



Centro de Electricidad Electrónica Y Telecomunicaciones (CEET)

Análisis y desarrollo en sistemas de información

DOCUMENTO DE SOFTWARE DE TESTEO

PRUEBAS DE SOFTWARE EFFECTIVE RECORD

Ficha:

2049891

Aprendices:

Andrés Felipe Sáenz Salazar

Lady Tatiana Chivito Caicedo

Andrés Esteban Sossa Rodríguez

Instructor:

Ing. Gerardo Mauricio Polania

JUNIO 02 de 2021, Bogotá D.C

DOCUMENTO DE SOFTWARE DE TESTEO

PRUEBAS DE SOFTWARE EFFECTIVE RECORD

SOFTWARE DE GESTIÓN Y ORGANIZACIÓN DE INVENTARIOS EN LAS MISCELÁNEAS

Aprendices:

Andrés Felipe Sáenz Salazar

Lady Tatiana Chivito Caicedo

Andrés Esteban Sossa Rodríguez

Ficha:

2049891

Servicio Nacional de Aprendizaje Sena

Centro de Electricidad Electrónica Y Telecomunicaciones (CEET)

Programa Análisis y desarrollo en sistemas de información

Contenido

Alcance	5
Definición de niveles de pruebas	6
Software de testeo	7
¿Qué es un SW de testeo?	7
¿Qué es JMeter?.....	7
¿Para que se usa JMeter?	7
Instalación y configuración del software	8
Paso 1: Descargar JMeter.....	8
Paso 2: Descargar o verificar versión de JDK	12
Paso 3: Configuración del entorno.....	13
Pruebas realizadas con el software.....	18
Pruebas no funcionales.....	18
Prueba de carga registrar usuarios.	18
Paso 1	18
Paso 2	19
Paso 3	19
Paso 4	19
Paso 5	20
Paso 6	20
Pruebas de estrés registrar productos.....	21
Paso 1	21
Paso 2	22
Paso 3	23
Paso 4	23
Paso 5	24
Paso 6	25
Paso 7	26
Paso 8	27
Pruebas funcionales	28

Pruebas de integración	28
Prueba funcional	29
Prueba 1	29
Prueba 2	30
Pruebas unitarias.....	31
Conclusiones	32
Anexos.....	33
VIDEO DE LA PRUEBA	33
BIBLIOGRAFÍA ELECTRÓNICA.....	33

Alcance

Este documento pretende dar conocer los tipos y etapas de pruebas de software que se van a desarrollar y a poner en marcha durante el desarrollo y ejecución del sistema de información, de igual forma pretende entregar las pautas y estrategias que se seguirán para llevar a cabo con la certificación de software que establecerá que las pruebas se realizaron adecuadamente y en el momento oportuno.

El objetivo general de este documento es establecer la cronología y condiciones para la aplicación de pruebas, de esta forma obtener, un sistema que pueda ser completo y que cumpla con las expectativas del cliente final. Además de esto poner el mismo en operación mostrando todas sus funcionalidades. Ya que va dirigido al cliente final y este espera que el sistema esté funcionando a la perfección con todos los requisitos que el solicito al comienzo del desarrollo del sistema.

Definición de niveles de pruebas

Las pruebas que se realizaran durante el desarrollo y ejecución del sistema de información serán aquellas que ayuden a encontrar problemas y solucionarlos en tiempo oportuno, Además de esto ayudan a comprobar si los resultados obtenidos son iguales que los resultados esperados y así garantizar de forma segura y eficaz que el sistema está libre de errores y defectos. Las pruebas de software que se usaran para el sistema de información **EFFECTIVE RECORD** han sido previamente investigadas y probadas en el mismo sistema.

Software de testeo

¿Qué es un SW de testeo?

Los probadores de software (también conocidos como testers, su denominación en inglés) planifican y llevan a cabo pruebas de software de los ordenadores para comprobar si funcionan correctamente. Identifican el riesgo de sufrir errores de un software, detectan errores y los comunican. Evalúan el funcionamiento general del software y sugieren formas de mejorarlo.

¿Qué es JMeter?

La aplicación **Apache JMeter**™ es un software de código abierto, una aplicación Java 100% pura diseñada para cargar el comportamiento funcional de prueba y medir el rendimiento. Originalmente fue diseñado para probar aplicaciones web, pero desde entonces se ha expandido a otras funciones de prueba.

¿Para que se usa JMeter?

Apache JMeter se puede utilizar para probar el rendimiento tanto en recursos estáticos como dinámicos, aplicaciones web dinámicas.

Se puede utilizar para simular una carga pesada en un servidor, grupo de servidores, red u objeto para probar su fuerza o para analizar el rendimiento general bajo diferentes tipos de carga.

Las características de Apache JMeter incluyen:

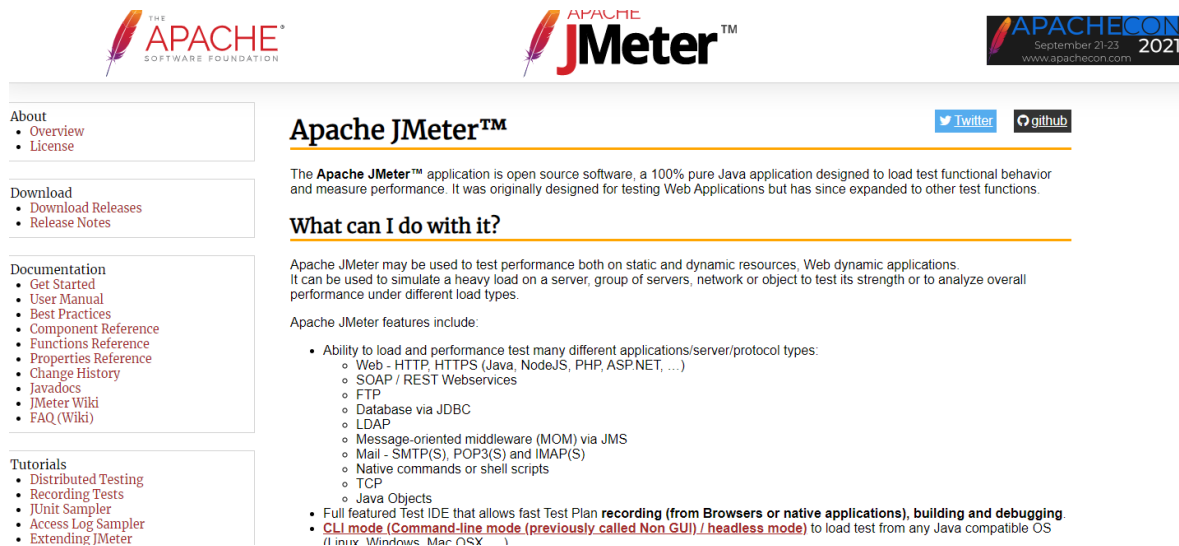
Capacidad para cargar y probar el rendimiento de muchas aplicaciones / servidores / tipos de protocolos diferentes:

- Web: HTTP, HTTPS (Java, NodeJS, PHP, ASP.NET,...)
- Servicios web SOAP / REST
- FTP
- Base de datos a través de JDBC
- LDAP
- Middleware orientado a mensajes (MOM) a través de JMS
- Correo: SMTP (S), POP3 (S) e IMAP (S)
- Comandos nativos o scripts de shell
- TCP
- Objetos Java

Instalación y configuración del software

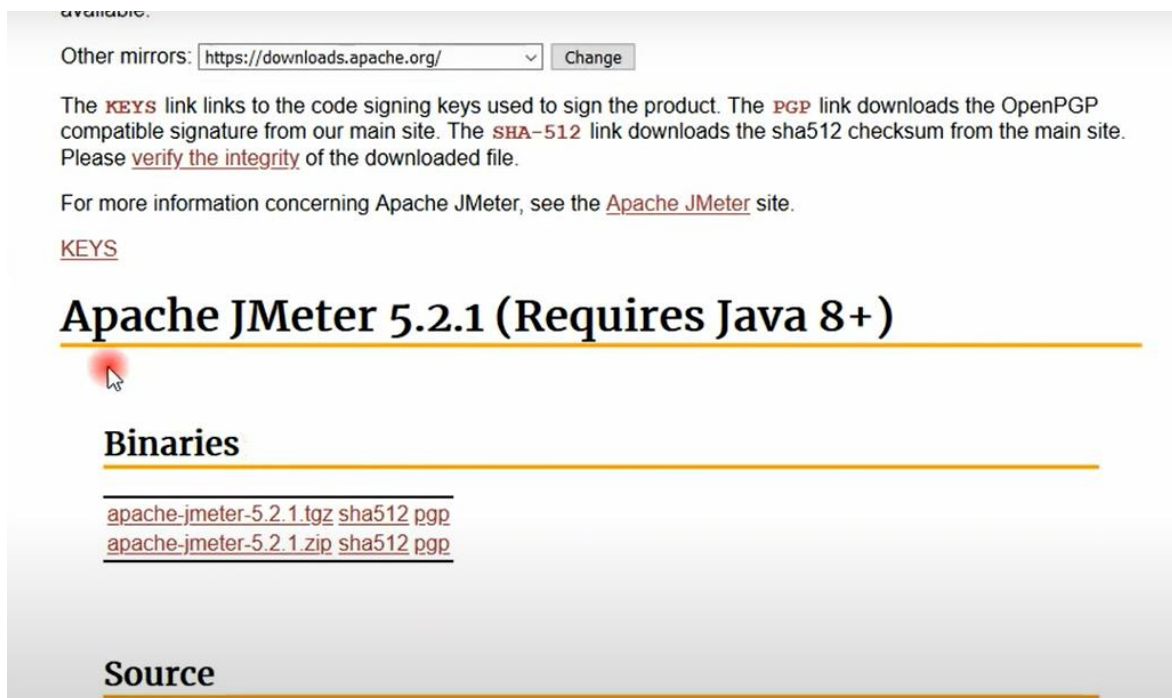
Paso 1: Descargar JMeter

Para descargar JMeter nos vamos a la página oficial de Jmeter.Apache.org. Nos aparecerá la interfaz que se mostrará en la siguiente imagen.



