

**Duración:** 2 sesiones

La práctica se realizará de manera individual. Tiene un peso del **20%** del total de prácticas.

**Requisitos Previos:**

- Haber completado satisfactoriamente las Prácticas 1 y 2 o tener experiencia equivalente en configuración de balanceadores de carga con Docker.
- Conocimientos básicos sobre seguridad web, protocolos HTTPS y certificados SSL.

## Introducción

La seguridad en las aplicaciones web es un pilar fundamental en el desarrollo y mantenimiento de sitios y servicios en línea. Es crucial garantizar que la comunicación entre el usuario y el servidor se realice de manera segura, protegiendo la integridad y la confidencialidad de los datos intercambiados. En esta práctica, nos enfocaremos en la implementación de la seguridad mediante el uso de certificados SSL, estableciendo un entorno HTTPS en nuestra granja web.

**Objetivos de la Práctica:**

Esta práctica tiene como meta reforzar la seguridad de nuestra infraestructura web utilizando contenedores Docker y aplicando conceptos clave de cifrado y autenticación.

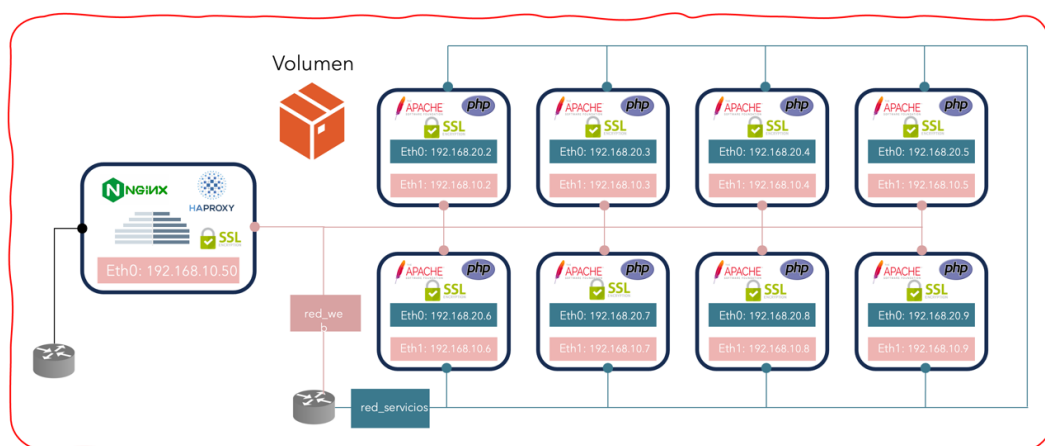
1. Aprender a crear un certificado SSL autofirmado utilizando OpenSSL.
2. Instalar y configurar el módulo SSL en los servidores web para peticiones HTTPS.
3. Configurar un entorno seguro en la granja web, asegurando la comunicación mediante el protocolo HTTPS en un balanceador de carga.

**Descripción de la Práctica:**

Se comenzará por generar un certificado SSL autofirmado, que nos permitirá implementar una conexión segura aunque sin la validación de una entidad certificadora. Posteriormente, procederemos a configurar nuestros servidores web para utilizar dicho certificado y atender de forma segura las peticiones HTTPS así como el balanceador.

La práctica se estructurará en torno a la creación, configuración y despliegue de un entorno que simula una configuración real y segura de una granja web, utilizando las técnicas y herramientas adquiridas en las prácticas anteriores, con el añadido de la capa de seguridad del protocolo HTTPS.

## Esquema general de la práctica:



## Desarrollo:

Se pretende desarrollar un entorno seguro utilizando SSL para la granja web. Implementaremos SSL en los servidores web Apache y configuraremos Nginx como balanceador de carga para gestionar peticiones HTTPS.

### Parte 0: Creación del espacio de trabajo SSL

En esta parte se establecerá el espacio de trabajo a través de directorios específicos donde se crearán los archivos de configuración, los certificados SSL así como la web del escenario.

