

23-04-2025

Rendimiento, Calidad e Incidencias de Servicios Web

Unidad de trabajo 6

*Francisco Javier Otero Herrero
Grupo ATU*

Rendimiento, Calidad e Incidencias de Servicios Web

Rendimiento, Calidad e Incidencias de Servicios Web

Pregunta 1. Utilizando el programa para Windows, NStalker Free Edition X de detección de vulnerabilidades. Instálalo y pásalo a una web que conozcas. Muestra el resultado del escaneo. Agregar a la solución las capturas de pantalla de los resultados. Describe los pasos que vayas realizando junto a las capturas de pantalla.

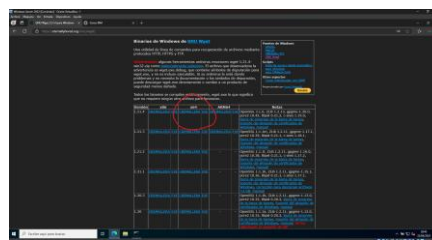
Debo informar de que **NStalker Free Edition X** ya no está disponible ni se recomienda su uso. Las herramientas de seguridad cambian rápidamente, y las versiones antiguas pueden ser obsoletas o incluso inseguras.

Sin embargo, podemos utilizar otra herramienta más actualizada, para ello, vamos a ver el proceso general de instalación y uso.

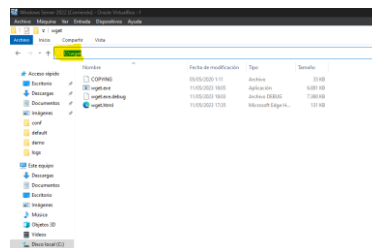
Es una herramienta de escaneo de vulnerabilidades web ampliamente utilizada como **Wget (para enumeración básica)**.

Vamos a comenzar con la descarga e instalación de esta herramienta, para descargar:

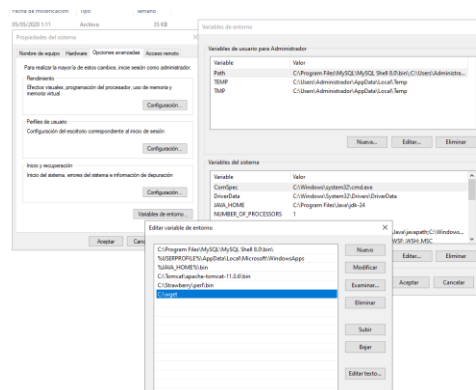
GNU Wget,1.21.4 para Windows



En mi caso he instalado la versión de 64 bits. Una vez descargo el fichero he descomprimido su contenido en **C:/wget**.



Añadir esta ruta al **Path (variables de entorno)**, es opcional pero recomendable para poder hacer uso de esta herramienta a través de la línea de comandos.

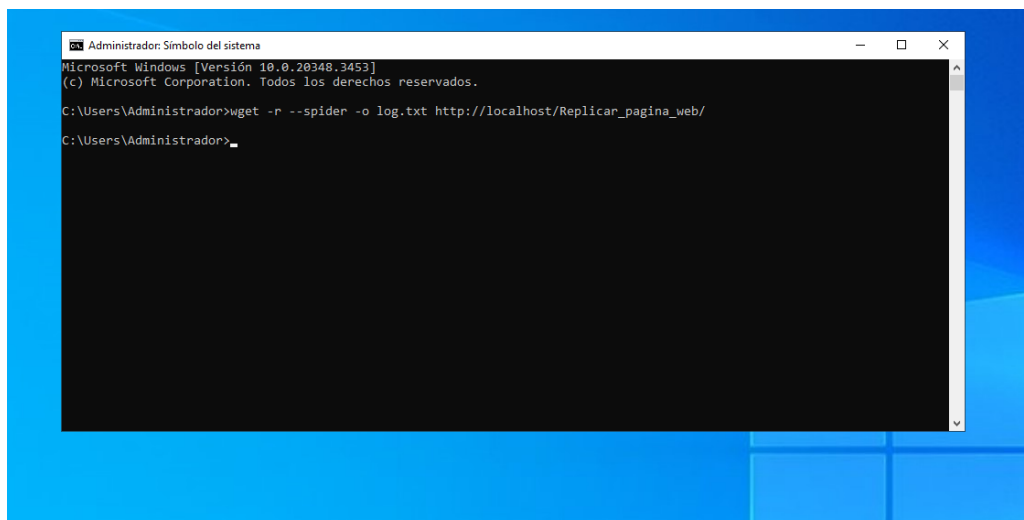


Rendimiento, Calidad e Incidencias de Servicios Web

Una vez llegados a este punto es momento de probar esta herramienta desde la línea de comandos. En mi caso he copiado al servidor Apache del servidor de Windows, una de las páginas web que hice durante el curso de iniciación en Python para probar sobre ella esta herramienta. La web es la siguiente:



Una web sencilla con varios enlaces, botones e imágenes. Para ver los resultados de esta que esta herramienta proporciona sobre esta web lanzamos el siguiente comando desde CMD: “`wget -r --spider -o log.txt http://tuweb.local/`”



- ✓ **-r:** Recursivo (sigue los enlaces).
- ✓ **--spider:** No descarga los archivos, solo los verifica.
- ✓ **-o log.txt:** Guarda la salida en un archivo llamado log.txt.
- ✓ **http://localhost/Replicar_pagina_web:** URL de la web local.