The screenshot shows the Apache JMeter website. At the top, there are logos for 'THE APACHE SOFTWARE FOUNDATION', 'Apache JMeter™', and 'APACHECON September 21-23 2021 www.apachecon.com'. Below the logos, there is a navigation menu on the left with links for 'About' (Overview, License), 'Download' (Download Releases, Release Notes), 'Documentation' (Get Started, User Manual, Best Practices, Component Reference, Functions Reference, Properties Reference, Change History, Javadocs, JMeter Wiki, FAQ (Wiki)), and 'Tutorials' (Distributed Testing, Recording Tests, JUnit Sampler, Access Log Sampler, Extending JMeter). The main content area features the title 'Apache JMeter™' with social media links for Twitter and GitHub. Below the title, a paragraph states: 'The Apache JMeter™ application is open source software, a 100% pure Java application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions.' This is followed by a section titled 'What can I do with it?' which explains that JMeter can be used to test performance on static and dynamic resources, Web dynamic applications, and to simulate heavy load on a server, group of servers, network or object to test its strength or to analyze overall performance under different load types. A list of features follows: 'Apache JMeter features include: Ability to load and performance test many different applications/server/protocol types: Web - HTTP, HTTPS (Java, NodeJS, PHP, ASP.NET, ...), SOAP / REST Webservices, FTP, Database via JDBC, LDAP, Message-oriented middleware (MOM) via JMS, Mail - SMTP(S), POP3(S) and IMAP(S), Native commands or shell scripts, TCP, Java Objects; Full featured Test IDE that allows fast Test Plan recording (from Browsers or native applications), building and debugging; CLI mode (Command-line mode (previously called Non GUI) / headless mode) to load test from any Java compatible OS (Linux, Windows, Mac OS X, ...)'.

Seleccionamos Download Releases, y nos aparecerá la siguiente interfaz que nos muestra la última versión disponible de JMeter. Debemos tener en cuenta que para tener esta versión de JMeter debemos tener instalado la última versión de Java o la versión que se muestra en la imagen.

A screenshot of the Apache JMeter download page. At the top, there's a section for 'Other mirrors' with a dropdown menu showing 'https://downloads.apache.org/' and a 'Change' button. Below this, a paragraph explains the links for keys, PGP signatures, and SHA-512 checksums, advising users to verify the integrity of the downloaded file. It also points to the Apache JMeter site for more information. A red mouse cursor is hovering over the 'KEYS' link. The main heading is 'Apache JMeter 5.2.1 (Requires Java 8+)' in a large, bold, black font, underlined with a thick orange line. Below this, the 'Binaries' section is also underlined with a thick orange line. It contains two lines of links: 'apache-jmeter-5.2.1.tgz sha512 pgp' and 'apache-jmeter-5.2.1.zip sha512 pgp', both underlined. The 'Source' section is also underlined with a thick orange line.

Other mirrors: <https://downloads.apache.org/> [Change](#)

The **KEYS** link links to the code signing keys used to sign the product. The **PGP** link downloads the OpenPGP compatible signature from our main site. The **SHA-512** link downloads the sha512 checksum from the main site. Please [verify the integrity](#) of the downloaded file.

For more information concerning Apache JMeter, see the [Apache JMeter](#) site.

[KEYS](#)

Apache JMeter 5.2.1 (Requires Java 8+)

Binaries

[apache-jmeter-5.2.1.tgz sha512 pgp](#)
[apache-jmeter-5.2.1.zip sha512 pgp](#)

Source

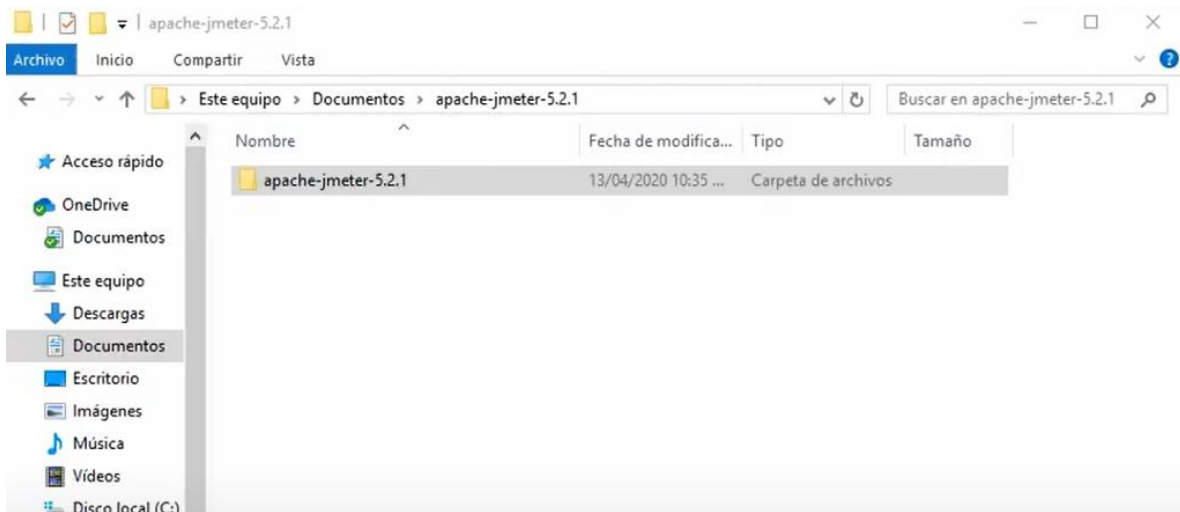
Después, vamos a la sección de “Binaries” y damos clic en la segunda opción que nos aparece allí para iniciar la descarga del .zip

A close-up screenshot of the 'Binaries' section of the Apache JMeter download page. The section is underlined with a thick orange line. It contains two lines of links: 'apache-jmeter-5.2.1.tgz sha512 pgp' and 'apache-jmeter-5.2.1.zip sha512 pgp', both underlined.

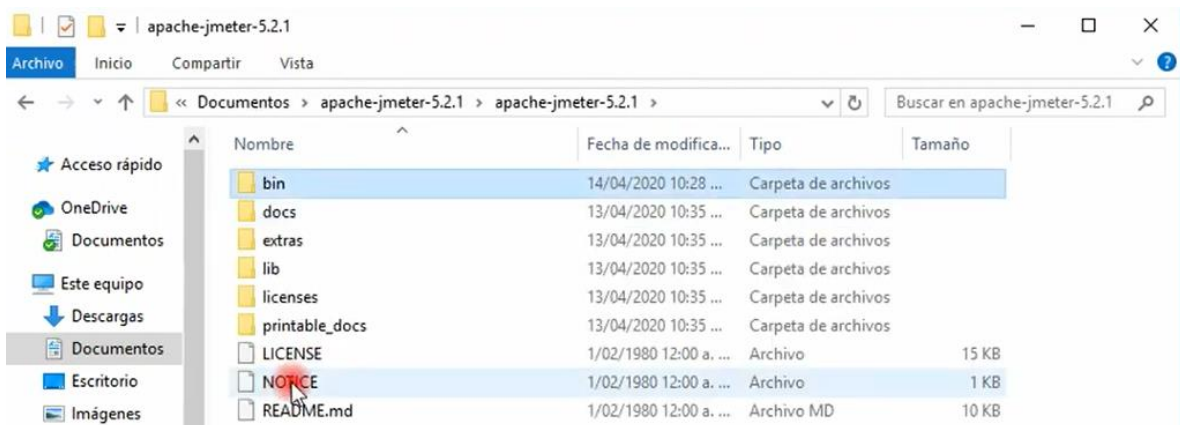
Binaries

[apache-jmeter-5.2.1.tgz sha512 pgp](#)
[apache-jmeter-5.2.1.zip sha512 pgp](#)

Una vez descargado el Zip lo descomprimos debería quedar una carpeta como se muestra en la siguiente imagen.



Después procedemos a abrir la carpeta, nos aparecerán estas carpetas.



Luego abrimos la carpeta "bin", que debe contener el siguiente archivo

« Documentos » apache-jmeter-5.2.1 » apache-jmeter-5.2.1 » bin »

Buscar en bin

Nombre	Fecha de modifica...	Tipo	Tamaño
examples	13/04/2020 10:34 ...	Carpeta de archivos	
report-template	13/04/2020 10:34 ...	Carpeta de archivos	
templates	13/04/2020 10:34 ...	Carpeta de archivos	
ApacheMeter.jar	1/02/1980 12:00 a. ...	Executable Jar File	14 KB
BeanShellAssertion.bshrc	1/02/1980 12:00 a. ...	Archivo BSHRC	2 KB
BeanShellFunction.bshrc	1/02/1980 12:00 a. ...	Archivo BSHRC	3 KB
BeanShellListeners.bshrc	1/02/1980 12:00 a. ...	Archivo BSHRC	2 KB
BeanShellSampler.bshrc	1/02/1980 12:00 a. ...	Archivo BSHRC	3 KB
create-rmi-keystore.bat	1/02/1980 12:00 a. ...	Archivo por lotes ...	2 KB
create-rmi-keystore.sh	1/02/1980 12:00 a. ...	Archivo SH	2 KB
hc.parameters	1/02/1980 12:00 a. ...	Archivo PARAMET...	2 KB
heapdump.cmd	1/02/1980 12:00 a. ...	Script de comand...	2 KB
heapdump.sh	1/02/1980 12:00 a. ...	Archivo SH	2 KB
jaas.conf	1/02/1980 12:00 a. ...	Archivo CONF	2 KB
jmeter	1/02/1980 12:00 a. ...	Archivo	9 KB
jmeter.bat	1/02/1980 12:00 a. ...	Archivo por lotes ...	9 KB
jmeter.log	14/04/2020 10:28 ...	Documento de tex...	4 KB
jmeter.properties	1/02/1980 12:00 a. ...	Archivo PROPERTI...	56 KB
jmeter.sh	1/02/1980 12:00 a. ...	Archivo SH	4 KB
jmeter-n.cmd	1/02/1980 12:00 a. ...	Script de comand...	2 KB