- Crea un directorio en tu máquina local llamado **P3-tuusuariougr-certificados** para crear los certificados SSL, otro directorio **P3-tuusuariougr-apache** para trabajar con los archivos de configuración de apache y otro directorio llamado **P3-tuusuariougr-nginx** para trabajar con los archivos de configuración de nginx.
- Copia el directorio que creaste en la práctica 1 llamado **web\_tuusuariougr** para que los servidores web sirvan el `index.php` que creaste en la práctica 1.

### Parte 1: Creación de certificados SSL

En esta parte crearemos los archivos necesarios para el certificado SSL: clave privada y certificado autofirmado con OpenSSL en el directorio **P3-tuusuariougr-certificados**. El certificado SSL deberá tener las siguientes especificaciones:

- **Generar un certificado:** Generar un certificado autofirmado en lugar de una solicitud de firma de certificado.
- **Validez de 1 año:** El certificado debe ser válido por 365 días desde el momento de su creación.
- **Encriptación RSA de 2048 bits:** La clave privada asociada con el certificado debe ser una clave RSA con una longitud de 2048 bits para asegurar una encriptación fuerte.
- **Sin necesidad de passphrase:** La clave privada no debe requerir una passphrase para facilitar su uso en automatizaciones y evitar la intervención manual durante el reinicio de los servicios.
- **Autofirmado:** El certificado debe ser autofirmado, lo que significa que la misma entidad que lo crea, lo firma, lo que es adecuado para entornos de prueba.
- **Clave privada:** El archivo con la clave privada debe llamarse certificado\_**tuusuariougr**.key
- **Certificado autofirmado:** El archivo con el certificado autofirmado debe llamarse certificado\_**tuusuariougr**.crt
- **Datos para el certificado de dominio:**
  - Nombre de país: *ES*
  - Provincia: *Granada*
  - Localidad: *Granada*
  - Organización: *SWAP*
  - Organización sección: *Práctica 3*
  - Nombre: "Nombre y apellidos"
  - Email: "email\_ugr"

### Parte 2: Configuración de Servidores Web Apache con SSL

En esta parte configuraremos los servidores web apache finales para atender peticiones HTTPS usando certificados SSL autofirmados. Trabajaremos en el directorio **P3-tuusuariougr-apache**.



## 2.1 - Archivo de configuración SSL - **tuusuariougr-apache-ssl.conf**

Este archivo debe contener la configuración de Apache para habilitar SSL y definir la configuración de los hosts virtuales. Incluirá:

1. La especificación del puerto 443 para escuchar las peticiones HTTPS.
2. La ruta al certificado SSL y a la clave privada.
3. Las directivas para configurar el host virtual para atender peticiones HTTPS, incluyendo la opción `SSLEngine on`.

Ejemplo básico de archivo configuración:

```
<VirtualHost *:443>
    DocumentRoot /var/www/html
    SSLEngine on
    SSLCertificateFile
    /etc/apache2/ssl/certificado_tuusuariougr.crt
    SSLCertificateKeyFile
    /etc/apache2/ssl/certificado_tuusuariougr.key
</VirtualHost>
```

## 2.2 - Dockerfile para Apache con SSL - **DockerFileApacheP3**

La imagen de Docker adecuada, el archivo **DockerFileApacheP3** debe crearse en el directorio **P3-tuusuariougr-apache** y llamarse **tuusuariougr-nginx-image:p3**. Podría contener instrucciones para:

1. Partir de una imagen para Apache creada en la práctica 1.
2. Instalar los módulos necesarios de Apache para soportar SSL, como `mod_ssl` que suele venir preinstalados en imágenes oficiales.
3. Activa SSL y crea un directorio para copiar los archivos de certificado y clave privada (previamente generados) al directorio adecuado dentro del contenedor, comúnmente `/usr/local/apache2/conf/` o `/etc/apache2/ssl/`.
4. Copia el archivo **tuusuariougr-apache-ssl.conf** (archivo que creaste en el apartado anterior) al sitio por defecto (`/etc/apache2/sites-available/`) para habilitar y configurar el módulo SSL.
5. Exponer el puerto 443, que es el puerto estándar para tráfico HTTPS.