Esta herramienta nos creará un fichero .txt, el cual guarda en la siguiente ruta o ubicación del equipo “`C:\Users\Administrador\Documents`”

Rendimiento, Calidad e Incidencias de Servicios Web

Debido a que mis conocimientos sobre esta herramienta son nulos, es la primera vez que la uso, le he preguntado a **Gemini (IA)** que me explicase el contenido del log que genera esta herramienta y aquí están los resultados:

El log muestra las acciones de **wget** al intentar explorar la página web `http://localhost/Replicar_pagina_web/`. Usaste el modo "**spider**" (**--spider**), lo que significa que **wget** intentó identificar la estructura del sitio (enlaces, archivos) sin descargar realmente el contenido completo (excepto cuando parecía necesario para seguir explorando).

- **Inicio del escaneo:**

Spider mode enabled. Check if remote file exists. Indica que **wget** está en modo "**araña**".

--2025-04-22 20:37:29-- http://localhost/Replicar_pagina_web/ **wget** comienza a explorar la página principal.

Resolving localhost (localhost)... ::1, 127.0.0.1 **wget** resuelve el nombre "localhost" a las direcciones IP (IPv6 e IPv4).

Connecting to localhost (localhost)::1:80... connected. **wget** se conecta al servidor web local en el **puerto 80 (HTTP)**.

HTTP request sent, awaiting response... 200 OK. El servidor responde con un código **200 OK**, lo que significa que la página principal se encontró correctamente.

Length: 4355 (4,3K) [text/html]. La página principal tiene un tamaño de **4355 bytes** y es de tipo **HTML**.

Remote file exists and could contain links to other resources -- retrieving. **wget** decide obtener la página principal porque podría contener enlaces a otros archivos.

- **Descarga temporal de la página principal:**

--2025-04-22 20:37:29-- http://localhost/Replicar_pagina_web/ **wget** vuelve a solicitar la página principal (esto puede ocurrir internamente para analizar los enlaces).

Reusing existing connection to [localhost]:80. **wget** reutiliza la conexión anterior.

HTTP request sent, awaiting response... 200 OK. El servidor responde de nuevo con **200 OK**.

Length: 4355 (4,3K) [text/html]. El tamaño y tipo de la página son los mismos.

Saving to: 'localhost/Replicar_pagina_web/index.html.tmp.tmp' **wget** guarda temporalmente la página principal en un archivo.

OK 100% 208M=0s. Indica el progreso de la descarga (**muy rápido en local**).

2025-04-22 20:37:29 (208 MB/s) 'localhost/Replicar_pagina_web/index.html.tmp.tmp' saved [4355/4355]. La descarga se completa.

Rendimiento, Calidad e Incidencias de Servicios Web

- **Búsqueda de robots.txt:**

Loading robots.txt; please ignore errors. **wget** intenta encontrar el archivo robots.txt, que indica a los rastreadores web qué partes del sitio no deben explorar.

--2025-04-22 20:37:29-- http://localhost/robots.txt **wget** solicita el archivo robots.txt.

HTTP request sent, awaiting response... 404 Not Found. El servidor responde con un código 404, lo que significa que el archivo robots.txt no existe.

2025-04-22 20:37:29 ERROR 404: Not Found. **wget** registra el error 404.

Removing localhost/Replicar_pagina_web/index.html.tmp.tmp. **wget** elimina el archivo temporal de la página principal.

- **Exploración de otros recursos:**

Spider mode enabled. Check if remote file exists. **wget** continúa en modo araña.

--2025-04-22 20:37:29-- http://localhost/Replicar_pagina_web/styles/style.css **wget** explora el archivo CSS.

HTTP request sent, awaiting response... 200 OK El archivo CSS se encuentra.

Length: 5631 (5,5K) [text/css] Tamaño y tipo del archivo CSS.

Remote file exists and could contain links to other resources -- retrieving. **wget** descarga temporalmente el CSS.

Removing localhost/Replicar_pagina_web/styles/style.css.tmp.tmp. Se elimina el archivo temporal CSS.

- **Exploración de imágenes:**

Spider mode enabled. Check if remote file exists. **wget** explora varias imágenes (img2.jpg, img3.jpg, img.jpg, img4.jpg, python.jpg).

HTTP request sent, awaiting response... 200 OK Las imágenes se encuentran.

Remote file exists but does not contain any link -- not retrieving. **Importante: wget** detecta que estas imágenes no contienen enlaces a otros recursos, por lo que no las descarga. Esto es una optimización del **modo "spider"**.