« Documentos » apache-jmeter-5.2.1 » apache-jmeter-5.2.1 » bin »

Buscar

Nombre	Fecha de modifica...	Tipo	Ta
jmeter-server.bat	1/02/1980 12:00 a. ...	Archivo por lotes	
jmeter-t.cmd	1/02/1980 12:00 a. ...	Script de comand...	
jmeterw.cmd	1/02/1980 12:00 a. ...	Script de comand...	
krb5.conf	1/02/1980 12:00 a. ...	Archivo CONF	
log4j2.xml	1/02/1980 12:00 a. ...	Documento XML	
mirror-server	1/02/1980 12:00 a. ...	Archivo	
mirror-server.cmd	1/02/1980 12:00 a. ...	Script de comand...	
mirror-server.sh	1/02/1980 12:00 a. ...	Archivo SH	
reportgenerator.properties	1/02/1980 12:00 a. ...	Archivo PROPERTI...	
saveservice.properties	1/02/1980 12:00 a. ...	Archivo PROPERTI...	
shutdown.cmd	1/02/1980 12:00 a. ...	Script de comand...	
shutdown.sh	1/02/1980 12:00 a. ...	Archivo SH	
stoptest.cmd	1/02/1980 12:00 a. ...	Script de comand...	
stoptest.sh	1/02/1980 12:00 a. ...	Archivo SH	
system.properties	1/02/1980 12:00 a. ...	Archivo PROPERTI...	
threaddump.cmd	1/02/1980 12:00 a. ...	Script de comand...	
threaddump.sh	1/02/1980 12:00 a. ...	Archivo SH	
upgrade.properties	1/02/1980 12:00 a. ...	Archivo PROPERTI...	
user.properties	1/02/1980 12:00 a. ...	Archivo PROPERTI...	
utility.groovy	1/02/1980 12:00 a. ...	Archivo GROOVY	

Paso 2: Descargar o verificar versión de JDK

En este paso vamos a verificar nuestra versión de Java o si esta no está instalada procedemos a descargar el JDK en la página de Oracle.

Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

Important Oracle JDK License Update


The Oracle JDK License has changed for releases starting April 16, 2019.

The new [Oracle Technology Network License Agreement for Oracle Java SE](#) is substantially different from prior Oracle JDK licenses. The new license permits certain uses, such as personal use and development use, at no cost -- but other uses authorized under prior Oracle JDK licenses may no longer be available. Please review the terms carefully before downloading and using this product. An FAQ is available [here](#).

Commercial license and support is available with a low cost [Java SE Subscription](#).

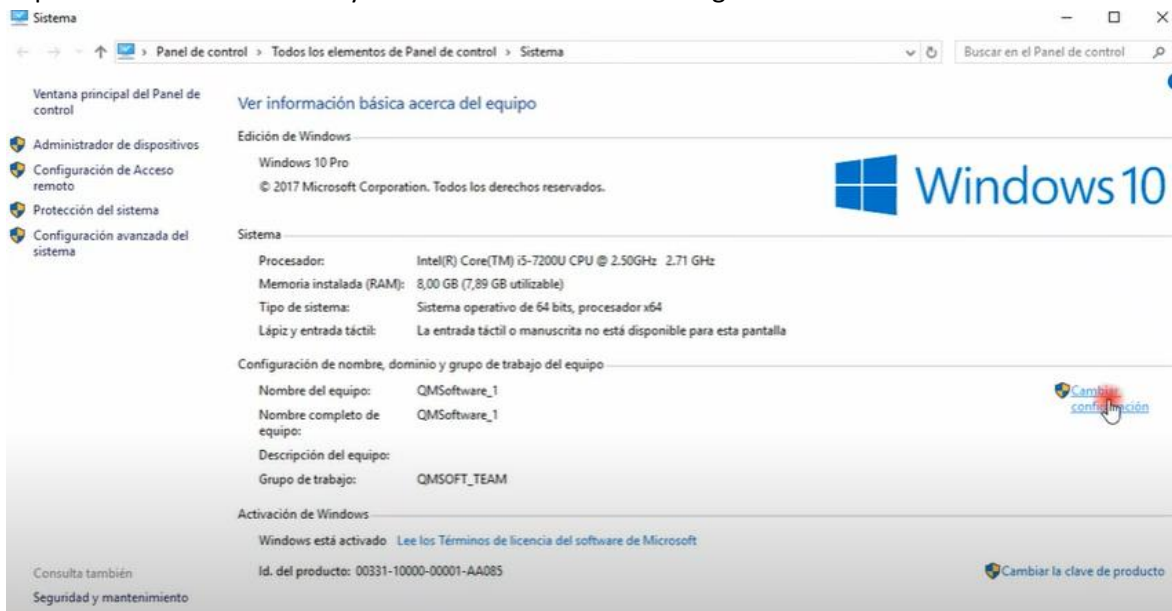
Oracle also provides the latest OpenJDK release under the open source [GPL License](#) at [jdk.java.net](#).

Nos desplazamos a la parte inferior de la página y seleccionamos la opción que sea compatible con el sistema operativo de nuestro ordenador en este caso es para Windows 10

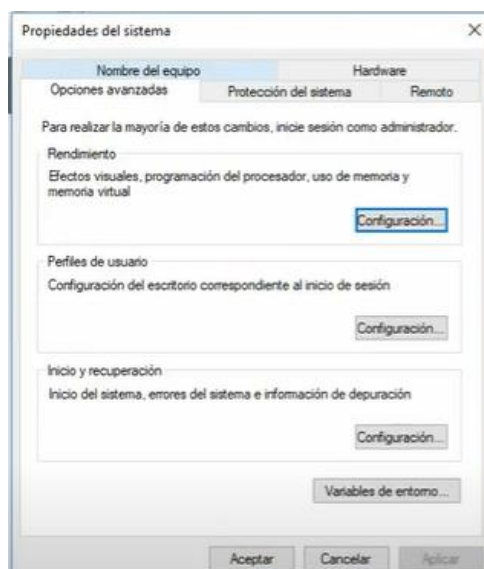
Windows x64	211.54 MB	 jdk-8u251-windows-x64.exe
-------------	-----------	---

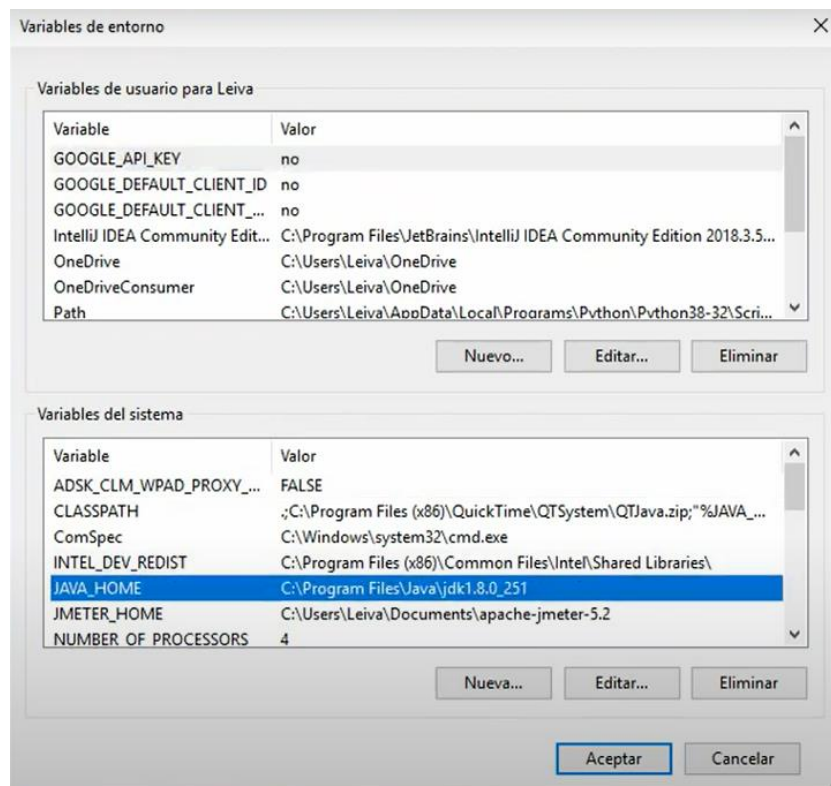
Paso 3: Configuración del entorno

Para comenzar a configurar el entorno, nos dirigimos las propiedades del sistema para esto vamos al panel de control> sistema y seleccionamos cambiar configuración



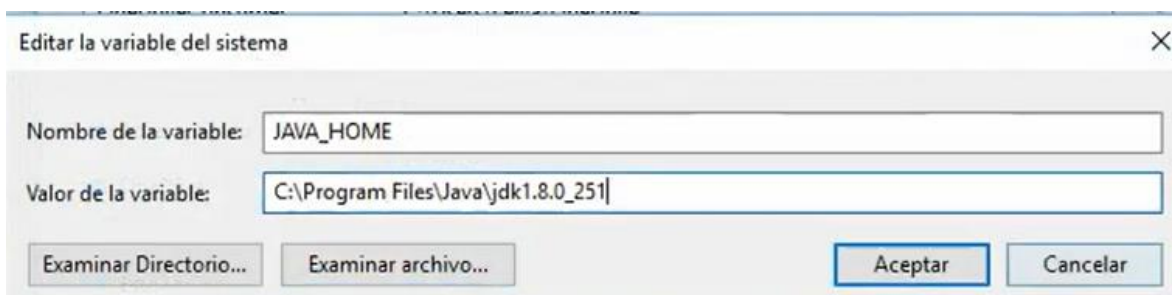
Y seguidamente seleccionamos en variables de entorno



