**Duración:** 2 sesiones

**Requisitos Previos:**

- Haber completado satisfactoriamente las Prácticas 1, 2 y 3 o tener experiencia equivalente en configuración de balanceadores de carga y certificados SSL con Docker.
- Conocimientos básicos sobre seguridad web y cortafuegos.

# Introducción

La seguridad en línea es un pilar fundamental para mantener la integridad y la disponibilidad de los sitios web, especialmente en entornos de granjas web donde múltiples servidores trabajan en conjunto para ofrecer servicios críticos. En esta práctica, nos centraremos en fortalecer las defensas de nuestra granja web mediante la implementación meticulosa de reglas de IPTABLES. Aprenderemos a manejar el tráfico de red de forma proactiva, configurando reglas que no solo controlen el acceso de entrada y salida, sino que también mitiguen riesgos potenciales. Estas políticas de IPTABLES serán diseñadas para proteger nuestros servidores contra ataques comunes y filtrar tráfico no deseado, asegurando así la continuidad y la seguridad del servicio que proporcionamos.

**Objetivos de la Práctica:**

Esta práctica tiene como meta reforzar la seguridad de nuestra infraestructura web utilizando contenedores Docker y aplicando conceptos básicos de cortafuegos.

1. Implementar reglas básicas de IPTABLES para mejorar la seguridad del servidor.
2. Configurar políticas de IPTABLES para gestionar y filtrar el tráfico de red de forma eficiente.
3. Entender y aplicar prácticas de seguridad para proteger los servidores web.

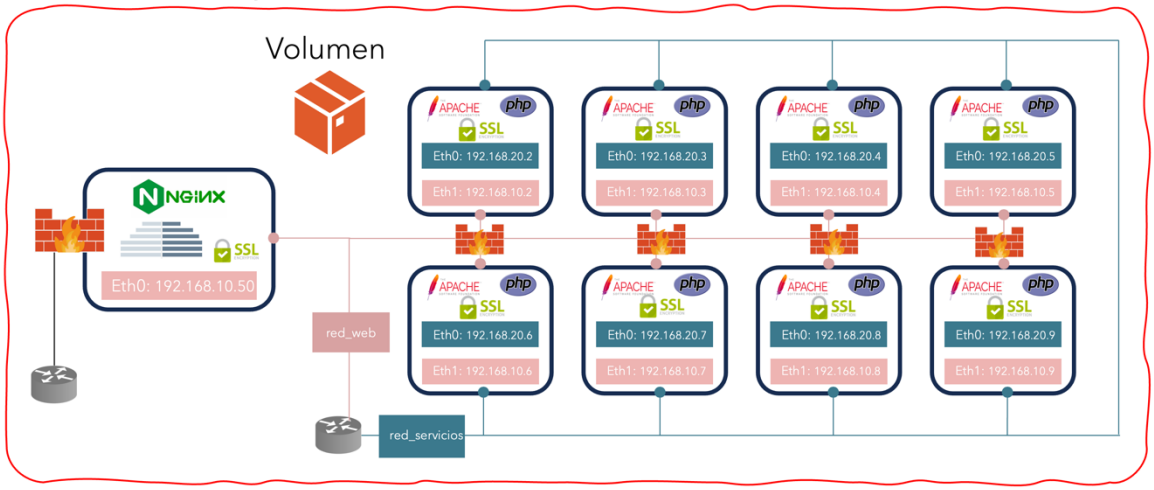
**Descripción de la Práctica:**

Se abordará la configuración y manejo de IPTABLES para reforzar la seguridad en una granja web. Se aprenderá a diseñar e implementar políticas de defensa a través de iptables, permitiendo un control detallado sobre el tráfico de red tanto entrante como saliente. Se desarrollarán scripts que encapsularán estas reglas específicas, centradas en filtrar accesos no deseados y bloquear amenazas potenciales, como ataques de denegación de servicio y escaneos de puertos. Estos scripts se integrados en los servidores web mediante ajustes en Dockerfile y docker-compose.yml, con el objetivo de proteger los servidores sin comprometer su rendimiento o accesibilidad.