Removingtmp. **wget** intenta eliminar archivos temporales (que en realidad nunca creó completamente, debido al **--spider**).

unlink: No such file or directory Aparece este error porque **wget** está intentando eliminar un archivo temporal que nunca llegó a crear completamente. Puedes ignorar estos errores **"unlink"**.

Rendimiento, Calidad e Incidencias de Servicios Web

- **Finalización:**

Found no broken links. **wget** informa que **no** encontró enlaces rotos (enlaces a páginas que no existen).

FINISHED --2025-04-22 20:37:29-- El escaneo finaliza.

Total wall clock time: 0,07s El escaneo tardó **0.07 segundos (muy rápido, al ser local)**.

Downloaded: 2 files, 9,8K in 0s (232 MB/s) **wget** descargó 2 archivos (la página principal y el CSS) con un tamaño total de 9.8 KB.

- **Resumen de lo que el log nos dice:**

wget exploró la estructura de tu sitio web local. Encontró la página principal (**index.html**) y un archivo CSS (**style.css**). No encontró un archivo robots.txt. Encontró varias imágenes, pero no las descargó porque no contenían enlaces. No encontró enlaces rotos.

- **Interpretación para prácticas de seguridad:**

- ✓ **Enumeración exitosa:** **wget** logró enumerar los archivos y directorios de tu sitio web. Esto es una parte importante del reconocimiento en las pruebas de seguridad. Un atacante también intentaría enumerar archivos para encontrar información sensible o posibles puntos de entrada.
- ✓ **Falta de robots.txt:** La ausencia de robots.txt significa que los rastreadores web (y los atacantes que usan herramientas similares) pueden explorar todo el sitio sin restricciones. Para prácticas, considera la importancia de un robots.txt bien configurado.
- ✓ **Estructura de archivos:** El log revela la estructura de tu sitio (directorios styles/, assets/image/). Esto ayuda a un atacante a entender dónde buscar posibles vulnerabilidades.

Imágenes no exploradas: Que **wget** no explore las imágenes es normal en modo "spider", pero recuerda que las imágenes también pueden contener metadatos o vulnerabilidades (aunque no es el foco principal de este escaneo).

Rendimiento, Calidad e Incidencias de Servicios Web

Pregunta 2. Utilizando una herramienta de Apache como ApacheBench prueba a realizar una medición del rendimiento de tu servidor local.

En este apartado, vamos hacer la prueba en el servidor de Windows, el cual tiene Apache instalado por lo tanto ya disponemos de esta herramienta.

Para probar si la tenemos, lanzamos el siguiente comando (“ab”) que podemos ver en la imagen, si muestra la ayuda de la herramienta significa que funciona:

```

C:\Users\Administrador> C:\Apache24\bin\ab
C:\Apache24\bin\ab
ab: wrong number of arguments
Usage: ab (-h) [-n] [http://hostname[:port]/path]
Options are:
-n requests      Number of requests to perform
-c concurrency  Number of multiple requests to make at a time
                 seconds to wait to spend on benchmarking
                 This implies -n 10000
-t timeout      Seconds to wait for each response
                 Default is 30 seconds
-s addresssize  Size of TCP send/receive buffer, in bytes
-b address      Address to bind to when making outgoing connections
-p postfile     File containing data to POST. Remember also to set -T
-P postfile     File containing data to PUT. Remember also to set -T
-T content-type Content type. Must be used for POST/PUT data, eg:
                 'application/x-www-form-urlencoded'
                 Default is: text/plain
-v verbosity    How much troubleshooting info to print
-l             Print out results in HTML tables
-L            Use HEAD instead of GET
-x attributes   String to insert as table attributes
-y attributes   String to insert as tr attributes
-z attributes   String to insert as td or th attributes
-c attributes   Add cookie, eg. 'Apache=1234' (remember!)
-h attributes   Add arbitrary header line, eg. 'Accept-encoding: gzip'
                 Inserted after all normal header lines. (remember!)
-A attributes   Add Basic WWW authentication, the attributes
                 are a colon separated username and password.
-P attributes   Add Basic Proxy Authentication, the attributes
                 are a colon separated username and password.
-x proxy:port   Proxyserver and port number to use
-v             Print version number and exit
-d            Use HTTP keepalive feature
-e            Do not show percentiles served table.
-l            Do not show confidence estimators and warnings.
-r            Do not show progress when doing more than 100 requests
-f            Accept variable document length (use this for dynamic pages)
-f filename    Output collected data to filename format file.
-F filename    Output CSV file with percentages served
-m            Don't wait on socket receive errors.
-h            Method name
-h            Display usage information (this message)
  
```

Está funcionando por lo que vamos a probar sobre localhost directamente el siguiente comando de nuevo desde CMD:

```
ab -n 1000 -c 100 http://localhost/
```