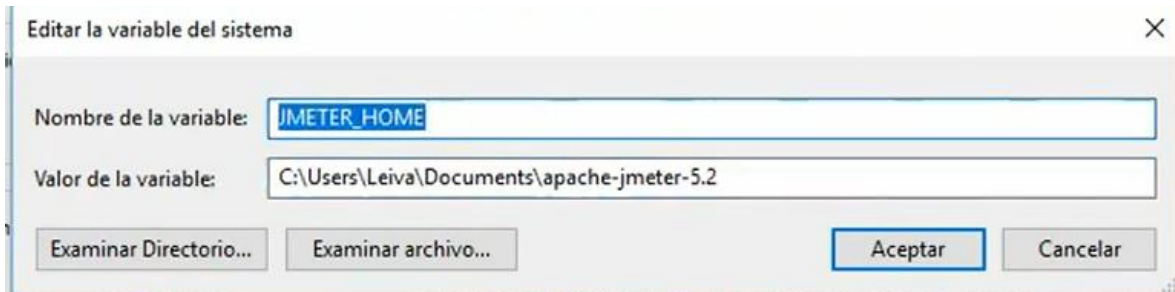


Luego damos clic en “opciones avanzadas” y verificamos si existe la variable JAVA_HOME

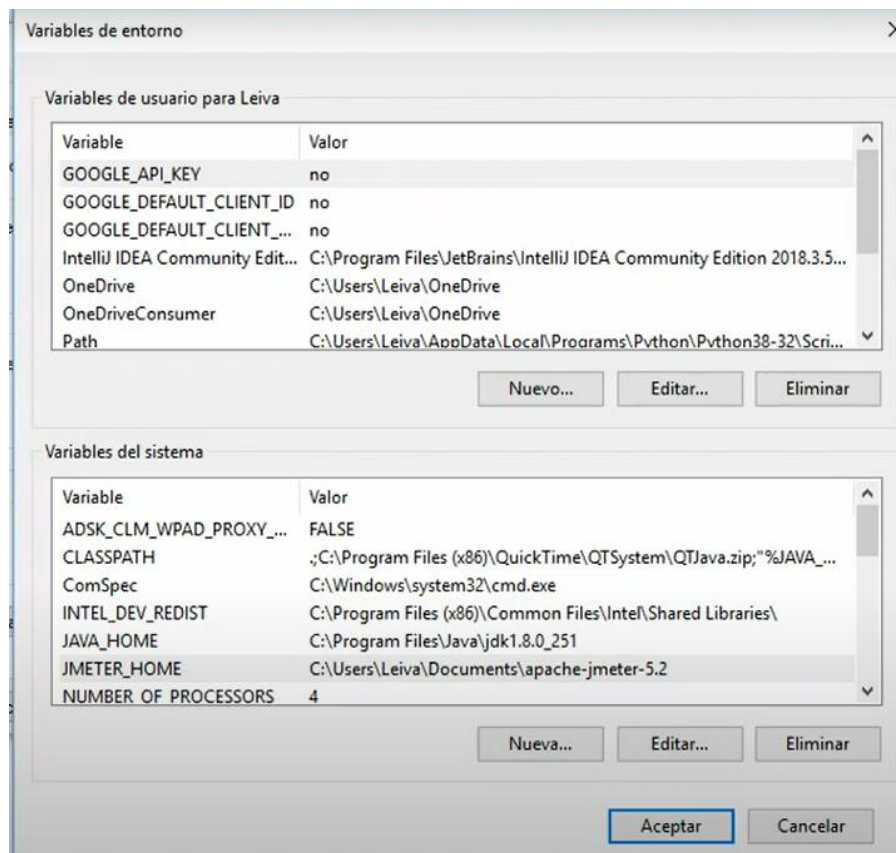
En caso de que no esté la variable, la creamos seleccionando en Nueva llamándola JAVA_HOME y ponemos la ruta donde se encuentre instalado nuestro Java.

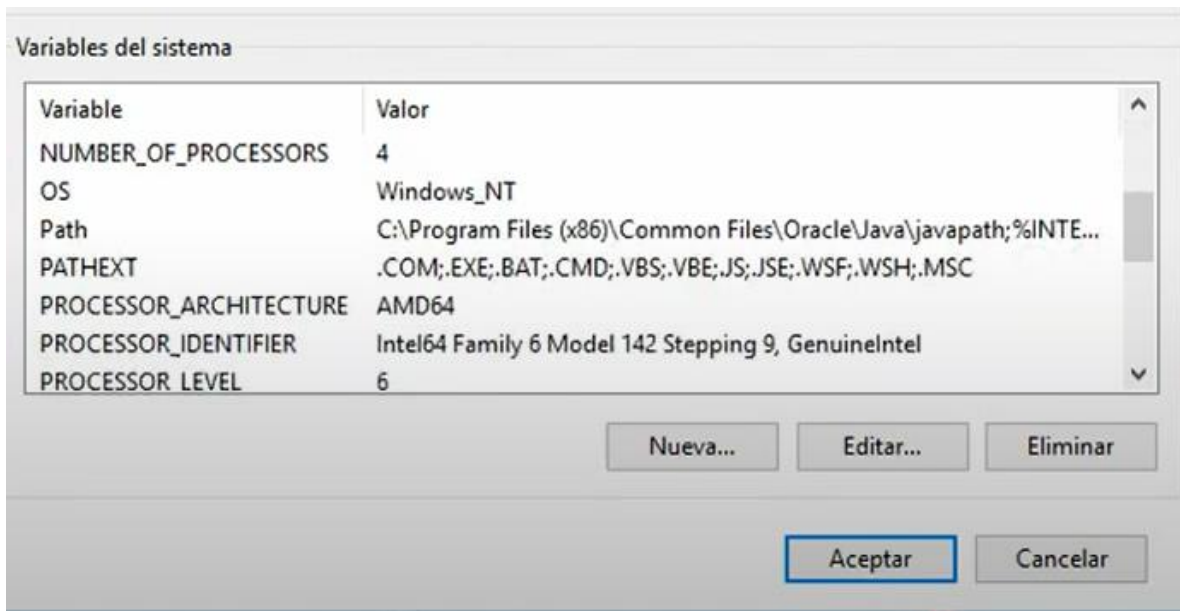


Después crearemos otra variable llamado JMETER_HOME y la ruta será donde se encuentran los archivos de trabajo de JMeter que estemos trabajando

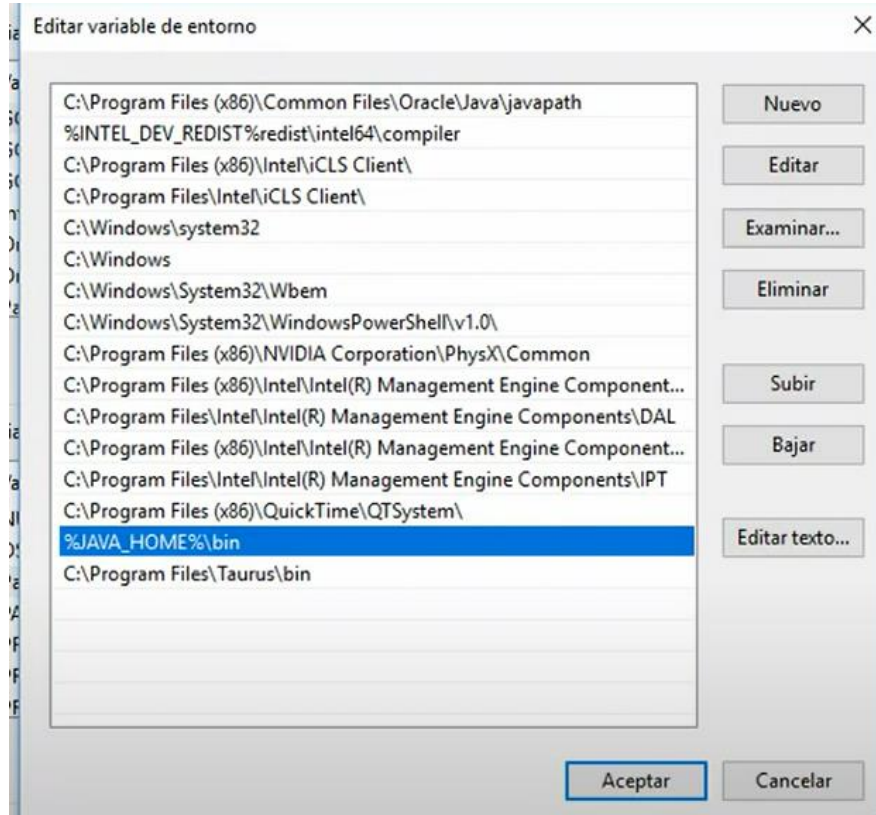


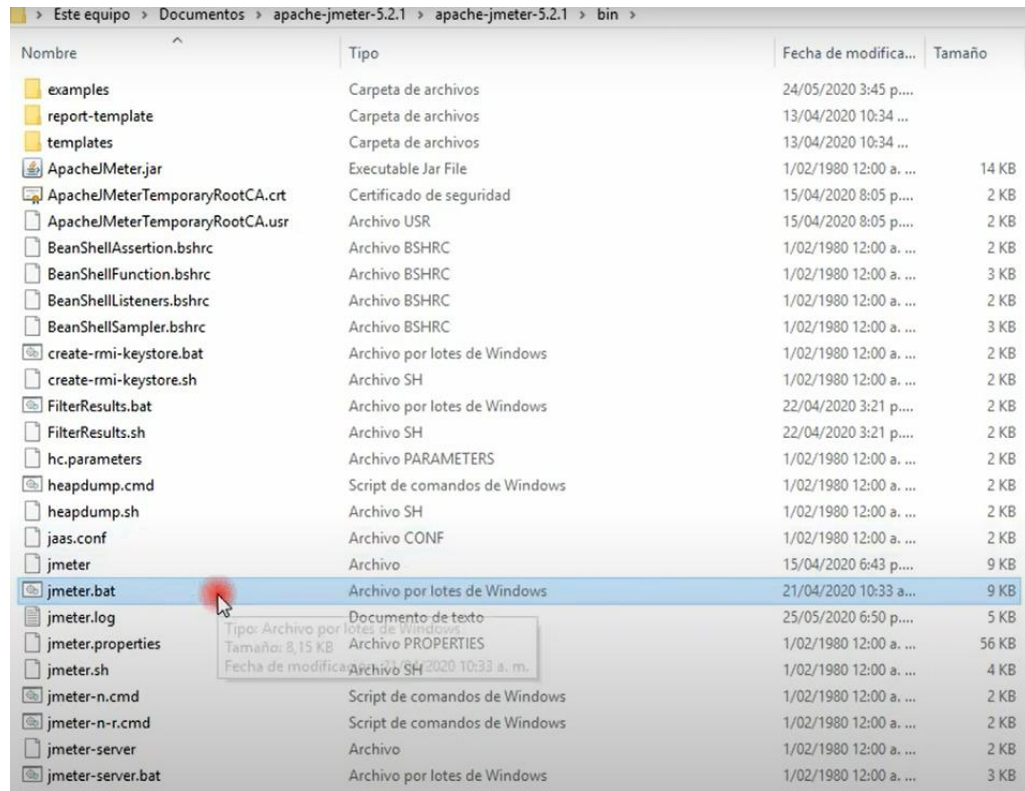
Después buscamos la variable "Path"





