

Tabla de contenido

| | |
|---|----|
| 1. Conceptos Básicos JMeter..... | 2 |
| ¿Qué es JMeter? | 2 |
| 1.1 ¿Cómo realiza JMeter las pruebas? | 3 |
| 1.2 Diagrama de flujo de trabajo de JMeter | 4 |
| 1.3 Características de JMeter..... | 4 |
| 1.4 ¿Qué son las pruebas de rendimiento? | 5 |
| 1.5 Herramientas utilizadas para las pruebas de rendimiento | 6 |
| 1.6 Tipos de prueba de rendimiento | 6 |
| 2. Instalación de JMeter | 7 |
| 3. Creación del Primer test | 10 |
| 3.2 Agregar un Sampler | 15 |
| Constant Timer | 16 |
| Uniform Random Timer | 18 |
| 3.3 Agregar un Assertion..... | 19 |
| 3.4 Agregar un Listener | 22 |
| 3.5 Visualización de Resultados | 23 |
| 3.6 Instalación del PerfMon Listener en JMeter..... | 25 |
| Instalación de Plugins Manager | 25 |
| Instalación del plugin Standard Set en JMeter | 26 |

1. Conceptos Básicos JMeter

1.1 ¿Qué es JMeter?

JMeter, también conocido como '**Apache JMeter**', es una aplicación de código abierto, 100% basada en Java, con una interfaz gráfica. Está diseñado para analizar y medir el rendimiento y el comportamiento funcional de la carga de la aplicación web y una variedad de servicios.

JMeter se utiliza principalmente para probar aplicaciones web o aplicaciones FTP, pero actualmente es aplicable en pruebas funcionales, conexiones de bases de datos JDBC, servicios web, conexiones TCP genéricas y procesos nativos del sistema operativo. Puede realizar varias actividades de prueba, como pruebas de rendimiento, carga, estrés, regresión y funcionales, para obtener métricas de rendimiento precisas en su servidor web.

JMeter no es un navegador y no presenta páginas HTML como lo hace cualquier navegador, sino que funciona a nivel de protocolo.

La siguiente es una lista de protocolos compatibles con JMeter:

- Servicios web - SOAP / XML-RPC
- Web - HTTP, sitios HTTPS 'web 1.0' web 2.0 (ajax, flex y flex-ws-amf)
- Base de datos a través de controladores JDBC
- Directorio - LDAP
- Servicio orientado a la mensajería a través de JMS
- Servicio - POP3, IMAP, SMTP



1.2 ¿Cómo realiza JMeter las pruebas?

Echemos un vistazo a los diferentes pasos realizados por JMeter durante las pruebas:

1. Crea una solicitud y la envía al servidor.
2. Recibe la respuesta del servidor, la recopila y visualiza esos detalles en un cuadro o gráfico.
3. Procesa la respuesta del servidor.
4. Genera el resultado de la prueba en varios formatos, como texto, XML, JSON para que el probador pueda analizar los datos.