Esto enviará 1000 peticiones a http://localhost, con 100 peticiones concurrentes. Los resultados fueron:

```

C:\Apache24\bin\ab -n 1000 -c 100 http://localhost/
This is ApacheBench, Version 2.3 <$Revision: 1023142 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests

Server Software:      Apache/2.4.6.3
Server Hostname:      localhost
Server Port:          80

Document Path:        /
Document Length:       72 bytes

Concurrency Level:    100
Time taken for tests:  0.781 seconds
Complete requests:    1000
Failed requests:       0
Total transferred:     32000 bytes
HTML transferred:     72000 bytes
Requests per second:  1280.79 [#/sec] (mean)
Time per request:      78.077 [ms] (mean)
Time per request:      0.781 [ms] (mean, across all concurrent requests)
Transfer rate:         418.25 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median max
Connect:    0   0  1.4     0   17
Processing: 17  72 11.1    78  94
Waiting:    14  72 11.2    78  94
Total:      17  72 11.0    78  94

Percentage of the requests served within a certain time (ms)
 50%    78
 60%    78
 75%    78
 80%    78
 90%    79
 95%    79
 98%    79
 99%    79
100%   94 (longest request)
  
```

Es otra de las herramientas que tampoco he usado nunca por lo que pregunte de nuevo a Gemini por estos resultados y que me explicara qué significado tienen.

Rendimiento, Calidad e Incidencias de Servicios Web

- **La respuesta de Gemini:**

➤ **Información general:**

- *This is ApacheBench, Version 2.3 ...:* Indica la versión de ApacheBench que estás utilizando.
- *Benchmarking localhost (be patient):* Informa que se está probando el servidor local.
- *Completed 100 requests ... Finished 1000 requests:* Muestra el progreso de las peticiones completadas.

➤ **Información del servidor:**

- *Server Software: Apache/2.4.63:* Indica que el software del servidor es Apache, versión 2.4.63.
- *Server Hostname: localhost:* El nombre del servidor es localhost.
- *Server Port: 80:* El servidor está escuchando en el puerto 80 (el puerto predeterminado para HTTP).
- *Document Path: /:* La ruta del documento solicitado es la raíz ("/").
- *Document Length: 72 bytes:* El tamaño del documento (la respuesta del servidor) es de 72 bytes. Esto es muy pequeño y sugiere que estás probando una página muy simple o que ab no está obteniendo el contenido completo por alguna razón.

➤ **3. Métricas de rendimiento principales:**

- *Concurrency Level: 100:* El número de peticiones concurrentes (el valor de -c).
- *Time taken for tests: 0.781 seconds:* El tiempo total que tardó ab en completar todas las 1000 peticiones. ¡Es muy rápido!
- *Complete requests: 1000:* El número total de peticiones completadas con éxito.
- *Failed requests: 0:* El número de peticiones que fallaron. ¡Excelente, no hubo fallos!
- *Total transferred: 328000 bytes:* La cantidad total de datos transferidos durante las pruebas.
- *HTML transferred: 72000 bytes:* La cantidad de datos HTML transferidos. (1000 requests * 72 bytes = 72000 bytes).

Rendimiento, Calidad e Incidencias de Servicios Web

- *Requests per second: 1280.79 [#/sec] (mean): Métrica clave: El número promedio de peticiones que el servidor pudo manejar por segundo. En este caso, ¡1280.79 peticiones por segundo! Esto indica un muy buen rendimiento para una página tan simple.*
- *Time per request: 0.781 [ms] (mean): El tiempo promedio que tardó el servidor en responder a una petición (en milisegundos). 0.781 milisegundos es extremadamente rápido.*
- *Time per request: 0.781 [ms] (mean, across all concurrent requests): El tiempo promedio por petición, considerando la concurrencia. Es similar al anterior en este caso, lo que sugiere que la concurrencia no afectó significativamente el tiempo de respuesta.*
- *Transfer rate: 410.25 [Kbytes/sec] received: La velocidad a la que se recibieron los datos (en kilobytes por segundo).*

➤ **4. Tiempos de conexión:**

- *Connect: min mean[+/-sd] median max: Estadísticas sobre el tiempo empleado en establecer la conexión. En este caso, el tiempo de conexión es prácticamente 0 ms en promedio, lo que es típico para localhost.*
- *Processing: 17 72 11.1 78 94: Estadísticas sobre el tiempo empleado por el servidor en procesar la petición. El tiempo promedio de procesamiento es de 72 ms.*
- *Waiting: 14 72 11.2 78 94: Estadísticas sobre el tiempo que el cliente tuvo que esperar por la respuesta del servidor. Similar al tiempo de procesamiento.*
- *Total: 17 72 11.0 78 94: El tiempo total transcurrido desde que se envió la petición hasta que se recibió la respuesta.*

➤ **5. Porcentaje de peticiones servidas dentro de un tiempo determinado:**

- *Esta tabla muestra la distribución de los tiempos de respuesta. Por ejemplo, el 50% de las peticiones se sirvieron en 78 ms, el 90% en 79 ms, y el 100% (la petición más lenta) tardó 94 ms. Esto da una idea de la consistencia del rendimiento.*

Rendimiento, Calidad e Incidencias de Servicios Web

Resumen e interpretación

Excelente rendimiento: Tu servidor está funcionando muy bien para esta prueba. Puede manejar un gran número de peticiones por segundo con tiempos de respuesta muy rápidos.