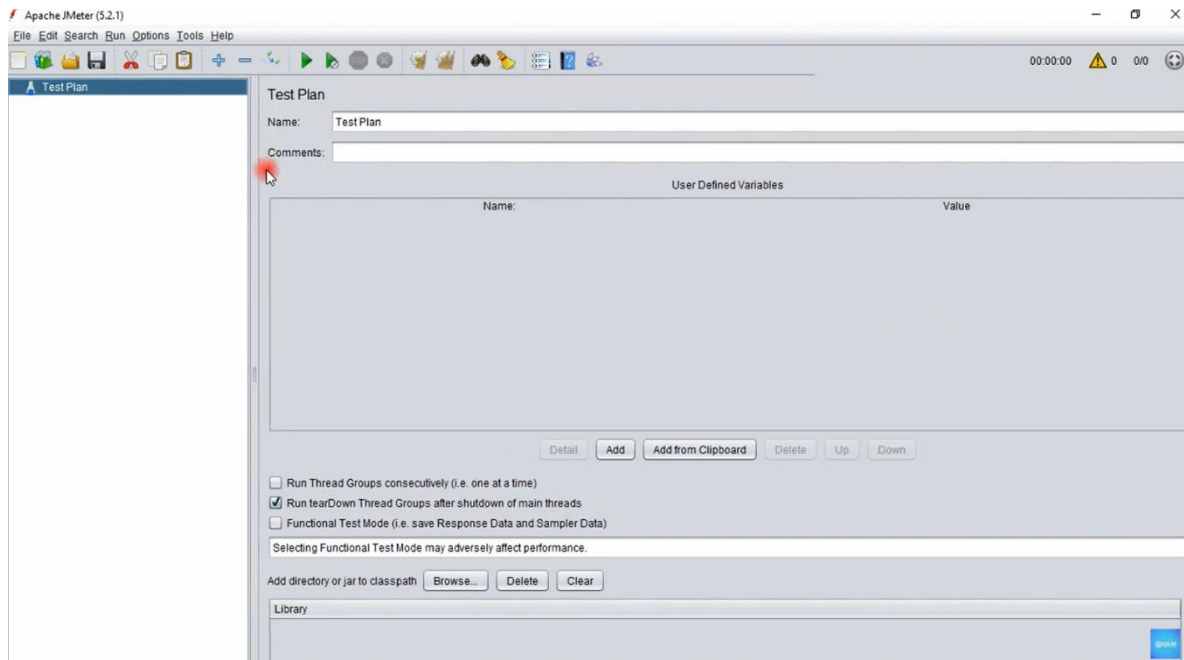
Y aquí nos debe aparecer el JAVA_HOME como se muestra en la siguiente imagen. Para finalizar, damos clic en aceptar y ya estará configurada la máquina virtual lista para ejecutar la herramienta JMeter.





Ahora nos dirigimos a la carpeta bin de JMeter y seleccionamos el archivo jmeter.bat

Esta será la interfaz donde empezaremos a diseñar nuestro plan de pruebas.



Pruebas realizadas con el software

Pruebas no funcionales

Prueba de carga registrar usuarios.

La prueba de carga se refiere a la capacidad que tiene un software, para atender a un conjunto de usuarios al mismo tiempo, Por ello, las actividades de esta etapa tienen relación con comprobar, de manera anticipada, el funcionamiento que tendrá el software cuando esté en plena operación.

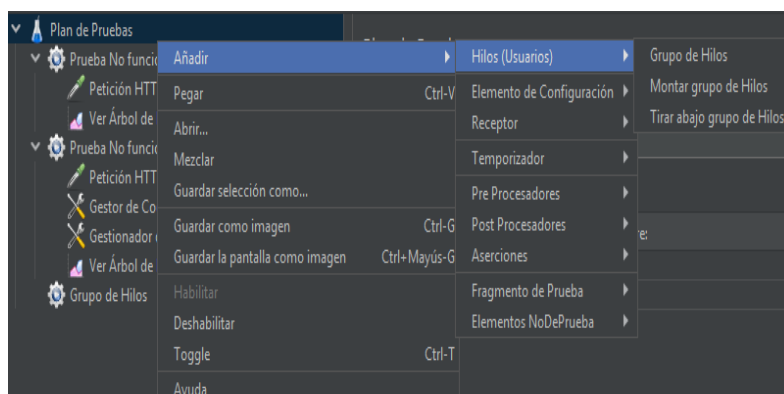
Las pruebas en este caso consisten en simular una carga de trabajo similar y superior a la que tendrá cuando el software esté funcionando, con el fin de detectar si los (programas y aplicaciones) cumple con los requerimientos de muchos usuarios simultáneos.

Una de las pruebas que realizo **EFFECTIVE RECORD** en el entorno de testeo del software **JMETER** es la prueba de carga la cual consiste en configurar el entono de prueba para que cien usuarios hagan un petición en un segundo sin que el software se colapse o presente fallas.

La prueba se realiza de la siguiente manera.

Paso 1

En el primer paso se creara el entorno de **Plan de Pruebas** pruebas dando clic derecho en “**Plan de pruebas**”, después de esto se desplegara un sub menú en el cual vamos a seleccionar “**Hilos (usuarios)**”, una se realice esto se desplegara otro sub menú en el cual vamos a clicar “**Grupo de hilos**” cuando realicemos esto se nos creara el grupo de hilos de prueba donde asignaremos un nombre.



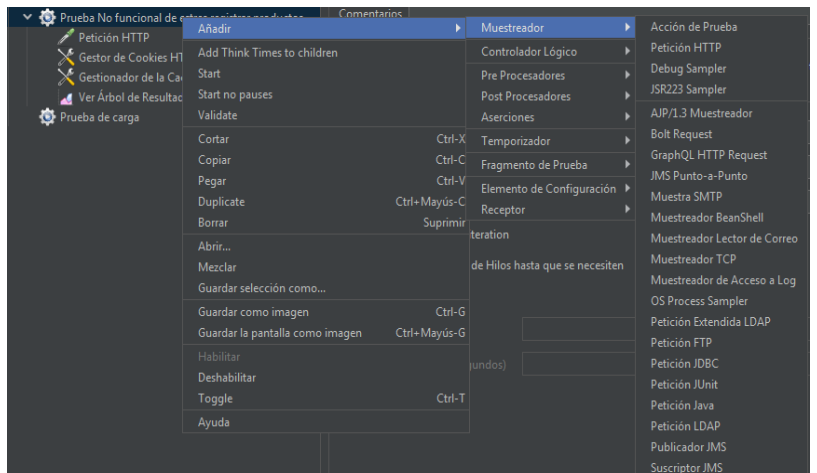
Paso 2

Cuando ya se cree el grupo de hilos y le hallamos asignado un nombre, digitaremos el número de hilos (usuarios) que van a realizar la petición y el tiempo en el que van a realizaran la petición.

The screenshot shows the 'Grupo de Hilos' configuration window. The 'Nombre' field is set to 'Prueba No funcional de estres registrar productos'. The 'Acción a tomar después de un error de Muestreador' section has 'Continuar' selected. The 'Propiedades de Hilo' section shows 'Número de Hilos' set to 100, 'Periodo de Subida (en segundos)' set to 1, and 'Contador del bucle' set to 1 with 'Sin fin' checked. There are also checkboxes for 'Same user on each iteration', 'Retrasar la creación de Hilos hasta que se necesiten', and 'Planificador'.

Paso 3

Una vez realizado el paso número tres crearemos lo que es la petición HTTP de la siguiente manera. Daremos clic derecho en el grupo de hilos y se desplegara un sub menú en el cual vamos a clicar en **“Añadir”**, después nos saldrá otro submenú en el cual vamos a clicar **“Muestreador”**, lo cual nos va a desplegar otro submenú en donde ya podremos seleccionar **“petición HTTP”**



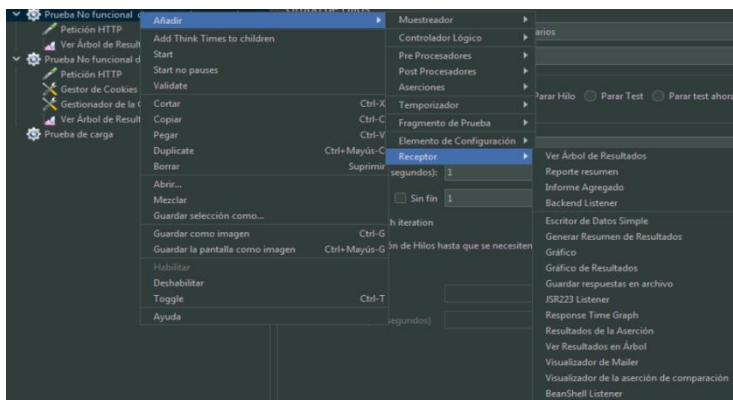
Paso 4

Una vez se nos cree el entorno grafico de la petición HTTP, lo configuraremos de la siguiente forma, primero donde dice **Nombre de servidor IP**, digitaremos **“localhost”**, y segundo donde dice puerto pondremos **“8080”**, esto varia depende el servidor web que usemos, una vez realicemos esto pondremos la ruta del proyecto donde especifiquemos el archivo al que vamos a realizar él es test, esto va ubicado en donde dice **“ruta”**.