A lo largo de la práctica, se realizarán pruebas para verificar la efectividad de las políticas de seguridad implementadas, asegurando que las configuraciones de IPTABLES cumplen con los requisitos de protección sin interferir en la funcionalidad de la granja web. Se aprenderá no solo a aplicar configuraciones técnicas avanzadas, sino también a evaluar y ajustar las políticas de seguridad en función de las necesidades específicas del entorno y las amenazas emergentes, fomentando un enfoque crítico y adaptativo hacia la seguridad en infraestructuras web complejas.


©José Manuel Soto Hidalgo

# Esquema general de la práctica:




# Desarrollo:

Se pretende seguir desarrollando un entorno seguro añadiendo configuraciones de cortafuegos. Continuando con la práctica anterior donde configurábamos SSL en los servidores web Apache y Nginx, añadiremos configuraciones de cortafuegos con IPTABLES.

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

En esta parte se establecerá el espacio de trabajo a través de directorios específicos para el conjunto de reglas y script necesarios para la configuración de cortafuegos. Partiendo de la estructura de directorios de la práctica anterior:

- Crea un directorio en tu máquina local llamado **P4-tuusuariougr-certificados** donde incluirá los certificados SSL que creaste en la práctica 3, otro directorio otro directorio llamado **P4-tuusuariougr-nginx** para trabajar con los archivos de configuración de nginx al igual que en la práctica 3 y un directorio **P4-tuusuariougr-apache** para trabajar con los archivos de configuración de apache.
- Dentro del directorio **P4-tuusuariougr-apache**, crea un directorio llamado **P4-tuusuariougr-iptables-web** donde se crearán los scripts necesarios para IPTABLES.
- 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: Definición de políticas de seguridad a los servidores web

En esta parte definiremos e implementaremos las políticas de seguridad que vamos a utilizar en la práctica para las réplicas de los servidores web. Trabajaremos en el directorio **P4-tuusuariougr-iptables-web** y se deberá crear un script llamado `tuusuariougr-iptables-web.sh` que implemente reglas IPTABLES con las siguientes especificaciones:

- **Denegación implícita de todo el tráfico.** Política por defecto de cualquier paquete entrante, saliente o reenvío que no cumpla con una regla explícita será descartado automáticamente. Esta política ayuda a proteger el sistema contra accesos no autorizados. Es una práctica de seguridad conservadora y restrictiva.


©José Manuel Soto Hidalgo

- **Manejar el tráfico de red entrante basado en el estado de las conexiones.** Permitir conexiones establecidas y relacionadas al tráfico entrante para permitir paquetes que sean parte de una conexión ya existente o paquetes que estén iniciando una nueva conexión pero que estén asociados a una ya existente. Es una configuración común para asegurar que las respuestas a solicitudes iniciadas por el sistema local sean permitidas.
- **Manejar el tráfico de red saliente basado en el estado de las conexiones.** Permitir conexiones nuevas al tráfico saliente para que el host envíe paquetes asociados con nuevas solicitudes, así como aquellos que forman parte de conexiones ya establecidas o relacionadas. Esto es crucial para permitir que las aplicaciones en el host inicien y mantengan conexiones de red sin interrupción.
- **Manejar tráfico de red de la misma máquina**. Permitir el tráfico que el host envía a sí mismo, tanto entrante como saliente. Es una configuración común para la operación normal del sistema, ya que muchos procesos internos del sistema operativo y aplicaciones usan esta interfaz para comunicarse entre sí.
- **Manejar tráfico HTTP y HTTPS.** Permitir el tráfico TCP entrante al puerto 80 y 443 respectivamente pero solo proveniente del balanceador de carga.