Página muy simple: El tamaño de la respuesta (**72 bytes**) indica que la página que estás probando es extremadamente pequeña. Esto influye en los resultados de rendimiento.

Baja latencia (localhost): Al probar en localhost, la latencia de la red es mínima, lo que contribuye a los tiempos de respuesta rápidos.

Consistencia: Los tiempos de respuesta son bastante consistentes, como se muestra en la tabla de porcentajes.

Posibles limitaciones: Los resultados son específicos para esta prueba y esta página. El rendimiento podría ser diferente con páginas más complejas, mayor carga del servidor o tráfico de red real.

➤ **Para profundizar en las pruebas:**

- **Prueba con una página más compleja:** Crea una página de prueba con más contenido (HTML, CSS, JavaScript, imágenes) para obtener resultados más realistas.
- **Aumenta la carga:** Prueba con más peticiones y más usuarios concurrentes para ver cómo se comporta el servidor bajo mayor estrés.
- **Prueba diferentes URLs:** Prueba diferentes páginas de tu sitio web.
- **Monitoriza el servidor:** Monitoriza los recursos del servidor (CPU, memoria, disco) durante la prueba para identificar posibles cuellos de botella.

Rendimiento, Calidad e Incidencias de Servicios Web

Pregunta 3. Realiza una descripción y clasificación de los elementos determinantes del rendimiento de un servidor web.

El rendimiento de un servidor web es crucial para proporcionar una buena experiencia de usuario, asegurar la disponibilidad del sitio y optimizar los recursos. Hay muchos factores que influyen en cómo de rápido y eficiente es un servidor web. Podemos clasificarlos en varias categorías principales:

1. Hardware:

✓ Procesador (CPU):

- **Descripción:** La CPU es el "cerebro" del servidor. Procesa todas las peticiones y ejecuta el código del servidor (como scripts PHP, Python, etc.).
- **Clasificación:**
 - *Velocidad de reloj:* Medida en GHz, indica la velocidad a la que la CPU puede ejecutar instrucciones.
 - *Número de núcleos:* Más núcleos permiten al servidor manejar múltiples peticiones simultáneamente.
 - *Caché:* Una memoria de alta velocidad que almacena datos de uso frecuente para un acceso más rápido.
- **Impacto en el rendimiento:** Una CPU potente puede procesar más peticiones rápidamente, reduciendo la latencia.

✓ Memoria RAM:

- **Descripción:** La RAM almacena datos temporalmente para un acceso rápido. El sistema operativo, las aplicaciones del servidor y los datos en uso se cargan en la RAM.
- **Clasificación:**
 - *Capacidad:* La cantidad de RAM disponible (en GB).
 - *Velocidad:* La velocidad a la que la RAM puede transferir datos.
- **Impacto en el rendimiento:** Suficiente RAM evita que el servidor tenga que usar el disco duro para la memoria virtual (swap), que es mucho más lento, mejorando significativamente la velocidad.

Rendimiento, Calidad e Incidencias de Servicios Web

✓ **Almacenamiento (Discos duros o SSDs):**

- **Descripción:** Almacena el sistema operativo, las aplicaciones del servidor, los archivos del sitio web (HTML, CSS, imágenes, etc.) y las bases de datos.
- **Clasificación:**
 - *Tipo:*
 - HDD (Disco Duro Mecánico): Más económicos, pero más lentos debido a las partes mecánicas.
 - SSD (Unidad de Estado Sólido): Mucho más rápidos que los HDDs, ya que no tienen partes móviles.
 - NVMe SSD: Un tipo de SSD aún más rápido que se conecta directamente a la placa base a través del bus PCIe.
 - *Velocidad de lectura/escritura:* La velocidad a la que los datos se pueden leer y escribir en el disco.
 - *Capacidad:* La cantidad de almacenamiento disponible.
- **Impacto en el rendimiento:** Los SSDs mejoran drásticamente los tiempos de carga de las páginas, especialmente cuando se accede a muchos archivos pequeños o a bases de datos. Los HDDs pueden convertirse en un cuello de botella.

✓ **Red:**

- **Descripción:** La infraestructura de red que conecta el servidor con los clientes.
- **Clasificación:**
 - *Ancho de banda:* La cantidad de datos que se pueden transferir por unidad de tiempo (Mbps o Gbps).
 - *Latencia:* El tiempo que tarda un paquete de datos en viajar desde el cliente al servidor y viceversa.
 - *Infraestructura:* Calidad de los cables, switches, routers, etc.
- **Impacto en el rendimiento:** Un ancho de banda suficiente es esencial para manejar el tráfico del sitio web. La alta latencia puede ralentizar los tiempos de carga, incluso con buen ancho de banda.