The screenshot shows the 'Petición HTTP' configuration window. The 'Nombre' field is set to 'Petición HTTP'. The 'Basic' tab is selected. The 'Servidor Web' section shows 'Protocolo' set to 'GET' and 'Nombre de Servidor o IP' set to 'localhost'. The 'Puerto' field is set to '8080'. The 'Ruta' field is set to 'Effective/vista/registroUsuarios.jsp'. The 'Codificación del contenido' field is empty. The 'Parameters' section is visible at the bottom.

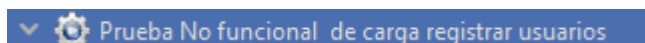
Paso 5

Cuando ya hallamos realizado el cuarto paso crearemos el árbol de resultados que como su nombre lo indica nos va a mostrar los resultados en función de la ejecución del test. Para crear este árbol de resultados daremos clic derecho nuevamente en el grupo de hilos, se desplegara un submenú donde daremos **“Añadir”**, después se desplegara otro submenú donde daremos clic en **“receptor”**, y por ultimo daremos clic en **“ver árbol de resultados”**.

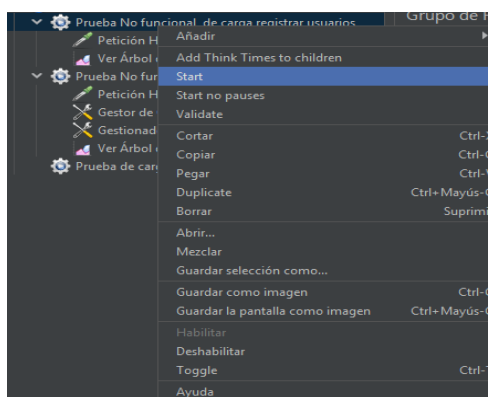


Paso 6

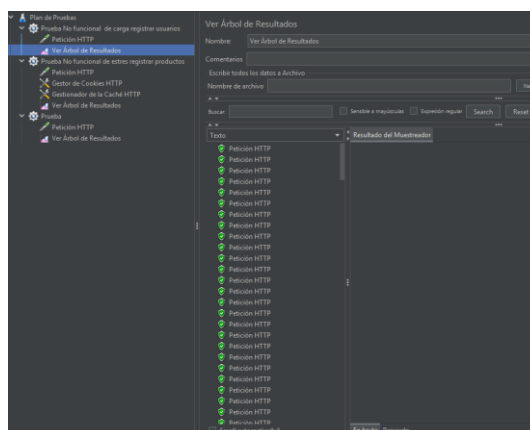
Una vez realicemos los cinco pasos, ejecutaremos el test que hemos configurado de esta forma. Primero daremos clic derecho en el grupo de hilos



Donde nos saldrá un submenú en el cual vamos a clicar en **“Star”** y se nos comenzara a ejecutar la prueba previamente configurada.



Una vez realicemos este último paso podremos observar como el test se logró ejecutar exitosamente sin presentar ningún clase de problemas.



Pruebas de estrés registrar productos

La prueba de *estrés*. Evalúa y pone a prueba la robustez y la confiabilidad del *software* sometiéndolo a condiciones de uso extremas. Entre estas condiciones se incluyen el envío excesivo de peticiones y la ejecución en condiciones de hardware limitadas. El objetivo es saturar el programa hasta un punto de quiebre donde aparezcan *bugs* (defectos) potencialmente peligrosos.

Una de las pruebas que realizo **EFFECTIVE RECORD** en el entorno de testeo del software **JMETER** es la prueba de estrés la cual consiste en configurar el entono de prueba para que cien usuarios hagan un petición de registrar productos en un segundo sin que el software se colapse o presente fallas, el objetivo es buscar un punto o un cuello botella en el cual el software presente una serie de fallas o problemas.

Paso 1

Comenzando con esta prueba crearemos un grupo de hilos de la siguiente manera.

Dándole click derecho a “**Plan de pruebas**” don se desplegara un apartado donde seleccionaremos “**Añadir**” en el cual no abrirá otro apartado para finalmente escoger la opción “**Hilos (Usuarios)**” y grupo de hilos.



Paso 2

Ahora ya creado el hilo le daremos un nombre y después seleccionaremos el número de hilos (Usuarios) y el tiempo (en segundos) con los que se realizara esta prueba tal como se muestra a continuación.

Grupo de Hilos

Nombre:

Prueba No funcional de estres registrar productos

Comentarios

Acción a tomar después de un error de Muestreador

☒ Continuar

☐ Comenzar siguiente iteración

☐ Parar Hilo

☐ Parar Test

☐ Parar test ahora

Propiedades de Hilo

Número de Hilos

100

Periodo de Subida (en segundos):

1

Contador del bucle:

☐ Sin fin

1

☒ Same user on each iteration

☐ Retrasar la creación de Hilos hasta que se necesiten

☐ Planificador

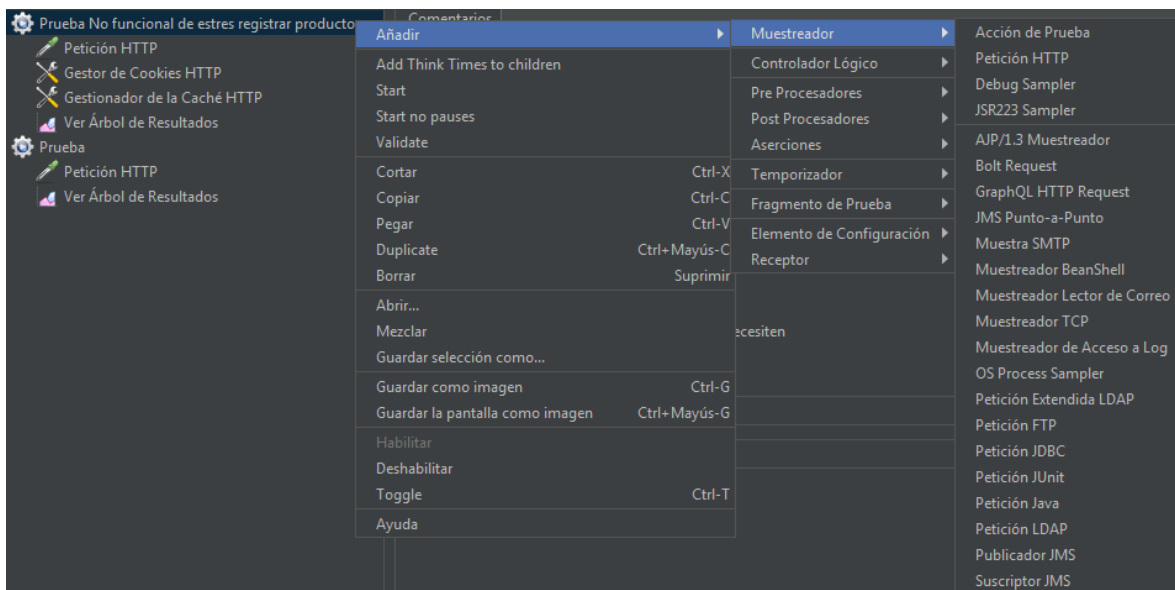
Duración (segundos)

Retardo de arranque (segundos)

Paso 3

Ya teniendo creado el grupo de hilo y asignado su nombre crearemos la petición HTTP de la siguiente manera.

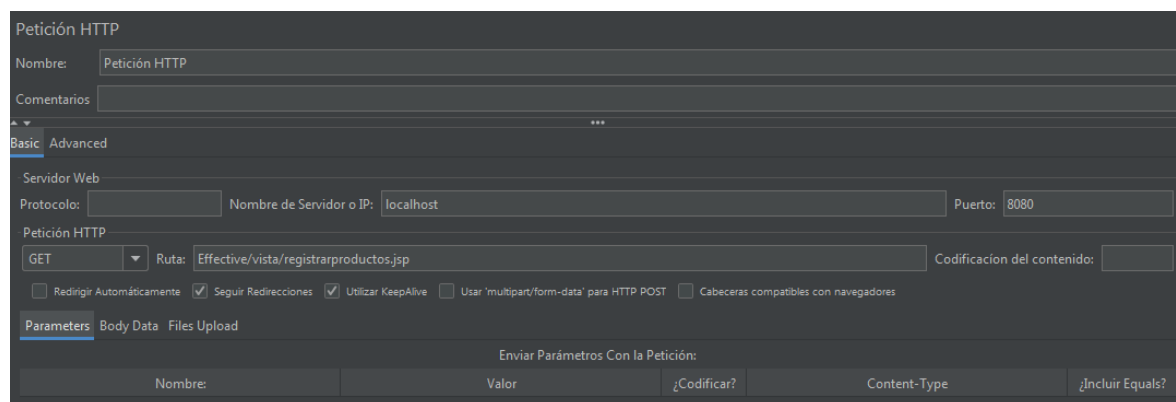
Oprimiremos click derecho en el grupo de hilos donde seleccionaremos la opción de **“Añadir”**, luego seleccionaremos **“Muestreador”** para finalmente seleccionar el archivo de petición HTTP.



Paso 4

Ahora configuraremos la petición con los siguientes datos.