Ejemplo básico de DockerFile para Apache:

```
#Instalar módulo y habilitar sitio SSL por defecto y crear directorio para
certificados
RUN a2enmod ssl && a2ensite default-ssl && mkdir /etc/apache2/ssl

# Copiar certificado y clave privada
COPY certificados_usuario/certificado_tuusuariougr.crt /etc/apache2/ssl/
certificado_tuusuariougr.crt
COPY certificados_usuario/certificado_tuusuariougr.key /etc/apache2/ssl/
certificado_tuusuariougr.key

# Configurar los permisos adecuados
RUN chmod 600 /etc/apache2/ssl/certificado_tuusuariougr.crt

# Incluir la configuración SSL
COPY tuusuariougr-apache-ssl.conf /etc/apache2/sites-available/tuusuariougr-
apache-ssl.conf

# Exponer el puerto HTTPS
EXPOSE 443
```



### Parte 3: Configuración del balanceador de carga Nginx con SSL

En esta sección, configuraremos Nginx para balancear la carga entre los servidores Apache configurados con SSL, gestionando las conexiones HTTPS de manera efectiva. Trabajaremos en el directorio local **P3-tuusuariougr-nginx** para luego montar el directorio y/o copiar al contenedor los archivos necesarios.

#### 3.1 - Archivo de configuración SSL - **tuusuariougr-nginx-ssl.conf**

Este archivo debe contener la configuración para que Nginx maneje el tráfico HTTPS. Además de la configuración de la práctica 2 que definía el upstream, server, etc. se debe incluir:

1. Configurar el servidor para escuchar en el puerto 443 con SSL.
2. Especificar la ruta al certificado SSL y a la clave privada.

Ejemplo básico de archivo de configuración para Nginx:

```
server {
    listen 443 ssl;
    ssl_certificate      /etc/nginx/ssl/certificado_tuusuariougr.crt;
    ssl_certificate_key  /etc/nginx/ssl/certificado_tuusuariougr.key;

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

#### 3.2 - Dockerfile para Nginx con SSL - **DockerFileNginxP3**

La imagen de Docker adecuada, el archivo **DockerFileNginxP3** debe crearse en el directorio **P3-tuusuariougr-nginx** y llamarse **tuusuariougr-nginx-image:p3**. Podría contener instrucciones para:

1. Partir de la imagen base oficial de Nginx.
2. Copiar los archivos de certificado SSL y la clave privada al contenedor.
3. Incluir el archivo de configuración de Nginx que establece las conexiones SSL.
4. Exponer el puerto 443 para el tráfico HTTPS.

Ejemplo básico de Dockerfile para Nginx:

```
# Crear directorio para SSL
RUN mkdir -p /etc/nginx/ssl

# Copiar certificados SSL al contenedor
COPY certificados_usuario/certificado_tuusuariougr.crt /etc/nginx/ssl/certificado_tuusuariougr.crt
COPY certificados_usuario/certificado_tuusuariougr.key /etc/nginx/ssl/certificado_tuusuariougr.key