Rendimiento, Calidad e Incidencias de Servicios Web

2. Software:

✓ **Sistema Operativo:**

- **Descripción:** El sistema operativo (como Linux o Windows Server) gestiona los recursos del servidor.
- **Clasificación:**
 - *Eficiencia:* Algunos sistemas operativos están mejor optimizados para el rendimiento del servidor que otros.
 - *Configuración:* La forma en que se configura el sistema operativo (por ejemplo, la gestión de procesos, la memoria) puede tener un gran impacto.
- **Impacto en el rendimiento:** Un sistema operativo eficiente y bien configurado puede maximizar el uso de los recursos del servidor.

✓ **Servidor Web:**

- **Descripción:** El software (como Apache, Nginx o IIS) que responde a las peticiones de los clientes (navegadores web).
- **Clasificación:**
 - *Arquitectura:* La forma en que el servidor web maneja las conexiones (por ejemplo, basado en procesos, basado en hilos, basado en eventos).
 - *Configuración:* La configuración del servidor web (por ejemplo, el manejo de la caché, la compresión, los módulos) es fundamental para el rendimiento.
- **Impacto en el rendimiento:** Un servidor web eficiente y bien configurado puede manejar muchas peticiones simultáneamente y entregar contenido rápidamente.

✓ **Lenguajes de programación del lado del servidor:**

- **Descripción:** Lenguajes como PHP, Python, Ruby, etc., que se ejecutan en el servidor para generar contenido dinámico.
- **Clasificación:**
 - *Eficiencia del código:* Un código bien escrito se ejecuta más rápido.
 - *Frameworks:* El uso de frameworks puede facilitar el desarrollo, pero también puede afectar al rendimiento si no se usan correctamente.
- **Impacto en el rendimiento:** Un código del lado del servidor ineficiente puede ralentizar significativamente el tiempo de respuesta del servidor.

Rendimiento, Calidad e Incidencias de Servicios Web

✓ **Base de datos:**

- **Descripción:** Almacena y gestiona los datos del sitio web (por ejemplo, información de los usuarios, productos, etc.).
- **Clasificación:**
 - Sistema de gestión de bases de datos (SGBD): (como MySQL, PostgreSQL, MariaDB).
 - Diseño de la base de datos: Una base de datos bien diseñada es más eficiente.
 - Optimización de consultas: Las consultas SQL eficientes son cruciales para el rendimiento.
 - Caché de la base de datos: Almacenar en caché los resultados de las consultas frecuentes.
- **Impacto en el rendimiento:** Las operaciones de la base de datos pueden ser un cuello de botella importante. Una base de datos optimizada es esencial para un buen rendimiento.

3. Contenido del Sitio Web

✓ **Tamaño del sitio web:**

- **Descripción:** El tamaño de las páginas HTML, las imágenes, los archivos CSS y JavaScript.
- **Clasificación:**
 - Optimización de imágenes: Comprimir las imágenes sin perder demasiada calidad.
 - Minificación de código: Reducir el tamaño de los archivos CSS y JavaScript eliminando espacios y comentarios innecesarios.
- **Impacto en el rendimiento:** Los archivos grandes tardan más en descargarse, lo que aumenta el tiempo de carga de la página.

✓ **Complejidad de la página:**

- **Descripción:** La cantidad de elementos en la página, la cantidad de peticiones HTTP que requiere y la cantidad de código del lado del cliente (JavaScript) que se ejecuta.

Rendimiento, Calidad e Incidencias de Servicios Web

- **Clasificación:**
 - *Número de peticiones HTTP: Reducir el número de peticiones combinando archivos CSS y JavaScript.*
 - *Ejecución de JavaScript: Optimizar el código JavaScript para que se ejecute de forma eficiente.*
- **Impacto en el rendimiento:** Las páginas complejas requieren más recursos del servidor y tardan más en cargarse en el navegador del cliente.

✓ **Caché:**

- **Descripción:** Almacenar copias de los archivos del sitio web en el servidor, en los servidores proxy o en el navegador del cliente para que no sea necesario descargarlos de nuevo cada vez.
- **Clasificación:**
 - *Caché del servidor: Almacenar en caché el contenido generado dinámicamente.*
 - *Caché del navegador: Indicar al navegador del cliente que almacene en caché los archivos estáticos.*
 - *CDN (Red de Distribución de Contenidos): Distribuir el contenido a través de servidores ubicados en diferentes ubicaciones geográficas.*
- **Impacto en el rendimiento:** La caché puede reducir drásticamente el tiempo de carga de la página, especialmente para los visitantes que regresan.

4. Factores de Red:

✓ **Ancho de banda:**

- **Descripción:** La cantidad de datos que se pueden transferir entre el servidor y los clientes en un período de tiempo determinado.
- **Clasificación:**
 - *Ancho de banda del servidor: La capacidad de la conexión del servidor a Internet.*
 - *Ancho de banda del cliente: La capacidad de la conexión del cliente a Internet.*
- **Impacto en el rendimiento:** El ancho de banda limita la velocidad a la que se pueden descargar los archivos del sitio web.