*NOTA: No olvides darle permisos de ejecución al script tuusuariougr-iptables-web.sh*

## Parte 2: Configuración de Servidores Web con las reglas IPTABLES

En esta parte configuraremos los servidores web finales para implementar las políticas de seguridad definidas en la parte 1. Trabajaremos en el directorio **P4-tuusuariougr-apache.**

### Parte 2.1 – Script de entrada – `entrypoint.sh`

Crea un script llamado `entrypoint.sh` dentro del directorio **P4-tuusuariougr-apache** que ejecute el script con las reglas IPTABLES y luego ejecute el comando principal del contenedor: exec "$@".

Ejemplo básico de entrypoint.sh

```
#!/bin/bash
# Ejecuta el script de iptables
./tuusuariougr-iptables-web.sh

# Luego, ejecuta el comando principal del contenedor
exec "$@"
```

*NOTA: No olvides darle permisos de ejecución al script entrypoint.sh*

### Parte 2.2 - DockerFile con ENTRYPOINT – `DockerFileApacheP4`

La imagen de Docker adecuada, el archivo **DockerFileApacheP4** debe crearse en el directorio **P4-tuusuariougr-apache** y llamarse **tuusuariougr-apache-image:p4**. Debe contener instrucciones para:

1. Partir de una imagen para Apache creada en la práctica 3.
2. Instalar IPTABLES.
3. Copia el script de entrada entrypoint.sh y el script de reglas tuusuariougr-iptables-web.sh al contenedor.
4. Da permisos de ejecución a los scripts en el contenedor.
5. Configura el ENTRYPOINT con el script de entrada.

©José Manuel Soto Hidalgo

Ejemplo básico de DockerFile para Apache:

```
#Instalar iptables
RUN apt-get update && apt-get install -y iptables

# Copiar script de entrada entrypoint y script de reglas iptables
COPY ./P4-tuusuariougr-iptables-web/entrypoint.sh /entrypoint.sh
COPY ./P4-tuusuariougr-iptables-web/tuusuaruiougr-iptables-web.sh
/tuusuariougr-iptables-web.sh

# Dar permisos de ejecución a los scripts
RUN chmod +x /entrypoint.sh /jmsoto-iptables-web.sh

# Configurar el script de entrada
ENTRYPOINT ["/entrypoint.sh"]
```

## Parte 3: Configuración de Docker Compose para la Granja Web con IPTABLES

Esta parte se configura DockerCompose para definir el escenario de la granja web, incluyendo servidores Apache con SSL y reglas IPTABLES, el balanceador de carga Nginx con SSL con conexiones a red_web y red_servicios para un despliegue coordinado y seguro. Parte de Docker-compose.yml de la práctica 3 y añade capacidades de administración de red a los servicios web.

- Definición de Servicios en docker-compose.yml:
  - Definir un servicio para cada instancia de Apache que incluya:
    - Imagen construida a partir del DockerFileApacheP4 y que se llame tuusuarioUGR-apache-image:p4.
    - 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.
    - Capacidades de administración de red para poder ejecutar IPTABLES.
      ```
      cap_add:
        -NET_ADMIN
      ```
  - Definir un servicio llamado balanceador-nginx-ssl que incluya:
    - Construcción de la imagen a partir del DockerFileNginxP4 y que se llame tuusuarioUGR-nginx-image:p4.
    - 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.