# Incluir configuración de Nginx para SSL
COPY tuusuariougr-nginx-ssl.conf /etc/nginx/nginx.conf

# Exponer el puerto HTTPS
EXPOSE 443
```



#### Parte 4: Configuración de Docker Compose para la Granja Web con SSL

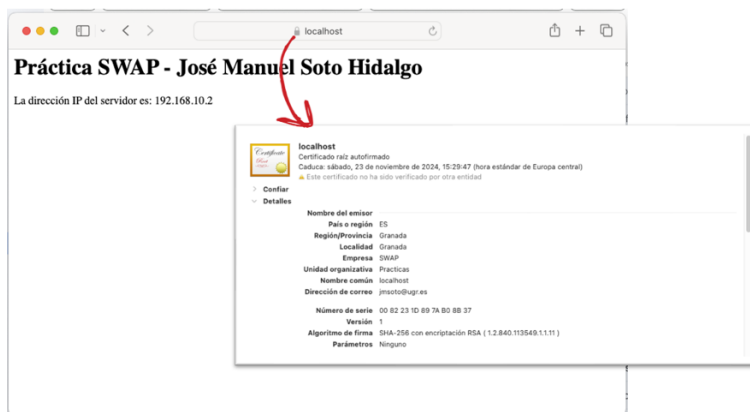
Esta parte se configura DockerCompose para definir el escenario de la granja web, incluyendo servidores Apache con SSL y el balanceador de carga Nginx con SSL con conexiones a red\_web y red\_servicios para un despliegue coordinado y seguro.

- Definición de Servicios en docker-compose.yml:
  - Definir un servicio para cada instancia de Apache que incluya:
    - Imagen construida a partir del DockerFileApacheP3 y que se llame **tuusuarioUGR-apache-image:p3**.
    - Nombre del contendedor: webX donde X es el número de contenedor del 1 al 8.
    - Volumen para montar el directorio local **web\_tuusuarioUGR** en la ruta por defecto de Apache para servir el index.php.
    - Volumen para montar el directorio local **certificados\_tuusuarioUGR** en la carpeta /etc/apache2/ssl/.
    - Conexión a las redes red\_web y red\_servicios con las IP indicadas en el esquema de la práctica.
  - Definir un servicio llamado balanceador-nginx-ssl que incluya:
    - Construcción de la imagen a partir del DockerFileNginxP3 y que se llame **tuusuarioUGR-nginx-image:p3**.
    - Volumen para montar el archivo tuusuariougr-nginx-ssl.conf en el contenedor en /etc/nginx/nginx.conf.
    - Volumen para montar el directorio local **certificados\_tuusuarioUGR** en la carpeta /etc/nginx/ssl/.
    - Asignación de dirección IP estática 192.168.10.50 en la red red\_web.
    - Dependencia establecida con los servicios de Apache para garantizar el orden correcto de despliegue.

#### Parte 5: Verificación y Pruebas del escenario con SSL

En esta sección realizará el despliegue del escenario y se verificará.

- Despliegue y Ejecución:
  - Ejecuta **docker-compose up -d** para arrancar todos los servicios definidos en el archivo **docker-compose.yml**.
  - Confirma que los servicios están activos y funcionando.
- Verificación y Pruebas:
  - Verifica que Nginx distribuya adecuadamente las solicitudes HTTPS entre los diferentes contenedores Apache.
  - Comprueba el certificado SSL que has creado.



## Evaluación

Para superar la práctica se deben realizar las siguientes **tareas básicas**:

### B1: Preparación del Entorno de Trabajo

- Crear directorios específicos para los archivos de configuración

### B2: Creación de Certificados SSL

- Generación correcta del certificado SSL y la clave privada con OpenSSL siguiendo las especificaciones dadas.

### B3: Configuración de Servidores Web Apache con SSL

- Redacción adecuada del Dockerfile para los servidores Apache con soporte SSL.
- Configuración precisa del archivo de host virtual de Apache para atender peticiones HTTPS.

### B4: Configuración del Balanceador de Carga Nginx con SSL

- Elaboración correcta del Dockerfile y la configuración de Nginx para gestionar conexiones SSL.

### B5: Docker Compose para la Granja Web con SSL

- Desarrollo de un archivo docker-compose.yml funcional que despliegue todos los servicios y configuraciones definidos en la práctica.

### B6: Verificación y Pruebas del Escenario con SSL

- Ejecución efectiva del despliegue usando Docker Compose y verificación del correcto funcionamiento del entorno SSL.

Se proponen, opcionalmente, las siguientes **tareas avanzadas**:

### A1: Exploraciones Avanzadas de creación de certificados SSL

- Generar un certificado raíz (CA) y uno o más certificados intermedios (subCA) para entender cómo se construye una cadena de confianza. Utilizar el certificado intermedio para firmar los certificados de los servidores, simulando una estructura más realista y segura que se encuentra en entornos de producción.
