Tarea 3.2 Pruebas de funcionamiento y rendimiento con Tomcat

Despliegue de Aplicaciones Web

Contenido

[FECHA DE INICIO: 19/12/2024 FECHA DE FINALIZACIÓN ESPERADA: 8/01/2025 2](#_Toc186750090)

[OBJETIVOS 2](#_Toc186750091)

[ENUNCIADO 2](#_Toc186750092)

[Requisitos 10](#_Toc186750093)

[Pasos de la práctica 1. Preparar el entorno 11](#_Toc186750094)

[2. Instalar y configurar JMeter 11](#_Toc186750095)

[4. Ejecutar la prueba 16](#_Toc186750096)

[5. Analizar los resultados 16](#_Toc186750097)

[6. Optimizar y repetir pruebas 18](#_Toc186750098)

[DOCUMENTACIÓN 20](#_Toc186750099)

**PRÁCTICA 3.2**

**PRUEBAS DE RENDIMIENTO DE APLICACIÓN WEB DESPLEGADA EN**

**TOMCAT**

# FECHA DE INICIO: 19/12/2024 FECHA DE FINALIZACIÓN ESPERADA: 8/01/2025

**RA ASOCIADO:** RA3. Implanta aplicaciones web en servidores de aplicaciones evaluando y aplicando criterios de configuración para su funcionamiento seguro. **CEs ASOCIADOS:** d, e, f, g, h.

# OBJETIVOS

* Aprender a implantar aplicaciones web en servidores de aplicaciones.
* Habituarse a la descripción del proceso.
* Creación de archivo war con netbeans en Linux usando Ant
* Evaluar el rendimiento de una aplicación web desplegada en un servidor Tomcat.
* Identificar el tiempo de respuesta, errores y capacidad de carga.

# ENUNCIADO

Se desarrollará la práctica en un Ubuntu Server que disponga de un servidor de Tomcat instalado y en un Ubuntu Desktop que tenga un IDE Netbeans y un servidor Tomcat sólo para desarrollo para crear el war.

Se facilitan en la plataforma los archivos ejecutables para la instalación del jdk, de netbeans y de tomcat, así como los archivos java (servlets) y jsp necesarios para la construcción del proyecto. Sólo debes cambiar el código para hacer que tras eliminar un artículo de la lista vuelva a mostrar el listado de artículos actualizado. Modifica esto tras construir el proyecto y desplegar en Tomcat.

Se deberá instalar el jdk, Tomcat 10 y Netbeans en el el Ubuntu Desktop. En este equipo se deberá crear el proyecto y generar el archivo war a partir de los archivos facilitados de la aplicación.

En Tomcat (tanto en el servidor como en el Ubuntu Desktop) habrá un usuario con capacidad de gestión de aplicaciones a través de la interfaz web, con credenciales: **useradmin / 4321.**

**PRIMERA PARTE** – Instalación de scripts en la máquina cliente para desarrrollo

Para usar los scripts de instalación del JDK, de Netbeans y de Tomcat 10:

1. **Guarda cada script en Linux en un archivo separado (puedes copiarlos como están o usar nano e incluir su contenido)**:
   * Para el JDK: instalar\_java21.sh
   * Para NetBeans: instalar\_netbeans22.sh o Para Tomcat 10: instalar\_tomcat10.sh

**He creado una carpeta compartida desde mi Windows al cliente de Ubuntu, y así puedo acceder a el contenido facilitado:**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. **Hazlos ejecutables**:
   * chmod +x instalar\_java21.sh instalar\_netbeans22.sh instalar\_tomcat10.sh

**Accedemos a la terminal en el directorio donde tenemos los archivos y les hacemos ejecutables a todos, utilizando el comando chmod +x \*.sh**

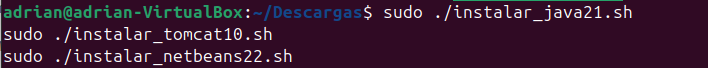
(He modificado los scripts. Estaban incorrectos debido a problemas con los espacios, los nombres de los archivos y errores en su funcionamiento. Además, uno de los scrils pts estaba vacío. Los he corregido y ahora deberían funcionar correctamente.)