1.3 Diagrama de flujo de trabajo de JMeter

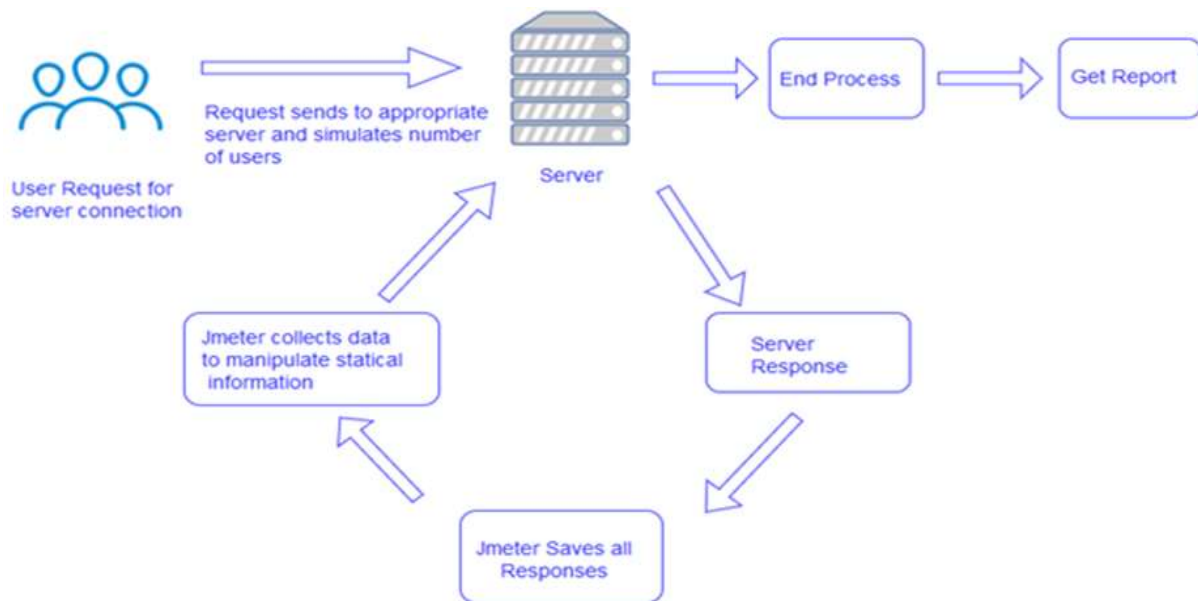


Figure 1: JMeter workflow Diagram

1.4 Características de JMeter

Algunas de las características más importantes de JMeter se enumeran a continuación:

- **Aplicación de código abierto:** JMeter es una aplicación gratuita de código abierto que facilita a los usuarios o desarrolladores el uso del código fuente para el desarrollo de otras aplicaciones.
- **GUI fácil de usar:** JMeter viene con una GUI simple e interactiva.
- **Admite varios enfoques de prueba:** JMeter admite varios enfoques de prueba, como pruebas de carga, pruebas distribuidas y pruebas funcionales, etc.

- **Independiente de la plataforma:** JMeter está escrito y desarrollado usando Java, por lo que puede ejecutarse en cualquier entorno/estación de trabajo que acepte una máquina virtual Java, por ejemplo, Windows, Linux, Mac, etc.
- **Admite varios tipos de servidores:** JMeter es altamente extensible y capaz de cargar la prueba de rendimiento en diferentes tipos de servidores:

Web: HTTP, HTTPS, SOAP,

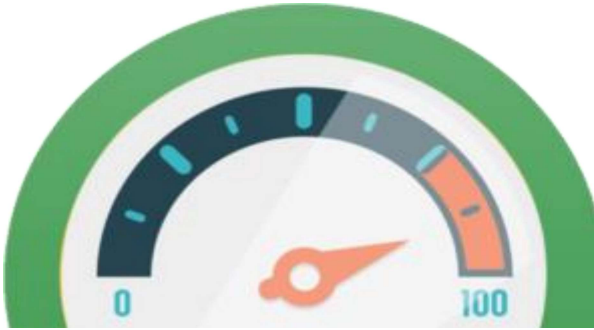
Base de datos: JDBC, LDAP, JMS y

Correo: POP3.

- **Compatibilidad con múltiples protocolos:** JMeter admite protocolos como HTTP, JDBC, LDAP, SOAP, JMS y FTP.
- **Simulación:** JMeter puede simular múltiples usuarios mediante el uso de usuarios virtuales o usuarios únicos para generar una carga pesada contra la aplicación web bajo prueba.
- **Marco:** JMeter es un marco de subprocesos múltiples que permite el muestreo concurrente y simultáneo de diferentes funciones por muchos o grupos de subprocesos separados.
- **Pruebas distribuidas remotas:** JMeter tiene el concepto Maestro-Esclavo para pruebas distribuidas donde el maestro distribuirá las pruebas entre todos los esclavos y los esclavos ejecutarán scripts contra su servidor.
- **Visualización de resultados** de prueba: los resultados de la prueba se pueden ver en diferentes formatos, como gráfico, tabla, árbol e informe, etc.

1.5 ¿Qué son las pruebas de rendimiento?

Las pruebas de rendimiento se definen como un tipo de prueba de software para garantizar que las aplicaciones de software funcionarán bien bajo su carga de trabajo esperada. Se centra en ciertos factores de un programa de software, tales como:



Velocidad – Comprueba si la respuesta de la aplicación es rápida.