  - *Recomendaciones: Primero, genera clave privada de la CA y úsala para crear un certificado autofirmado que servirá como CA raíz. Segundo, crea una Autoridad Certificadora Intermedia (subCA) similar a la CA, crea una clave privada para la subCA y utilízala para generar una solicitud de certificado (CSR) de la subCA que será firmada por la CA raíz, estableciendo una cadena de confianza.*

### A2: Optimización de la configuración SSL en los servidores web

- Optimizar la configuración SSL en Apache para mejorar tanto la seguridad como el rendimiento de las conexiones seguras. Para ello, deshabilitar protocolos inseguros y cifrados débiles que pueden ser explotados por ataques. Por ejemplo, permitir o denegar protocolo TLS v1, TLS v1.1 y TLS v1.2 así como cifrados MD5, RC4 y 3DES. Justifica la elección de cada uno.

### A3: Configuración de Caché de Sesiones SSL y Tickets de Sesión en el balanceador

- Configurar Nginx para utilizar caché de sesiones SSL y tickets de sesión para mejorar la velocidad de las conexiones seguras repetidas, reduciendo el tiempo necesario para negociar la seguridad de la conexión.

### A4: Optimización de conexiones HTTPS y cifrado en el balanceador

- Personalizar los protocolos SSL/TLS y suites de cifrado para equilibrar seguridad y rendimiento y activar HTTP/2 para mejorar la eficiencia de las conexiones HTTPS. Por ejemplo, protocolos TLSv1.2 y TLSv1.3 así como cifrado ECDH, AESGCM, AES256, etc. Justifica la elección de cada uno.



## Normas de entrega

Se desarrollará un documento siguiendo el guion de la práctica y **detallando** e indicando, en su caso, los **aspectos básicos y avanzados realizados**, comandos de terminal ejecutados, resultados de ejecución, etc.

Por ejemplo, si se ha realizado la tarea básica de configuración del entorno, el documento .pdf con la memoria de prácticas debe aparecer una sección titulada: *Tareas Básicas - B2: Creación de Certificados SSL* donde aparezcan detalladas las configuraciones, explicaciones sobre ellas. De igual forma, si por ejemplo, se han realizado tareas avanzadas sobre automatizaciones con Scripts, debe aparecer *Tareas Avanzadas - A3: Configuración de Caché de Sesiones SSL y Tickets de Sesión en el balanceador*, detalles y explicaciones de las configuraciones.

Se recomienda utilizar herramientas de control de Tiempo (por ejemplo, clockify) para contabilizar el tiempo de dedicado a la realización de la práctica.

Se deja a **libre elección** la **estructura y formato** del documento el cual reflejará el correcto desarrollo de la práctica a modo de diario/tutorial siguiendo los puntos descritos anteriormente. Asimismo, se recomienda incluir capturas de pantalla que reflejen el correcto desarrollo de los distintos apartados de la práctica. La **primera página** del documento debe incluir, al menos, **nombre, apellidos y tiempo dedicado a la práctica** medido con herramientas de control de tiempo.

Para la entrega se habilitará una tarea en PRADO cuya entrega debe seguir **OBLIGATORIAMENTE** el formato especificado.

1. Un archivo .pdf con el documento desarrollado siguiendo el formato **ApellidosNombreP3.pdf**
2. Un archivo .zip con los distintos archivos de configuraciones, carpetas, etc. necesarios para la ejecución de la práctica siguiendo el formato **ApellidosNombreP3.zip**

## Uso de Inteligencia Artificial Generativa.

Para cada práctica es **OBLIGATORIO** usar herramientas de IA generativa (ChatGPT, Copilot u otras) e incluir enlace al chat/prompt utilizado. También se debe analizar y justificar el resultado que proporciona la herramienta con el resultado final que opta el estudiante para la práctica.