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

1. **Ejecuta cada script con permisos de administrador**:
   * sudo ./instalar\_java21.sh
   * sudo ./instalar\_netbeans22.sh
   * sudo ./instalar\_tomcat10.sh

Utilizamos el comando sudo para ejecutar cada uno de los scripts:

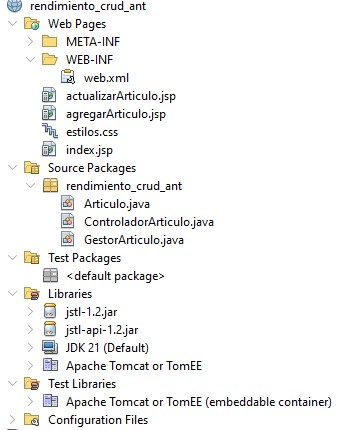


**SEGUNDA PARTE** – generar archivo war.

Para la asociación del servidor en Netbeans usaremos como Home y como Base ***/opt/tomcat***

Construye el proyecto de la aplicación con Netbeans, que desplegarás en el servidor Tomcat creado en la práctica anterior. Crea el archivo war en netbeans a partir de los servlets y JSP facilitados.

El proyecto en Netbeans tendrá la siguiente estructura:



Pasos que he realizado:

Lo primero, crear un proyecto tipo Web Application:

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Le damos un nombre:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Y seleccionamos Apache como tipo de servidor:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Elegimos como base para nuestro proyecto “/opt/tomcat”, e indicamos el nombre de usuario y la contraseña del usuario administrador:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Mi estructuración: Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**TERCERA PARTE** - Desplegar la aplicación.

Despliega la aplicación en el servidor Tomcat creado en la práctica anterior.

Se utilizará como nombre de dominio: **segundo.daw.iesagl.rendimiento**.

El servidor tendrá al menos una de sus interfaces en MODO PUENTE, de forma que se pueda acceder desde cualquier equipo del aula. Puedes utilizar el servidor creado en la práctica 1.

Agregamos la IP de nuestro servidor seguida del nombre de dominio que se indica en la práctica:

Texto

Descripción generada automáticamente

Hacemos clean and build en nuestro proyecto:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Y utilizamos el comando scp para copiar nuestro archivo WAR al servidor, y desplegarlo en Tomcat:

Texto

Descripción generada automáticamente

Abrimos en un navegador el dominio que hemos configurado, para verificar que funciona correctamente:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**CUARTA PARTE**- Pruebas de rendimiento con Apache JMeter

## Requisitos

1. Un servidor con Tomcat instalado y la aplicación web desplegada.
2. Apache JMeter instalado en el equipo que realizará las pruebas.
3. Configuración de red que permita el acceso al servidor Tomcat desde JMeter.

## Pasos de la práctica 1. Preparar el entorno

* Asegúrate de que Tomcat esté en funcionamiento y que la aplicación sea accesible desde un navegador.
* Verifica la URL base de la aplicación web (por ejemplo, http://<server\_ip>:8080/app).

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## 2. Instalar y configurar JMeter

* Descarga Apache JMeter desde [su página oficial.](https://jmeter.apache.org/)
* Extrae el archivo descargado y ejecuta el archivo jmeter.bat (Windows) o jmeter (Linux/MacOS) para abrir la interfaz gráfica.

Para ello, accedemos a la página oficial de apache para descargar Apache JMeter:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Una vez lo hayamos descargado y extraído, navegamos a la carpeta bin y ejecutamos JMeter con el comando ./jmeter, lo cual nos abrirá la interfaz gráfica de JMeter:

Interfaz de usuario gráfica

Descripción generada automáticamente

**3. Diseñar el plan de prueba en JMeter** 1. **Crear un grupo de usuarios:**

o Haz clic derecho en Test Plan > Add > Threads (Users) > Thread Group. o Configura:

* + - Número de Threads (Usuarios): Por ejemplo, 10 (incrementa progresivamente en pruebas posteriores).
    - Ramp-Up Period: 5 (segundos).
    - Loop Count: 10 (cantidad de repeticiones).

