de Perplexity.ai. De esta formas consigo que la respuesta sea lo más actualizada posible. Enlace a la conversación de 3 mensajes en el apartado final.

Finalmente he optado por aceptar solamente el protocolo TLS 1.2 por ser el mejor si contamos lo seguro que es y lo extendido que está y el protocolo TLS 1.3 por ser el que permite los algoritmos de cifrado más avanzados por ahora. Según la búsqueda de Perplexity.ai, SSLv2, SSLv3, TLS 1.0 y TLS 1.1 están obsoletos y son vulnerables a ataques como POODLE (un tipo de man-in-the-middle) o renegociación insegura.

Por otra parte, para los algoritmos de cifrado solamente permito ECDHE+AESGCM y ECDHE+CHACHA20. Por ahora permanecen sin vulnerabilidades conocidas. Desactivo las conexiones con RC4, 3DES o SHA-1 porque son sensibles a ataques man-in-the-middle y colisiones hash.

Mi configuración es:

```
SSLCertificateFile /etc/apache2/ssl/chain_apache.crt
SSLCertificateKeyFile /etc/apache2/ssl/server.key

SSLProtocol -all +TLSv1.2 +TLSv1.3
SSLCipherSuite ECDHE+AESGCM:ECDHE+CHACHA20
SSLHonorCipherOrder on
```

Comprobaciones:

1. Lo rechaza porque es RC4:

```
jorge@smart-refrigerator:~/Escritorio/3ro_de_carrera/segundo_cuatri/SWAP/practicas/p3/P3-jorgelpz-A2$ openssl s_client -connect localhost:9000 -cipher RC4
Call to SSL_CONF_cmd(-cipher, RC4) failed
40F7D3FC3B7B0000:error:0A0000B9:SSL routines:SSL_CTX_set_cipher_list:no cipher match:../ssl/ssl_lib.c:2779:
```

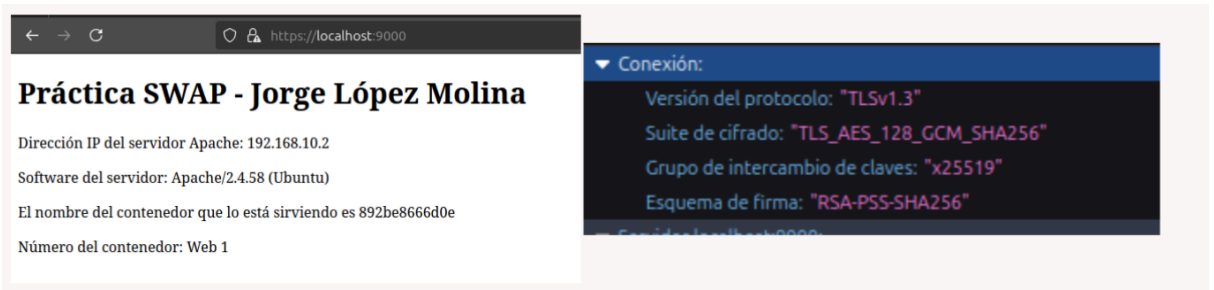
2. Lo rechaza por ser TLS 1:

```
jorge@smart-refrigerator:~/Escritorio/3ro_de_carrera/segundo_cuatri/SWAP/practic
as/p3/P3-jorgelpz-A2$ openssl s_client -connect localhost:9000 -tls1
CONNECTED(00000003)
40A748A754770000:error:0A0000BF:SSL routines:tls_setup_handshake:no protocols av
ailable:../ssl/statem/statem_lib.c:104:
---
no peer certificate available
---
No client certificate CA names sent
---
SSL handshake has read 0 bytes and written 7 bytes
Verification: OK
---
New, (NONE), Cipher is (NONE)
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 0 (ok)
---
```