Rendimiento, Calidad e Incidencias de Servicios Web

✓ **Latencia:**

- **Descripción:** El tiempo que tarda un paquete de datos en viajar desde el servidor al cliente y viceversa.
- **Clasificación:**
 - *Latencia de la red:* El retraso causado por la distancia geográfica, los enrutadores y otros dispositivos de red.
 - *Congestión de la red:* El retraso causado por el exceso de tráfico en la red.
- **Impacto en el rendimiento:** La latencia afecta el tiempo de respuesta del servidor y puede hacer que las páginas se carguen lentamente, incluso con buen ancho de banda.

✓ **DNS:**

- **Descripción:** El Sistema de Nombres de Dominio (DNS) traduce los nombres de dominio (como <https://www.google.com/search?q=google.com>) a direcciones IP (como 172.217.160.142).
- **Clasificación:**
 - *Tiempo de resolución de DNS:* El tiempo que tarda un servidor DNS en resolver un nombre de dominio.
 - *Servidores DNS rápidos:* Usar servidores DNS rápidos puede acelerar el tiempo de carga de la página.
- **Impacto en el rendimiento:** Un DNS lento puede agregar un retraso significativo al inicio de la carga de la página.

Rendimiento, Calidad e Incidencias de Servicios Web

5. Ataques y Seguridad:

✓ **Ataques DDoS:**

- **Descripción:** Los ataques de Denegación de Servicio Distribuido (DDoS) inundan el servidor con tráfico, haciéndolo inaccesible para los usuarios legítimos.
- **Clasificación:**
 - *Ataques volumétricos:* Inundan el servidor con grandes cantidades de tráfico.
 - *Ataques de protocolo:* Explotan vulnerabilidades en los protocolos de red.
 - *Ataques a la capa de aplicación:* Atacan aplicaciones específicas del servidor web.
- **Impacto en el rendimiento:** Los ataques DDoS pueden degradar gravemente o incluso detener el funcionamiento del servidor.

✓ **Vulnerabilidades de seguridad:**

- **Descripción:** Las vulnerabilidades en el software del servidor o en el código del sitio web pueden ser explotadas por los atacantes para realizar diversas acciones maliciosas, incluyendo la degradación del rendimiento.
- **Clasificación:**
 - *Inyección SQL:* Los atacantes insertan código SQL malicioso en las consultas de la base de datos.
 - *Cross-Site Scripting (XSS):* Los atacantes inyectan código JavaScript malicioso en las páginas web.
- **Impacto en el rendimiento:** Las vulnerabilidades explotadas pueden consumir recursos del servidor o interrumpir su funcionamiento.

Es importante considerar todos estos factores al optimizar el rendimiento de un servidor web. A menudo, la mejora del rendimiento requiere un enfoque integral que abarque tanto el hardware como el software, el contenido del sitio web y la infraestructura de red.

Rendimiento, Calidad e Incidencias de Servicios Web

Pregunta 4. Describe las técnicas de resolución de incidencias en entornos de producción de servicios Web de la forma más detallada posible.

Para dar una descripción detallada de las técnicas de resolución de incidencias en entornos de producción de servicios web, es importante estructurar la información en varias etapas y aspectos clave del proceso. Aquí te presento una descripción completa:

I. Detección y Monitorización

✓ Monitorización Proactiva:

- Herramientas de monitorización (e.g., Nagios, Zabbix, Prometheus) supervisan continuamente el estado del servidor, la red, la base de datos y las aplicaciones web.
- Se establecen umbrales para métricas clave (uso de CPU, memoria, espacio en disco, latencia de red, tiempos de respuesta de la aplicación).
- Las alertas automáticas notifican al equipo de operaciones cuando se superan los umbrales o cuando se detectan anomalías.

✓ Registro de eventos (Logging):

- Se implementan sistemas de registro centralizados para recopilar registros de eventos de diferentes componentes (servidor web, base de datos, sistema operativo, etc.).
- Los registros se analizan para identificar patrones, errores y advertencias que puedan indicar problemas.
- Herramientas de gestión de registros (e.g., ELK Stack - Elasticsearch, Logstash, Kibana) facilitan la búsqueda y el análisis de registros.

✓ Pruebas sintéticas:

- Se simulan transacciones de usuario para probar la disponibilidad y el rendimiento de las aplicaciones web.
- Se realizan pruebas periódicas para verificar que las funciones críticas del sitio web funcionan correctamente.

Rendimiento, Calidad e Incidencias de Servicios Web

✓ **Informes de usuarios:**

- Se establecen canales para que los usuarios informen de problemas (e.g., formularios de contacto, sistemas de tickets).
- Se recopila información detallada sobre los problemas reportados por los usuarios (e.g., mensajes de error, pasos para reproducir el problema).

II. Identificación y Diagnóstico

✓ **Recopilación de información:**

- Se recopila información detallada sobre la incidencia, incluyendo el momento en que ocurrió, los mensajes de error, los registros de eventos y los pasos para reproducir el problema.
- Se verifica si ha habido cambios recientes en el entorno (e.g., despliegues de código, actualizaciones de software).