Siguiendo los pasos que se nos indica, creamos un grupo de usuarios:

Captura de pantalla de computadora

Descripción generada automáticamente

Establecemos el grupo en 10 usuarios, 5 segundos y 10 repeticiones:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. **Añadir un sampler HTTP Request:**
   * Haz clic derecho en Thread Group > Add > Sampler > HTTP Request.
   * Configura:
     + Server Name or IP: IP o dominio del servidor Tomcat.
     + Path: Ruta relativa de la aplicación (por ejemplo, /app/home). ▪ Método HTTP: GET o POST según la funcionalidad.

De nuevo, seguimos los pasos indicados para añadir un sampler HTTP request:

Texto

Descripción generada automáticamente

Y lo configuramos de la siguiente forma:

Captura de pantalla de un celular

Descripción generada automáticamente

1. **Añadir un listener para visualizar resultados:**
   * Haz clic derecho en Test Plan > Add > Listener > View Results in Table.

Por último, añadimos el listener:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

## 4. Ejecutar la prueba

* Presiona el botón verde "Iniciar" en la parte superior.
* Observa los resultados en el listener View Results in Table o en gráficos como Graph Results.

## 5. Analizar los resultados

* **Tiempo de respuesta promedio**: Observa los valores registrados en el listener.
* **Errores**: Verifica si hubo respuestas no esperadas (códigos HTTP diferentes a 2xx).
* **Capacidad máxima**: Incrementa el número de Threads (usuarios) y observa el punto en el que el tiempo de respuesta aumenta considerablemente o aparecen errores.

Pulsamos el botón verde para iniciar la prueba, lo cual nos mostrará lo siguiente:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Tal y como se puede ver en la captura, el tiempo de respuesta para la prueba con 10 usuarios fue muy bajo (de 8 ms), y no hubo ningún error.

Repetimos las pruebas, pero esta vez con 100 usuarios:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Esta vez, el tiempo de respuesta aumentó ligeramente, a 15 ms, y no hay errores aún.

Por último, he hecho una prueba con una cantidad mucho mayor de usuarios, con 100000000:

Imagen de la pantalla de un computador

Descripción generada automáticamente con confianza media

El tiempo de respuesta sigue sin aumentar incluso con un número tan exagerado de usuarios (se ha mantenido en 15 ms), y tampoco ha habido errores.

## 6. Optimizar y repetir pruebas

* Ajusta configuraciones en Tomcat si es necesario:

o Modifica el archivo server.xml para aumentar el tamaño del pool de conexiones:

<Connector port="8080" protocol="HTTP/1.1"

connectionTimeout="20000" maxThreads="200" />

* Prueba con diferentes escenarios de carga para identificar el límite de rendimiento.

Modificamos el archivo server.xml, para agregarle, dentro del Connector port, el atributo maxThreads="200", y el connectionTimeout="20000":

Texto

Descripción generada automáticamente

Una vez ejecutada la optimización, repetimos las pruebas:

Pantalla de computadora

Descripción generada automáticamente

En la prueba con 5 usuarios, el tiempo de respuesta disminuye hasta 5 ms.

En la prueba con 100 usuarios, también disminuye, quedando en 10 ms.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

En la prueba con 100000000 usuarios, el tiempo de respuesta no mejora, debido a los límites físicos del hardware y la capacidad del sistema.

Captura de pantalla de computadora

Descripción generada automáticamente

# DOCUMENTACIÓN

Deberás documentar los procedimientos indicando:

* Los pasos realizados (comandos, modificaciones a ficheros de configuración y rutas de los mismos, etc.).
* capturas de pantalla que demuestren que se han logrado los objetivos planteados.
* Presenta gráficos de tiempo de respuesta y errores para diferentes niveles de carga.
* Describe los límites de la aplicación y posibles mejoras.