©José Manuel Soto Hidalgo

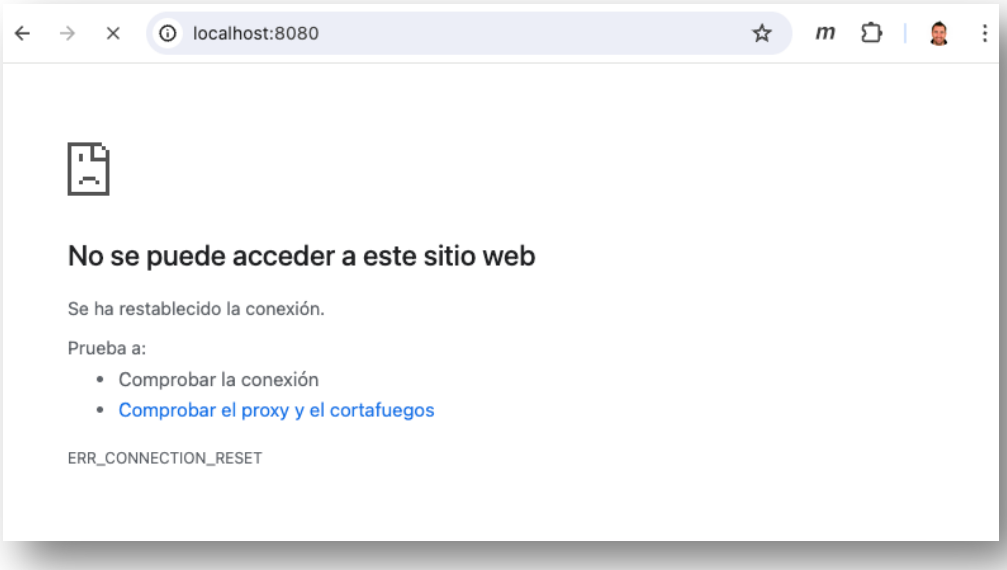
Parte 4: Verificación y Pruebas del escenario con IPTABLES

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 HTTP y HTTPS entre los diferentes servidores Apache y que no puedas acceder directamente a los servidores web.

Peticiones al balanceador:


Peticiones a un servidor web apache:






©José Manuel Soto Hidalgo

# Evaluación

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

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

### B1. Preparación del Entorno de Trabajo

- Crear y preparar directorios específicos para los archivos de configuración de IPTABLES y certificados SSL previamente generados. Esto incluye:
  - Un directorio para scripts IPTABLES específicos.

### B2. Creación y Configuración de Scripts IPTABLES

- Desarrollar y escribir un script tuusuariougr-iptables-web.sh que establecerá las reglas de IPTABLES en los servidores web. Este script debe:
  - Establecer políticas por defecto para rechazar todo tráfico no explícitamente permitido.
  - Permitir conexiones entrantes y salientes específicas necesarias para la operación normal de los servidores web.
  - Asegurar que las conexiones entre el balanceador de carga y los servidores web estén adecuadamente configuradas para permitir solo tráfico HTTP y HTTPS.

### B3. Implementación de Scripts IPTABLES en Docker

- Integrar el script IPTABLES en la configuración de Docker de los servidores web Apache. Esto implica:
  - Modificar los Dockerfiles para incluir y ejecutar el script IPTABLES al iniciar los contenedores.
  - Asegurar que los scripts tienen los permisos adecuados para ejecutarse y modificar las reglas de IPTABLES dentro de los contenedores.

### B4. Configuración de Docker Compose

- Modificar y adaptar el archivo docker-compose.yml de la práctica anterior para incluir los cambios necesarios que permitan la ejecución de IPTABLES dentro de los contenedores de Apache y Nginx. Esto incluirá:
  - La configuración para montar los scripts y directorios necesarios dentro de los contenedores.
  - Asegurar que los contenedores tienen las capacidades de red necesarias (CAP_NET_ADMIN) para modificar IPTABLES.

### B5. Verificación y Pruebas

- Ejecutar y verificar el entorno configurado. Esto implica:
  - Desplegar los servicios usando Docker Compose.
  - Verificar que las reglas de IPTABLES están activas y funcionando como se espera dentro de los contenedores.
  - Confirmar que el tráfico no permitido está siendo correctamente bloqueado, mientras que el tráfico legítimo fluye según lo esperado.

©José Manuel Soto Hidalgo

Se proponen, opcionalmente, las siguientes **tareas avanzadas,** las cuales deben, aparte de implementarse, analizarse y demostrar la efectividad de la implementación ante las simulaciones de ataques.