3. Lo admite por ser TLS 1.2:

```
jorge@smart-refrigerator:~/Escritorio/3ro_de_carrera/segundo_cuatri/SWAP/practic
as/p3/P3-jorgelpz-A2$ openssl s_client -connect localhost:9000 -tls1_2
CONNECTED(00000003)
Can't use SSL_get_servername
depth=0 C = ES, ST = Granada, L = Granada, O = SWAP, OU = Practica 3, CN = jorge
lpz, emailAddress = jorgelpz@correo.ugr.es
verify error:num=18:self-signed certificate
verify return:1
depth=0 C = ES, ST = Granada, L = Granada, O = SWAP, OU = Practica 3, CN = jorge
lpz, emailAddress = jorgelpz@correo.ugr.es
verify return:1
---
Certificate chain
 0 s:C = ES, ST = Granada, L = Granada, O = SWAP, OU = Practica 3, CN = jorgelpz
, emailAddress = jorgelpz@correo.ugr.es
   i:C = ES, ST = Granada, L = Granada, O = SWAP, OU = Practica 3, CN = jorgelpz
, emailAddress = jorgelpz@correo.ugr.es
   a:PKEY: rsaEncryption, 2048 (bit); sigalg: RSA-SHA256
   v:NotBefore: May  3 11:11:43 2025 GMT; NotAfter: May  3 11:11:43 2026 GMT
 1 s:C = ES, ST = Granada, L = Granada, O = SWAP, OU = Practica 3, CN = jorgelpz
, emailAddress = jorgelpz@correo.ugr.es
```

4. Si nos conectamos a través del navegador podemos observar como funciona correctamente al ser TLS 1.3 y usar un cifrado permitido:



Tareas Avanzadas - A3: Configuración de Caché de Sesiones SSL y Tickets de Sesión en el balanceador.

La caché de sesiones y los tickets de sesión nacen como solución a lo costoso que es el handshake. Cada vez que se produce un intercambio de peticiones para realizar el handshake en una conexión se pierde tiempo y recursos. Si prevemos que se va a tener que recargar muchas veces es interesante guardar en una caché estos inicios de sesión durante un lapso de tiempo de cara a evitar repetir intercambios innecesarios.

Para configurarla debemos añadir las siguientes directivas:

```
ssl_session_cache shared:SSL:5m;
ssl_session_timeout 30m;
ssl_session_tickets on;
```

- ssl_session_cache shared:SSL:5m indica que guardamos una caché compartida de 5MB en la que se almacenan las conexiones recientes.
- ssl_session_timeout 30m condiciona cuánto tiempo pasarán dichas conexiones dentro de la caché.
- ssl_session_tickets on habilita el uso de tickets de sesión.

Comprobamos que funciona:

- Ejecutando $ openssl s_client -connect localhost:9000 -tls1_3 -sess_out session_ticket.pem observamos que se genera el ticket de sesión.

```
jorge@smart-refrigerator:~/Escritorio/3ro_de_carrera/segundo_cuatri/SWAP/practicas/p3/borrar$ cat session_ticket.pem
-----BEGIN SSL SESSION PARAMETERS-----
MIIFHgIBAQICAwQEAhMCBCA+PUIK+TuWllZIXyjGOxI7k8a3FKChmbWa/HCfXWt+
7AQwlZxysyG33shtrYjvUP9VNoPnxTkJ+0zzcozQvaU67bLNkOsrz5mRl2vh1lNa
lTyfoQYCBGgWWPqiBAICHCCjggOrMIIDpzCCAo8CFBndYAqlnUvHWNK3wGNVa/Ux
PAWNMA0GCSqGSIb3DQEBCwUAMIGPMQswCQYDVQQGEwJFUzEQMA4GA1UECAwHR3Jh
bmFkYTEQMA4GA1UEBwwHR3JhbmFkYTENMAsGA1UECgwEU1dBUDETMBEGA1UECwwK
UHJhY3RpY2EgMzERMA8GA1UEAwwIam9yZ2VscHoxJTAjBgkqhkiG9w0BCQEWFmpv
cmdlbHB6QGNvcnJlby51Z3IuZXMwHhcNMjUwNTAzMTExMTQzWhcNMjYwNTAzMTEx
MTQzWjCBjzELMAkGA1UEBhMCRVMxEDAOBgNVBAgMB0dyYW5hZGExEDAOBgNVBAcM
B0dyYW5hZGExDTALBgNVBAoMBFNXQVAxEzARBgNVBAsMClByYWN0aWNhIDMxETAP
BgNVBAMMCGpvcmdlbHB6MSUwIwYJKoZIhvcNAQkBFhZqb3JnZWxwekBjb3JyZW8u
dWdyLmVzMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA0/zXnBdEmVYA
e2f+ZP9ylbWolnde6FO4j2T/ozpc4XJ2JpHPUYTWXvOITViTrH9Y7vkjl7Kjl0Gd
HRD55/6QFlhb+uiAwYJG1RDM1xQD+K4wgqviZwgByUb5gh0R4WQJEV+cFcRRZUFz
0OdAwOZB0fF5DtYRsmlcsUHRFX1z5EflpoyVcgAFyH9o4NiRVSwKnjAzkPtCO/5h
IfVSicYJkMx2xboByfOjA4awyM/gJDhsdeU2HsGeghsEjUld+jZ0jirbJpxRrngA
ZMflz+dE0tnl8D80SITwPXwoLr3vW6Y9artrsRlrzfEZaDZyh9yIbrUbUzHzSh4L
tqktd1P3/wIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQCIThi9xRu1G7yLuvHujVwV
ERqkHG/YJNRyTLW25WLVjuvm4729DV0TvANJKg7HOFIkF1aHU03mgB13nJIzDPO9
IzmvIaxN2uxQgjYu7m3HOBYOcmCrCE2U+eeE5hu331tR1/eYbK8m9+B/f1EDYGub
+wgVnOV1OwPadz4XxMtiZbtznXNryYvUA1QQJDojnTl7PoS4nt+XF1ZffaRJ8IA/
Aof8fCNF4+PULKKKTYU/dhognIgSHvmDOGTFsD4IC8/UPgIEhJG2z1jA6DaFn+hZ
eHvo2iCp1UOSTZzI+EsU0mKS9vOzpV0bYsFiBG5fjS7FP5TO1SW4A3xMy7w25NEk
pAIEAKUDAgESqQQCAgJYqoHjBIHgzQEYFUtLwRUVP8iPfPwxP3c19h8mYnt4pLRH
7fiSElb/baP8GGJBYsRQpacLY4IzhaWlFVnz7odSz8veEkzfGjxMEgpe0mckzNi1
Q8IVKLKKu4Cj2Sp9Z5mWF7wLcCoaAFhCNdknb1PtvN1erSE8ToN9f598PUZoxboF
3F4w6kGbXwx8t1h3aDlvUYnd+4f+syHaI/+WuKMYut6ANSgX3bpi/YT/a2oG2u+O
60xMa6FKtDDHlzl3sQzGSWesu8K3nqenkavOK5TESPv+n0zdeMGR3seSuoubi4xZ
0nWlSw6uBgIEGJytbbMDAgEd
-----END SSL SESSION PARAMETERS-----
```

- Si vemos la salida del comando anterior podemos observar la forma que tienen los tickets de sesión:

```
TLS session ticket:
0000 - 7d 27 dd fb b2 05 6c 02-6b 6a 77 1d a0 18 c1 b7   }'....l.kjw.....
0010 - 5b 55 e9 8c 4a bf 8f d6-9c 4d 04 22 72 fb 27 d9   [U..J....M."r.'.
0020 - b4 d5 dd 80 0e 03 e7 ec-37 16 e0 9d 94 49 3c 8b   ........7....I<.
0030 - 95 73 cb de ef c0 47 f1-9a 16 64 3e 64 fb 3c d2   .s....G...d>d.<.
0040 - 2b ca a5 28 78 c9 3a ee-a1 4d 2a 33 54 9e ff ab   +..(x.:..M*3T...
0050 - 06 b9 8e ff 66 ea 3d a0-57 54 92 d3 ec 32 cb 7d   ....f.=.WT...2.}
0060 - a7 a4 2a 9a aa a2 ee 14-66 e1 90 64 93 76 ae b8   ..*.....f..d.v..
0070 - 68 da 0c 39 29 98 a9 59-bf 0a 43 60 7e f6 ea 3f   h..9)..Y..C`~..?
0080 - 6f 71 7e fd ab 98 eb 73-9f c3 07 2a 15 04 e7 d4   oq~....s...*....
0090 - 9d 2c b1 b6 d5 9d 89 bd-fa 32 c3 36 b5 25 0a cc   .,.......2.6.%..
00a0 - b4 31 2c 13 81 8d 70 37-b3 a2 01 15 61 fa 8b fc   .1,...p7....a...
00b0 - e0 75 18 00 66 a5 2d e8-04 e5 c1 c5 1a aa 2c b6   .u..f.-.......,.
00c0 - 35 2f a3 1c 9d f5 07 3a-5a f1 bb a4 9b c9 ee 79   5/.....:Z......y
00d0 - a5 a0 5e 34 00 cd 71 d7-e5 65 e4 7a 4f 30 5a 1b   ..^4..q..e.zO0Z.
```