Es **OBLIGATORIO** incluir en el guion una sección titulada: **"Análisis propuesta IA"** donde se incluya enlace al chat/prompt con las consulta/as realizada/as, resultado que proporciona la IA y un párrafo con un análisis detallado del resultado proporcionado.

## Evaluación

La práctica se evaluará mediante el uso de rúbrica específica (accesible por el estudiante en la tarea de entrega) y una defensa final de prácticas.

Cuestiones sobre la calificación obtenida en cada práctica se realizarán **UNICAMENTE** en la sesión dedicada a recuperación/defensa al final de curso.

La detección de prácticas copiadas implicará el suspenso inmediato de todos los implicados en la copia (tanto del autor del original como de quien las copió). **OBLIGATORIO ACEPTAR LICENCIA EULA DE TURNITIN** en la entrega. Si la memoria supera un 40% de copia Turnitin implicará el suspenso automáticamente.



## Rúbrica

Criterios de Evaluación	Excelente (Puntuación máxima)	Bueno (75% de la puntuación máxima)	Adecuado (50% de la puntuación máxima)	Deficiente (25% de la puntuación máxima)	No Realizado (0 puntos)
<b>Tareas básicas</b>					
<b>B1. Preparación del Entorno de Trabajo (5 puntos)</b>	Directorios específicos correctamente creados y documentados detalladamente. <b>(5 puntos)</b>	Directorios creados con pequeños errores documentales. <b>(3.75 puntos)</b>	Directorios parcialmente completados o con errores significativos. <b>(2.5 puntos)</b>	Directorios configurados incorrectamente con errores graves. <b>(1.25 puntos)</b>	No se crearon directorios. <b>(0 puntos)</b>
<b>B2. Creación de certificados SSL (20 puntos)</b>	Certificados y claves generados correctamente y con especificaciones precisas. <b>(20 puntos)</b>	Pequeñas desviaciones de las especificaciones pero certificados funcionales. <b>(15 puntos)</b>	Certificados con errores que necesitan corrección pero no afectan a la funcionalidad general. <b>(10 puntos)</b>	Certificados generados con errores graves que impiden su uso. <b>(5 puntos)</b>	No realizado. <b>(0 puntos)</b>
<b>B3. Configuración de Apache con SSL (30 puntos)</b>	Archivo de configuración y Dockerfile para Apache con SSL perfectamente implementados. <b>(30 puntos)</b>	Implementación correcta con pequeñas inexactitudes. <b>(22.5 puntos)</b>	Errores que afectan parcialmente la funcionalidad de SSL en Apache. <b>(15 puntos)</b>	Configuraciones incorrectas que afectan la funcionalidad general. <b>(7.5 puntos)</b>	No realizado. <b>(0 puntos)</b>
<b>B4. Configuración de Nginx con SSL (25 puntos)</b>	Archivo de configuración y Dockerfile para Nginx con SSL perfectamente implementados. <b>(25 puntos)</b>	Implementación correcta con pequeñas inexactitudes. <b>(18.75 puntos)</b>	Errores que afectan parcialmente la funcionalidad de SSL en Nginx. <b>(12.5 puntos)</b>	Configuraciones incorrectas que afectan la funcionalidad general. <b>(6.25 puntos)</b>	No realizado. <b>(0 puntos)</b>
<b>B5. Docker Compose y Despliegue de la Granja Web con SSL (15 puntos)</b>	docker-compose.yml implementado sin errores, todos los servicios desplegados correctamente. <b>(15 puntos)</b>	Pequeñas inexactitudes pero funcionalidad asegurada. <b>(11.25 puntos)</b>	Errores que afectan parcialmente la funcionalidad del despliegue. <b>(7.5 puntos)</b>	Errores graves que impiden un despliegue correcto. <b>(3.75 puntos)</b>	No realizado. <b>(0 puntos)</b>