Escalabilidad: determina la carga máxima de usuarios.

Estabilidad: comprueba si la aplicación es estable bajo cargas variables.

1.6 Herramientas utilizadas para las pruebas de rendimiento

Las pruebas de rendimiento son significativas en tiempo real, particularmente desde el punto de vista de la satisfacción del cliente. Hay varias herramientas de prueba de rendimiento disponibles, tales como:

- Apache JMeter
- LoadRunner
- WebLOAD
- LoadUI
- LoadView
- NeoLoad

1.7 Tipos de prueba de rendimiento

Prueba de Carga

Es una forma de evaluar el comportamiento de cualquier aplicación, cuando se adapta con diferentes cargas (usuarios / tráfico) en palabras simples, la prueba de carga ayuda a comprender cómo funciona un programa (supongamos que un sitio web «abc.com») cuando 20 usuarios están usando la plataforma en el mismo momento.

Prueba de estrés

Las pruebas de estrés son un método para analizar lo que podría ocurrir en determinados escenarios, por lo tanto, no se utiliza como una herramienta de predicción.

Prueba de estabilidad

El sistema es sometido a cargas robustas, (menores a las de estrés) la carga se debe mantener durante un tiempo prolongado y se evalúa la estabilidad del sistema mientras soporta la carga.

Prueba de picos

El sistema es sometido a cargas robustas, la carga debe ser aplicada en un tiempo muy reducido, se evalúa el comportamiento del sistema ante subidas y bajadas repentinas.

2. Instalación de JMeter

Paso 1 – Instalar Java



JMeter es una aplicación de escritorio Java pura y requiere una JVM 6 o superior totalmente compatible. Puede descargar e instalar la versión más reciente de Java SE Development Kit.

Puede comprobar en el símbolo del sistema si la instalación se ha realizado correctamente. Le dará la siguiente salida:

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.19045.2251]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\nchaverra>java -version
java version "1.8.0_351"
Java(TM) SE Runtime Environment (build 1.8.0_351-b10)
Java HotSpot(TM) 64-Bit Server VM (build 25.351-b10, mixed mode)