Tareas Avanzadas - A4: Optimización de conexiones HTTPS y cifrado en el balanceador.

En este apartado se indica que se debe personalizar la configuración del balanceador para, primeramente, equilibrar la seguridad y el rendimiento. Para lograrlo, al igual que en apache, acepto solo peticiones de los protocolos TLS 1.2 y TLS 1.3. Esto representa una ventaja ya que las versiones anteriores de TLS se consideran obsoletas. TLS 1.2 es equilibrado en cuanto a compatibilidad y seguridad. TLS 1.3 ofrece mejor seguridad y mayor eficiencia en el proceso del handshake.

Por otra parte, he incluido en mi configuración cadenas de cifrados modernos. Los cifrados modernos son más rápidos y seguros que los clásicos. ECDHE (Elliptic Curve Diffie-Hellman Ephemeral) es especialmente común en estas cadenas de cifrados ya que logra proteger el tráfico incluso si la confidencialidad de la clave del servidor se ve vulnerada. Esto es gracias a que usa una clave única para cada intercambio de forma que si captura una no podrá descifrar más tráfico.

Si llega una conexión que no sea TLS 1.2 o 1.3 y que no esté cifrada con ninguno de estos algoritmos será revocada inmediatamente.

Por último añadiendo http2 a la directiva listen podemos también recibir peticiones HTTP/2. Este protocolo supera en eficiencia a las conexiones HTTPS ya que permite multiplexación y compresión de las cabeceras.

```
server {

    listen 443 ssl http2;
    ssl_certificate /etc/nginx/ssl/chain_nginx.crt;
    ssl_certificate_key /etc/nginx/ssl/nginx.key;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers 'ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:EC
    ssl_prefer_server_ciphers on;

    ssl_session_cache shared:SSL:5m;
    ssl_session_timeout 30m;
    ssl_session_tickets on;

    location / {
```

Comprobamos que funciona:

- Si tratamos de conectarnos con TLS 1.3 se efectúa correctamente:

```
jorge@smart-refrigerator:~/Escritorio/3ro_de_carrera/segundo_cuatri/SWAP/practicas/p3/borrar$ openssl s_client -connect localhost:9000 -tls1_3
CONNECTED(00000003)
Can't use SSL_get_servername
depth=0 C = ES, ST = Granada, L = Granada, O = SWAP, OU = Practica 3, CN = jorgelpz, emailAddress = jorgelpz@correo.ugr.es
verify error:num=18:self-signed certificate
verify return:1
depth=0 C = ES, ST = Granada, L = Granada, O = SWAP, OU = Practica 3, CN = jorgelpz, emailAddress = jorgelpz@correo.ugr.es
verify return:1
---
Certificate chain
 0 s:C = ES, ST = Granada, L = Granada, O = SWAP, OU = Practica 3, CN = jorgelpz, emailAddress = jorgelpz@correo.ugr.es
   i:C = ES, ST = Granada, L = Granada, O = SWAP, OU = Practica 3, CN = jorgelpz, emailAddress = jorgelpz@correo.ugr.es
   a:PKEY: rsaEncryption, 2048 (bit); sigalg: RSA-SHA256
   v:NotBefore: May  3 11:11:43 2025 GMT; NotAfter: May  3 11:11:43 2026 GMT
 1 s:C = ES, ST = Granada, L = Granada, O = SWAP, OU = Practica 3, CN = jorgelpz, emailAddress = jorgelpz@correo.ugr.es
   i:C = ES, ST = Granada, L = Granada, O = SWAP, OU = Practica 3, CN = jorgelpz, emailAddress = jorgelpz@correo.ugr.es
   a:PKEY: rsaEncryption, 2048 (bit); sigalg: RSA-SHA256
   v:NotBefore: May  3 09:48:49 2025 GMT; NotAfter: May  3 09:48:49 2026 GMT
---
```