✓ **Análisis de registros:**

- Se analizan los registros de eventos para identificar la causa raíz de la incidencia.
- Se utilizan herramientas de búsqueda y filtrado para encontrar mensajes de error relevantes y patrones inusuales.

✓ **Monitorización en tiempo real:**

- Se utilizan herramientas de monitorización para observar el comportamiento del sistema en tiempo real y correlacionar los síntomas con las métricas de rendimiento.
- Se identifican los componentes del sistema que están experimentando problemas (e.g., alto uso de CPU, consumo excesivo de memoria).

✓ **Pruebas de diagnóstico:**

- Se realizan pruebas específicas para verificar el funcionamiento de los diferentes componentes del sistema (e.g., pruebas de conectividad de red, pruebas de acceso a la base de datos).
- Se utilizan herramientas de diagnóstico (e.g., ping, traceroute, tcpdump) para identificar problemas de red.

Rendimiento, Calidad e Incidencias de Servicios Web

✓ **Reproducción del problema:**

- Se intenta reproducir la incidencia en un entorno de prueba para comprender mejor la causa raíz y desarrollar una solución.
- Se aíslan los componentes del sistema para determinar cuál está causando el problema.

III. Resolución y Recuperación

✓ **Soluciones temporales (Workarounds):**

- Se implementan soluciones temporales para restaurar rápidamente el servicio y minimizar el impacto en los usuarios.
- Las soluciones temporales no abordan la causa raíz del problema, pero permiten que el servicio funcione mientras se desarrolla una solución permanente.

✓ **Soluciones permanentes:**

- Se desarrolla e implementa una solución permanente para corregir la causa raíz de la incidencia.
- La solución puede implicar la corrección de errores en el código, la configuración de parámetros del sistema, la actualización de software o la sustitución de hardware.

✓ **Pruebas de la solución:**

- Se prueban exhaustivamente las soluciones temporales y permanentes en un entorno de prueba antes de implementarlas en producción.
- Se realizan pruebas de regresión para asegurar que la solución no introduce nuevos problemas.

✓ **Implementación de la solución:**

- Se implementa la solución en el entorno de producción siguiendo un plan de implementación cuidadosamente diseñado.
- Se monitoriza el sistema después de la implementación para asegurar que la solución ha resuelto la incidencia y no ha causado efectos secundarios.

Rendimiento, Calidad e Incidencias de Servicios Web

✓ **Recuperación:**

- Si el servicio se ha interrumpido, se realizan los pasos necesarios para restaurarlo a su estado normal.
- Se pueden utilizar copias de seguridad para restaurar datos perdidos o dañados.

IV. Comunicación y Coordinación

✓ **Comunicación interna:**

- Se establece un sistema de comunicación claro y eficiente entre los miembros del equipo de operaciones, los desarrolladores y otras partes interesadas.
- Se utilizan herramientas de comunicación (e.g., chat, correo electrónico, sistemas de tickets) para compartir información sobre la incidencia y coordinar los esfuerzos de resolución.

✓ **Comunicación externa:**

- Se informa a los usuarios y a otras partes interesadas sobre el estado de la incidencia y el progreso de la resolución.
- Se proporcionan actualizaciones periódicas y se gestionan las expectativas de los usuarios.

V. Análisis Post-Incidencia

✓ **Revisión de la incidencia:**

- Después de resolver la incidencia, se realiza una revisión para analizar la causa raíz, la respuesta del equipo y las lecciones aprendidas.
- Se identifican las áreas de mejora en los procesos, las herramientas o la infraestructura.
- Documentación:
 - Se documenta la incidencia, la causa raíz, la solución y las lecciones aprendidas en una base de conocimientos.
 - La documentación facilita la resolución de incidencias similares en el futuro y ayuda a mejorar la formación del equipo.

Rendimiento, Calidad e Incidencias de Servicios Web

✓ **Técnicas y Herramientas Específicas**

- Análisis de causa raíz (RCA): Metodologías como los "5 porqués" o los diagramas de Ishikawa se utilizan para identificar la causa fundamental de una incidencia.
- Gestión de incidentes: Se siguen los procesos ITIL u otros marcos de gestión de incidentes para estructurar la respuesta a las incidencias.
- Automatización: Se automatizan tareas repetitivas y procesos de resolución de incidencias para reducir el tiempo de respuesta y minimizar los errores humanos.
- Orquestación: Se utilizan herramientas de orquestación (e.g., Kubernetes, Docker Swarm) para gestionar y escalar los servicios web de forma eficiente.
- Infraestructura como código (IaC): Se gestiona la infraestructura del servidor web utilizando código, lo que facilita la replicación, la automatización y la recuperación.

La resolución de incidencias en entornos de producción de servicios web es un proceso complejo que requiere una combinación de habilidades técnicas, herramientas adecuadas y una comunicación efectiva. Un enfoque proactivo, una monitorización exhaustiva y una gestión estructurada de incidentes son esenciales para mantener la disponibilidad, el rendimiento y la fiabilidad de los servicios web.