A1: Definir e implementar políticas de seguridad en el balanceador de carga

- Definir e implementar políticas de seguridad en el balanceador de carga. Podría incluir, además de denegación implícita:
  - Limitar el número de conexiones simultáneas.
  - Bloquear escaneo de puertos.
  - Usar módulo string para mitigar ataques de inyección SQL o XSS a través de las peticiones HTTP.
  - Etc.

A2. Configuración Avanzada de IPTABLES para DDoS

- Implementar reglas avanzadas en IPTABLES para mitigar ataques de Denegación de Servicio Distribuido (DDoS). Esto podría incluir:
  - Limitación de la tasa de conexiones nuevas por IP para evitar la saturación de los recursos del servidor.
  - Uso de módulos como recent para detectar y bloquear rápidamente el tráfico anómalo y prevenir inundaciones de IPs.
  - Configuración de umbrales y reglas específicas que identifiquen patrones de tráfico asociados a ataques comunes de DDoS.
  - Protección Contra Ataques de Fragmentación.
  - Etc.

A3: Simular ataques a la granja web y configuraciones de seguridad realizadas

- Simular un ataque DDoS para probar la configuración de seguridad de la granja web.
- Simular un ataque de inyección SQL o XSS
- Simular otros ataques.

# Normas de entrega

Se entregará un documento .pdf con el desarrollo de la práctica según el guion **detallando** e indicando, en su caso, los **aspectos básicos y avanzados realizados**, comandos de terminal ejecutados, así como las configuraciones o soluciones proporcionadas por la IA generativa y las configuraciones o soluciones que finalmente utiliza el estudiante junto con su análisis crítico. 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 y Configuración de Scripts IPTABLES* donde aparezcan detalladas las configuraciones, explicaciones sobre ellas y resultados proporcionados por la IA generativa y **un análisis de éstos**. De igual forma, si por ejemplo, se han realizado tareas avanzadas sobre automatizaciones con Scripts, debe aparecer *Tareas Avanzadas – A2. Configuración Avanzada de IPTABLES para DDoS,* detalles de las configuraciones, explicaciones sobre ellas y resultados proporcionados por la IA generativa y **un análisis de éstos**.


©José Manuel Soto Hidalgo

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.

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 ApellidosNombreP4.pdf
2. Un archivo .zip con los distintos archivos de configuraciones, carpetas, etc. necesarios para la ejecución de la práctica siguiendo el formato ApellidosNombreP4.zip

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.

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.

Las faltas de ortografía en la redacción se penalizarán con hasta 2 puntos de la nota de la práctica.