<b>B6. Verificación del escenario SSL (5 puntos)</b>	Todos los servicios SSL desplegados correctamente y pruebas de seguridad realizadas. <b>(5 puntos)</b>	Mayoría verificada, con pequeños errores en las pruebas. <b>(3.75 puntos)</b>	Faltan servicios por verificar. <b>(2.5 puntos)</b>	Intento de verificación pero con errores graves. <b>(1.25 puntos)</b>	No realizado. <b>(0 puntos)</b>



Tareas avanzadas					
<b>A1. Cadena de Certificados de Confianza (20 puntos)</b>	CA y subCA creadas con precisión, incluyendo cadena de confianza correctamente implementada y documentada. (20 puntos)	Cadena de confianza establecida con menores errores o falta de detalle. (15 puntos)	Cadena de confianza parcialmente implementada o con errores significativos. (10 puntos)	Intento de creación de CA y subCA pero con errores graves. (5 puntos)	No realizado. (0 puntos)
<b>A2. Optimización de SSL en Servidores Web (15 puntos)</b>	Configuraciones de SSL en Apache optimizadas y probadas para rendimiento y seguridad. (15 puntos)	Optimizaciones realizadas con algunos errores menores o falta de análisis detallado. (11.25 puntos)	Configuraciones básicas de SSL implementadas sin análisis de impacto. (7.5 puntos)	Configuración con errores que afectan la eficacia del SSL. (3.75 puntos)	No realizado. (0 puntos)
<b>A3. Configuración de caché y tickets de sesión SSL en el balanceador (15 puntos)</b>	Caché de sesiones SSL y tickets de sesión implementados y documentados con precisión. (15 puntos)	Implementación correcta pero con pequeños errores o falta de optimización. (11.25 puntos)	Configuración básica de caché SSL y tickets de sesión sin ajustes avanzados. (7.5 puntos)	Configuraciones realizadas pero con errores que afectan el rendimiento. (3.75 puntos)	No realizado. (0 puntos)
<b>A4. Optimización de conexiones HTTPS y cifrado en el balanceador (15 puntos)</b>	Protocolos SSL/TLS y suites de cifrado optimizados y HTTP/2 activado correctamente. (15 puntos)	Optimizaciones de SSL y activación de HTTP/2 realizadas con inexactitudes menores. (11.25 puntos)	Implementación básica de optimizaciones SSL y HTTP/2. (7.5 puntos)	Intento de optimización con errores graves. (3.75 puntos)	No realizado. (0 puntos)
Documentación					
<b>Documentación de la Práctica (25 puntos)</b>	Memoria detallada, cuidada, bien estructurada, sin errores ortográficos. (25 puntos)	Memoria completa con algunos detalles menores faltantes o errores leves. (18.75 puntos)	Memoria completa pero con varias omisiones o errores ortográficos. (12.5 puntos)	Memoria incompleta, desorganizada o con numerosos errores. (6.25 puntos)	No realizada. (0 puntos)
<b>Análisis y Justificación de Resultados de IA Generativa (40 puntos)</b>	Análisis profundo y justificación detallada de las configuraciones y resultados de IA. (40 puntos)	Análisis adecuado con justificaciones superficiales o incompletas. (30 puntos)	Análisis básico con pocas justificaciones o reflexiones críticas. (20 puntos)	Intento de análisis pero con justificaciones inadecuadas. (10 puntos)	No realizado. (0 puntos)
<b>Comentarios en Configuraciones (20 puntos)</b>	Comentarios detallados y claros en todas las configuraciones. (20 puntos)	Comentarios adecuados en la mayoría de las configuraciones. (15 puntos)	Algunos comentarios útiles, pero falta de detalle o claridad. (10 puntos)	Comentarios escasos o poco claros. (5 puntos)	Sin comentarios en las configuraciones. (0 puntos)
Penalizaciones					
<b>Requisitos de entrega</b>	Cumple: 0 puntos		No cumple: -15 puntos		
<b>Entrega en plazo fijado</b>	En plazo: 0 puntos		Un poco tarde (horas): -10 puntos	Algo tarde (1 día): -15 puntos	Muy tarde (varios días): -20 puntos
<b>Porcentaje de copia en Turnitin</b>	1-10%: 0 puntos	11-20%: -20 puntos	21-30%: -30 puntos	31-40%: -40 puntos	Más del 40%: Suspensión automática