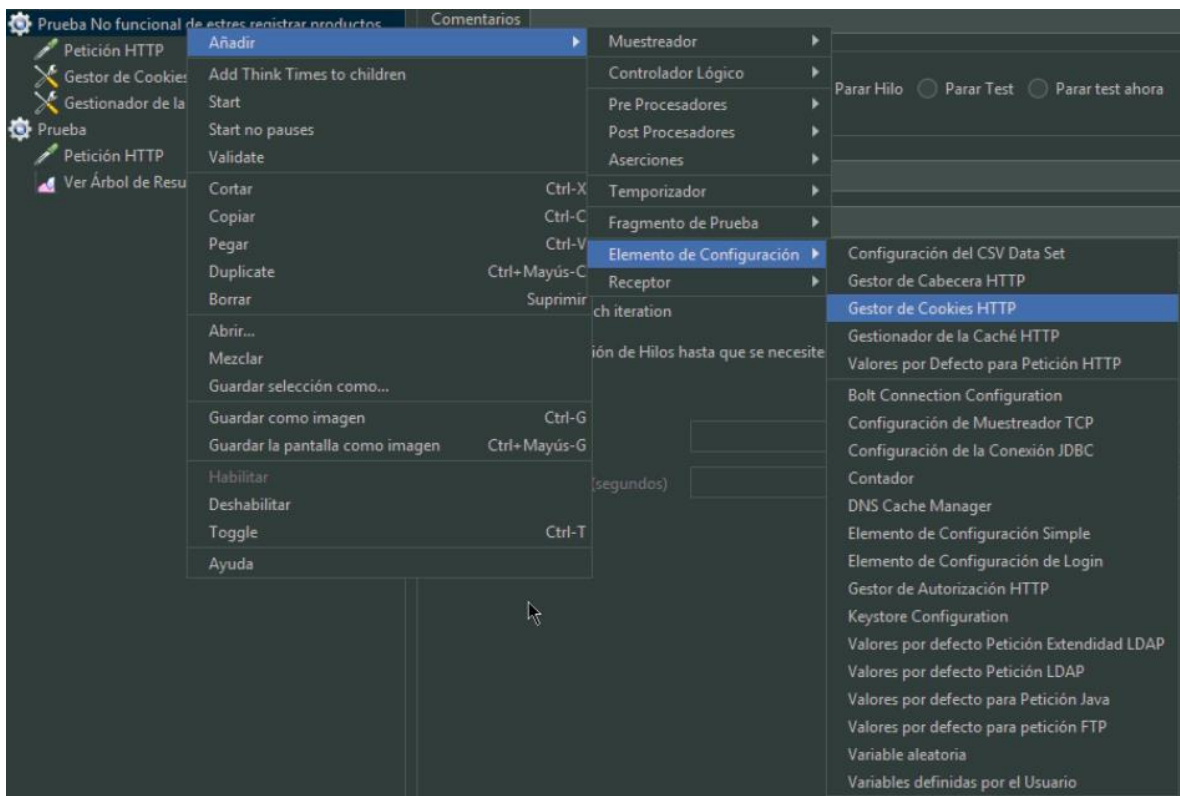
En el apartado de **“Nombre de servidor o IP”** escribiremos **“localhost”**, en el apartado que dice **“Puerto”** escribiremos **“8080”** y por último en el apartado de **“Ruta”** seleccionaremos el archivo del proyecto al cual se le quiere realizar esta prueba.



Paso 5

Ahora crearemos una herramienta que se llama **“Gestor de cookies HTTP”** a continuación se mostrará con crear esta herramienta.

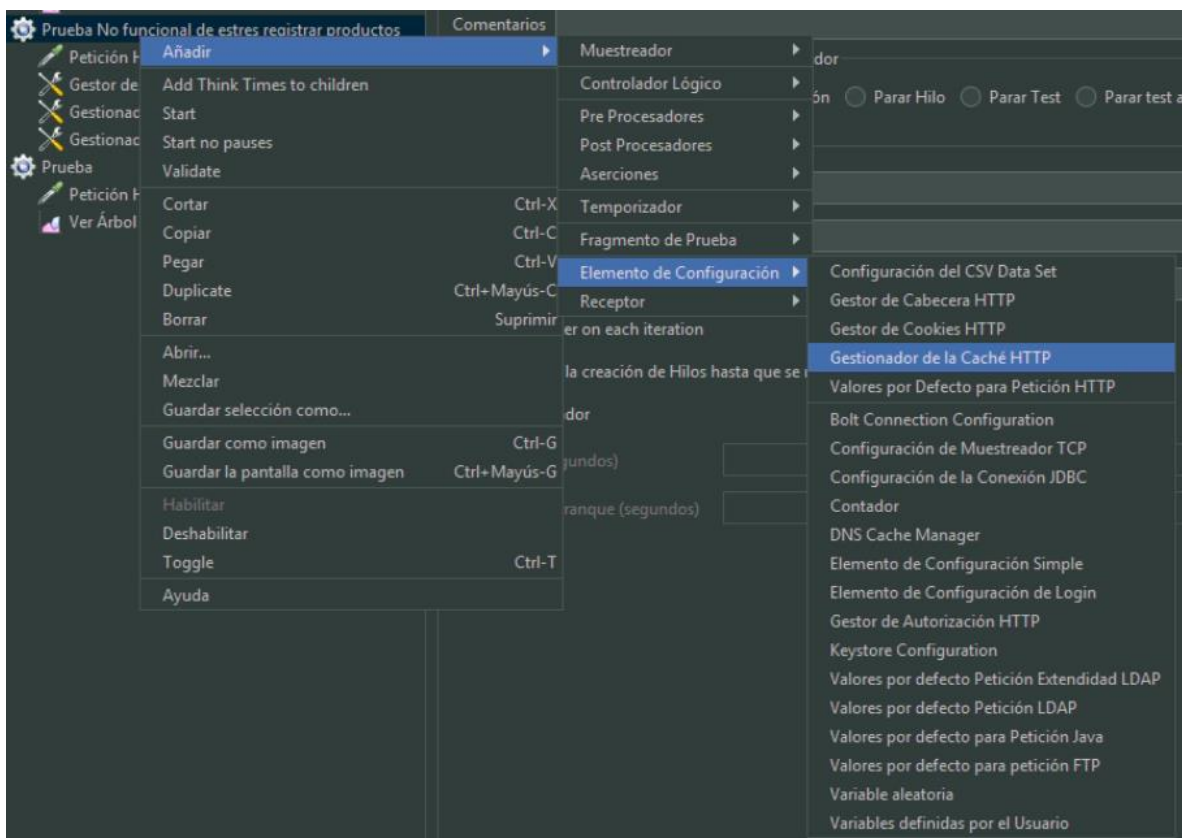
Primero le daremos click derecho en el grupo de hilos donde seleccionaremos en **“Añadir”**, después seleccionaremos la opción de **“Elemento de configuración”** para finalmente seleccionar el **“Gestor de cookies HTTP”**.



Paso 6

A continuación crearemos el “**Gestionador de la Caché HTTP**”.

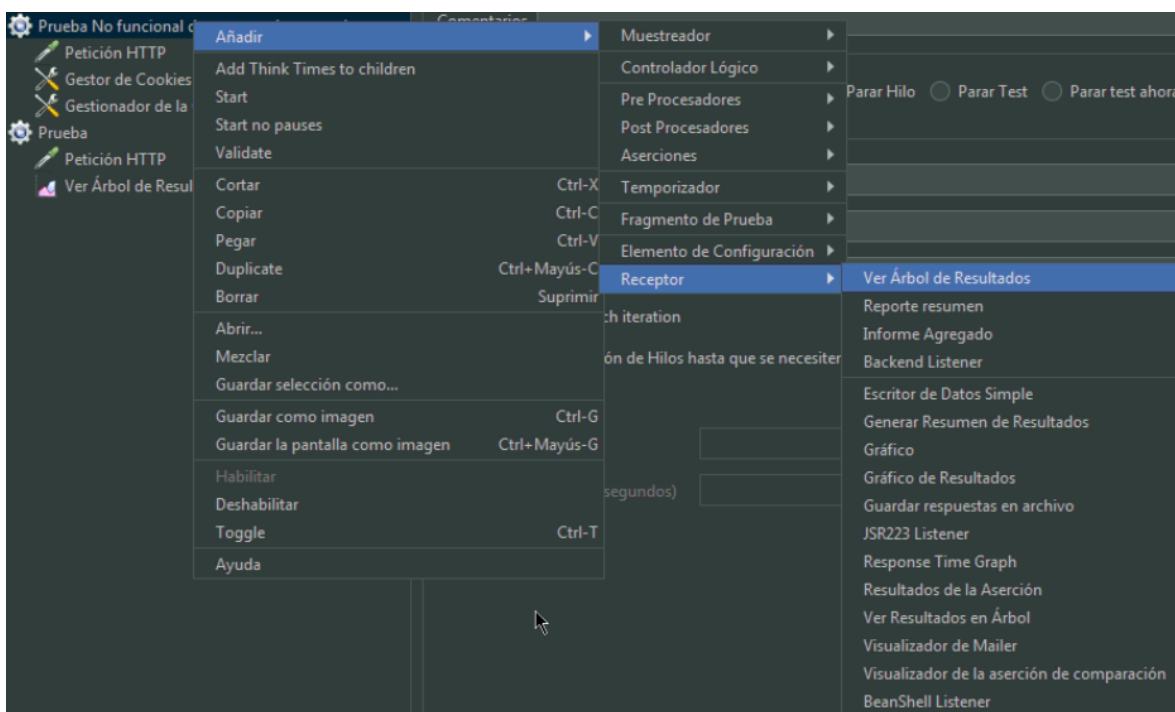
Para esto le daremos click derecho en el grupo de hilos para posteriormente seleccionar “Añadir”, luego seleccionaremos “**Elemento de Configuración**”, finalmente seleccionaremos la opción del “**Gestionador de la Cache HTTP**”.



Paso 7

En este paso crearemos el Árbol de resultados donde se evidenciara los resultados obtenidos tras realizar la prueba.

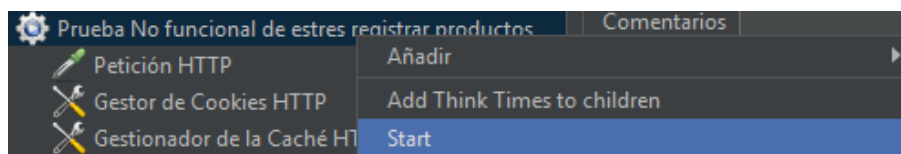
Para esto daremos click derecho en el grupo de hilos, luego seleccionaremos **“Añadir”**, después en el apartado de **“Receptor”** para finalmente seleccionar **“Ver árbol de resultados”**.