©José Manuel Soto Hidalgo

## 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) |
|---|---|---|---|---|---|
| **Tareas básicas** | | | | | |
| **B1. Preparación del Entorno de Trabajo (5 puntos)** | Directorios específicos correctamente creados y documentados detalladamente. **(5 puntos)** | Directorios creados con pequeños errores documentales. **(3.75 puntos)** | Directorios parcialmente completados o con errores significativos. **(2.5 puntos)** | Directorios configurados incorrectamente con errores graves. **(1.25 puntos)** | No se crearon directorios. **(0 puntos)** |
| **B2: Definición de Políticas de Seguridad en los Servidores Web (30 puntos)** | Políticas de seguridad definidas e implementadas correctamente con documentación completa. **(30 puntos)** | Implementación adecuada con pequeños errores o falta de detalle. **(22.5 puntos)** | Implementación con errores menores que no comprometen la seguridad básica. **(15 puntos)** | Políticas mal implementadas con errores significativos que afectan la seguridad. **(7.5 puntos)** | No realizado. **(0 puntos)** |
| **B3: Configuración de Servidores Web con IPTABLES (20 puntos)** | Configuración perfectamente realizada y servidores funcionando según lo esperado. **(20 puntos)** | Configuración funcional con algunos errores no críticos. **(15 puntos)** | Configuraciones con errores que requieren ajustes menores. **(10 puntos)** | Errores graves que afectan el funcionamiento de los servidores. **(5 puntos)** | No realizado. **(0 puntos)** |
| **B4. Docker Compose y Despliegue de la Granja Web con IPTABLES (10 puntos)** | docker-compose.yml configurado sin errores y todos los servicios funcionan correctamente. **(10 puntos)** | Configuración funcional con errores menores. **(7.5 puntos)** | Algunos problemas en la configuración que afectan parcialmente la funcionalidad. **(5 puntos)** | Errores graves que impiden un despliegue correcto. **(2.5 puntos)** | No realizado. **(0 puntos)** |
| **B5. Verificación del escenario con IPTABLES (5 puntos)** | Pruebas completas que verifican la efectividad de las configuraciones de seguridad. **(5 puntos)** | Mayoría verificada, con pequeños errores en las pruebas. **(3.75 puntos)** | Faltan servicios por verificar. **(2.5 puntos)** | Intento de verificación pero con errores graves. **(1.25 puntos)** | No realizado. **(0 puntos)** |
| **Tareas avanzadas** | | | | | |
| **A1. Políticas avanzadas en el balanceador (20 puntos)** | Políticas avanzadas implementadas y efectividad demostrada mediante simulación. **(20 puntos)** | Implementación adecuada con algunos detalles menores faltantes. **(15 puntos)** | Implementación básica sin análisis profundo de efectividad. **(10 puntos)** | Intento de implementación con errores que limitan la efectividad. **(5 puntos)** | No realizado. **(0 puntos)** |
| **A2. Configuración Avanzada de IPTABLES para DDoS (15 puntos)** | Reglas avanzadas implementadas correctamente con análisis de impacto detallado. **(15 puntos)** | Reglas implementadas con algunos errores menores. **(11.25 puntos)** | Reglas básicas implementadas sin ajustes avanzados. **(7.5 puntos)** | Implementación deficiente con errores significativos. **(3.75 puntos)** | No realizado. **(0 puntos)** |
| **A3. Simulación de ataques (15 puntos)** | Simulaciones de ataques realizadas y analizadas exhaustivamente. **(15 puntos)** | Simulaciones realizadas con análisis adecuado. **(11.25 puntos)** | Simulaciones básicas realizadas sin análisis detallado. **(7.5 puntos)** | Intento de simulaciones pero con fallos significativos. **(3.75 puntos)** | No realizado. **(0 puntos)** |


©José Manuel Soto Hidalgo

| | | | | | |
|---|---|---|---|---|---|
| **Documentación** | | | | | |
| **Documentación de la Práctica (25 puntos)** | 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)** |
| **Análisis y Justificación de Resultados de IA Generativa (40 puntos)** | 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)** |
| **Comentarios en Configuraciones (15 puntos)** | Comentarios detallados y claros en todas las configuraciones. **(15 puntos)** | Comentarios adecuados en la mayoría de las configuraciones. **(11.25 puntos)** | Algunos comentarios útiles, pero falta de detalle o claridad. **(7.5 puntos)** | Comentarios escasos o poco claros. **(3.75 puntos)** | Sin comentarios en las configuraciones. **(0 puntos)** |
| **Penalizaciones** | | | | | |
| **Requisitos de entrega** | Cumple: 0 puntos | | No cumple: -15 puntos | | |
| **Entrega en plazo fijado** | En plazo: 0 puntos | | Un poco tarde (horas): -10 puntos | Algo tarde (1 día): -15 puntos | Muy tarde (varios días): -20 puntos |
| **Porcentaje de copia en Turnitin** | 1-10%: 0 puntos | 11-20%: -20 puntos | 21-30%: -30 puntos | 31-40%: -40 puntos | Más del 40%: Suspenso automático |


©José Manuel Soto Hidalgo