- Si tratamos de conectarnos con TLS 1.1 (es un ejemplo de protocolo no aceptado) la conexión no se establecerá:

```
-tls1_1  -tls1_2  -tls1_3
jorge@smart-refrigerator:~/Escritorio/3ro_de_carrera/segundo_cuatri/SWAP/practicas/p3/borrar$ openssl s_client -connect localhost:9000 -tls1_1
CONNECTED(00000003)
40F7648F297A0000:error:0A0000BF:SSL routines:tls_setup_handshake:no protocols available:../ssl/statem/statem_lib.c:104:
---
no peer certificate available
---
No client certificate CA names sent
---
SSL handshake has read 0 bytes and written 7 bytes
Verification: OK
---
New, (NONE), Cipher is (NONE)
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 0 (ok)
---
jorge@smart-refrigerator:~/Escritorio/3ro_de_carrera/segundo_cuatri/SWAP/practicas/p3/borrar$
```

- Si nos conectamos usando HTTP/2 obtenemos un código 200 indicando que todo está correcto:

```
jorge@smart-refrigerator:~/Escritorio/3ro_de_carrera/segundo_cuatri/SWAP/practicas/p3/borrar$ curl -I --http2 -k https://localhost:9000
HTTP/2 200
server: nginx/1.27.5
date: Sat, 03 May 2025 18:54:48 GMT
content-type: text/html; charset=UTF-8
```

- El balanceador sigue trabajando con normalidad:

Práctica SWAP - Jorge López Molina

Dirección IP del servidor Apache: 192.168.10.4

Software del servidor: Apache/2.4.58 (Ubuntu)

El nombre del contenedor que lo está sirviendo es cddefa63ce97

Número del contenedor: Web 3

Análisis propuesta IA.

Enlace Principal:
https://github.com/copilot/share/c2655112-0b00-8c76-b001-5e0ac4fd49ad

Enlace Cuestión Teórica A2:
https://www.perplexity.ai/search/en-base-a-lo-mas-reciente-posi-YimtzfYyQbWw8EDhJ2uWAw

Al igual que en las prácticas anteriores, las herramientas de IA generativa han jugado un papel crucial en el desarrollo de las tareas avanzadas.

Si bien las tareas básicas se podían sacar sin acudir a recursos externos solo con lo aprendido en las prácticas anteriores, las tareas avanzadas han sido mucho más complejas.

Sin duda el punto en el que más me han ayudado ha sido en la tarea avanzada A1. No podía llegar a imaginar que existieran cadenas de confianza. En mi cabeza solamente con el certificado configurado en las tareas básicas ya era más que suficiente. Ha sido tarea de la IA el ayudar, primeramente, a entender el concepto de cadena de confianza y, una vez completada esa parte, a conocer los pasos de crear una. Gracias a sus explicaciones ahora tengo una imagen mental completa del concepto y de la importancia que tiene.

Destacar por otra parte lo potente que me ha parecido perplexity pro con su herramienta de búsqueda en internet. Me ha recordado a los trabajos en la ESO donde hacías una investigación de la que luego sacabas un trabajo con una webgrafía. Ahora puedes resumir todo en una consulta a la IA generativa obteniendo incluso las webs por las que ha navegado antes de mostrarte la respuesta. En las primeras versiones de Chat GPT que usé, no tenía nueva información en sus repertorios desde hacía más de un año atrás. Me ha parecido una forma brillante de superar el problema de la IA desactualizada si las webs están al día de los cambios.

Para finalizar, me quedo con el progreso en el uso de estas herramientas. En primero de carrera no usé IA generativa. En segundo comencé a usarla pero la entendía como algo despectivo, si algo estaba “hecho con IA” simplemente tenía menos mérito. Hoy en día me centro en aprender el uso de estas tecnologías orientadas a potenciar la productividad de todos y de todo.