Paso 8

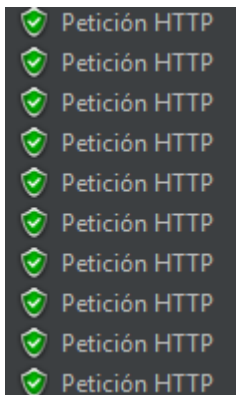
Finalmente ya teniendo todos los archivos necesarios, realizaremos la prueba de la siguiente manera.

Le daremos click derecho en el grupo de hilos daremos la opción de **“Star”** y empezara a realizar la prueba.



Ya finalizada la prueba se espera una prueba realizada satisfactoria y sin ningún problema.

Tal como se obtuvo en la siguiente imagen



Con estos resultados se finaliza la prueba.

Pruebas funcionales

Pruebas de integración

Este tipo de pruebas son ejecutadas por el equipo de desarrollo de **EFFECTIVE RECORD** y consisten en la comprobación de que elementos del software interactúan entre sí, además de esto se puede observar la forma de cómo funcionan de manera correcta los distintos módulos que componen la solución una vez que han sido probados unitariamente con el fin de comprobar que interactúan correctamente a través de sus interfaces internas o externas, cubriendo así lo que es la parte de la funcionalidad establecida por el cliente y que estos se ajusten a los criterios solicitados por el cliente final.

En este tipo de prueba se evidencia como los métodos de la clase interactúan entre sí con un formulario de registro para cumplir distintas funcionalidades, **EFFECTIVE RECORD** realiza una prueba donde se puede ver la funcionalidad de estos dos módulos trabajando entre sí para registrar un usuario en el sistema de información sin presentar ningún tipo de percance, ya que se realiza la prueba unitaria que nos ayuda a demostrar que los métodos de la clase están recibiendo los parámetros sin ningún tipo de problema.

MÉTODO QUE RESIBE LOS PARAMETROS

```
1  */
2  public class USUARIOS_DAO {
3
4      public String AdicionarUsuarios(USUARIOS Usuarios) {
5
6          String miRespuesta;
7          conexion miConexion = new conexion();
8          Connection nuevaCon;
9          nuevaCon = miConexion.getConn();
10
11          PreparedStatement sentencia;
12
13          try {
14
15              String Query = "Insert into USUARIOS (nombres_usuario, apellidos_usuario, identificacion_usuario, clave, telefono_usuario) values (?,?,?,?,?)";
16              sentencia = nuevaCon.prepareStatement(Query);
17
18              sentencia.setString(1, Usuarios.getNombres_usuario());
19              sentencia.setString(2, Usuarios.getApellidos_usuario());
20              sentencia.setString(3, Usuarios.getIdentificacion_usuario());
21              sentencia.setString(4, Usuarios.getClave());
22              sentencia.setString(5, Usuarios.getTelefono_usuario());
23
24              sentencia.executeUpdate();
25
26              return "Usuario Registrado con éxito.";
27          } catch (SQLException e) {
28              return "Error al registrar usuario: " + e.getMessage();
29          }
30      }
31  }
```

EJECUCIÓN DE PRUEBA

localhost:8080/Effective/RegistroUsuarios

localhost:8080 dice
Usuario Registrado con éxito.

Aceptar

Prueba funcional

Las pruebas funcionales se llevan a cabo para comprobar las características críticas para el negocio, la funcionalidad y la usabilidad. Las pruebas funcionales garantizan que las características y funcionalidades del software se comportan según lo esperado sin ningún problema. Valida principalmente toda la aplicación con respecto a las especificaciones mencionadas en el documento Software Requirement Specification (SRS). Los tipos de pruebas funcionales incluyen pruebas unitarias, pruebas de interfaz, pruebas de regresión, además de muchas.

Prueba 1

Modificar Producto	CPI	
	¿Prueba de despliegue?	Si/No
Descripción: El sistema deberá lograr la modificación de un producto que fue registrado con algún dato erróneo en el sistema de información. Esta modificación se realizara por parte del usuario.		
Prerrequisitos Tener registrado el producto previamente en la base de datos Tener presente el dato a modificar		
Pasos: Ingresar al módulo de productos Seleccionar el registro del producto con el dato erróneo Oprimir la opción de modificar Corregir el dato Seleccionar guardar		
Resultado esperado: Registrar el dato en el sistema de información con la modificación del dato erróneo y lograr una modificación correcta		
Resultado obtenido: El sistema logro modificar el dato que había sido registrado con el dato erróneo y pudo registrar nuevamente el producto en el sistema con normalidad		

Prueba 2

Consultar un producto	CP2	
	¿Prueba de despliegue?	Si/No
Descripción: Registrar un producto en el sistema de información, a través de esta prueba se lograra este cometido		
Prerrequisitos Tener el producto registrado en la base de datos Tener en cuenta el parámetro de búsqueda Asegurarse que el parámetro de búsqueda se halla registrado		
Pasos: Entrar al modulo Buscar el producto en el campo del buscador Seleccionar buscar el producto		
Resultado esperado: Consultar el producto exitosamente con sus datos ingresados previamente		
Resultado obtenido: El sistema logro encontrar el producto buscado por el usuario exitosamente		

Pruebas unitarias.

Este tipo de pruebas son ejecutadas normalmente por el equipo de desarrollo, básicamente consisten en la ejecución de actividades que le permitan verificar al desarrollador que los componentes unitarios están codificados bajo condiciones de robustez, esto es, soportando el ingreso de datos erróneos o inesperados y demostrando así la capacidad de tratar errores de manera controlada.

En el sistema de información **EFFECTIVE RECORD** se usa este tipo de prueba para verificar que los atributos de las clases estén recibiendo los parámetros correctamente sin ningún tipo de problema, es decir creamos una prueba para validar que los datos que nombramos al crear esta clase estén bien escritos y concuerden con la sintaxis del lenguaje de programación que se usa para desarrollar este sistema. Además de esto se verifica que los métodos que pertenecen a la clase estén depurando bien al momento de inicializar la prueba. Esta prueba como se puede evidenciar en la imagen se realiza de acuerdo al registro de usuarios del sistema y se puede observar además que se ejecuta de forma satisfactoria.

Cabe recalcar que estas pruebas se realizaron en la etapa de análisis donde se crearon las clases y sus métodos.

PRUEBA UNITARIA

```
public static void main(String[] args) {  
  
    USUARIOS_DAO mi_usuario_dao = new USUARIOS_DAO();  
    USUARIOS mi_usuario = new USUARIOS();  
  
    Scanner teclado = new Scanner(System.in);  
  
    String nombresusuario = "";  
    String apellidosusuario = "";  
    String identificacionusuario = "";  
    String claveusuario = "";  
    String telefonousuario;  
    String direccionusuario = "";  
    String correousuario = "";  
    int id_tipo_usuario;  
    int id_estado_usuario;  
    int id_genero_usuario;  
    int id_doc_identificacion_usuario;
```

EJECUCIÓN DE PRUEBA

```
Output - Effective (run)  
Carlos  
Ingrese los apellidos del usuario  
Andres  
Ingrese la nueva clave del usuario  
1212  
Ingrese la dirección del usuario  
Calle 68  
Ingrese el correo del usuario  
carlitos@gmail.com  
Ingrese el teléfono del usuario  
2003250  
Ingrese el número de identificación del usuario  
1000254110  
Ingrese el id del tipo usuario  
1  
Ingrese el id del estado del usuario  
1  
Ingrese el id del genero del usuario  
1  
Ingrese el id del tipo de documento del usuario  
1  
Conexión establecida con la base de datos effective  
Registro exitoso  
BUILD SUCCESSFUL (total time: 50 seconds)
```

Conclusiones

EFFECTIVE RECORD puede constatar que las diferentes pruebas que se realizaron en los métodos de las clases están funcionando en su totalidad ya que estas pruebas fueron diseñadas y estructuradas para que mostraran el funcionamiento completo de cada clase y sus métodos. Además de esto se puede evidenciar que se realizaron otras pruebas que muestran que algunas funcionalidades trabajando entre si con elementos del mismo sistema para realizar tareas en específico. Estas pruebas son las de integración que dejan examinar cómo funcionan componentes del sistema entre si sin presentar algún tipo de problema.

EFFECTIVE RECORD espera que las pruebas que se designaron anteriormente y que hacen falta por completarse cumplan con las expectativas finales, esto quiere decir que al momento de aplicarlas muestren algún tipo de error que se pueda corregir a tiempo, antes de que pase a pruebas de aceptación donde se le mostrara al cliente final que todos los requisitos que el planteo al comienzo del desarrollo del sistema están 100 % funcionales y no presentan ningún problema.

Además de esto se puede concluir lo siguiente.

Se puede concluir que la herramienta JMeter fue eficiente al momento de desplegar las pruebas, ya que es un software para pruebas de uso Open Source, bastante sencillo y asequible. Además, tiene una gran cantidad de documentación y tutoriales disponibles para la realización de pruebas, aparte fue una herramienta que sirve para pruebas funcionales en aplicaciones web

Ya estando en la herramienta se logra observar que es un entorno amigable, rápido, sencillo, es de fácil manejo y comprensión, ya que contaba con instrucciones que explicaban los procesos de forma sencilla y concreta para llevar a cabo las pruebas.

En cuanto a las pruebas, se concluye que los resultados fueron positivos, esto en base a que las pruebas se realizaron sin mayor complejidad, ya que el sistema no presento errores al momento de despliegue de las pruebas.

Por otra parte, los archivos previos para la preparación de la prueba fueron fáciles de encontrar y sencillos de configurar.

Anexos

VIDEO DE LA PRUEBA

Link: <https://drive.google.com/file/d/1meu9EBDtr3k2bfuEya3mkcXbnQZUNeP/view?usp=sharing>

BIBLIOGRAFÍA ELECTRÓNICA

<https://docs.cypress.io/>