C:\Users\nchaverra>
```

Paso 2 – Descargar JMeter

Apache JMeter 5.5 (requiere Java 8+)

Binarios

⇒ [apache-jmeter-5.5.tgz sha512 PGP](#)
[apache-jmeter-5.5.zip sha512 PGP](#)

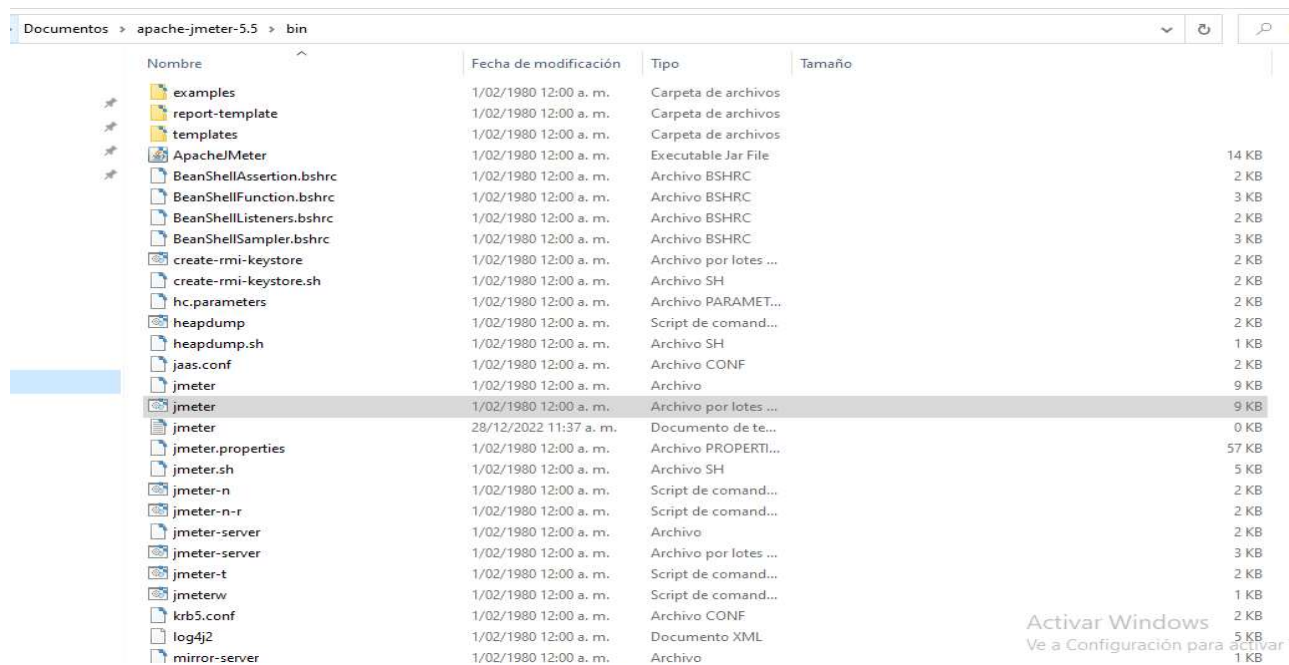
Fuente

[apache-jmeter-5.5_src.tgz sha512 PGP](#)
[apache-jmeter-5.5_src.zip sha512 PGP](#)

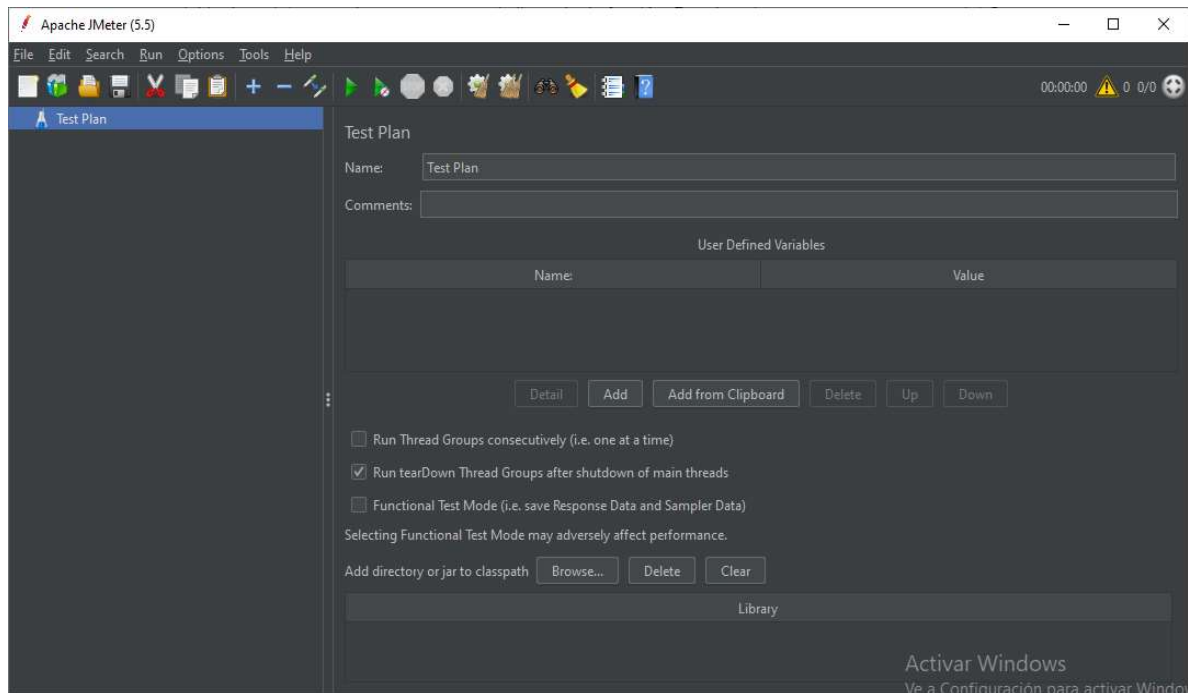
La última versión de JMeter disponible es 5.5. Puede descargar cualquiera de los binarios.

Paso 3 – Instalar JMeter

Descomprima el archivo zip / tar en el directorio donde desea que se instale JMeter.

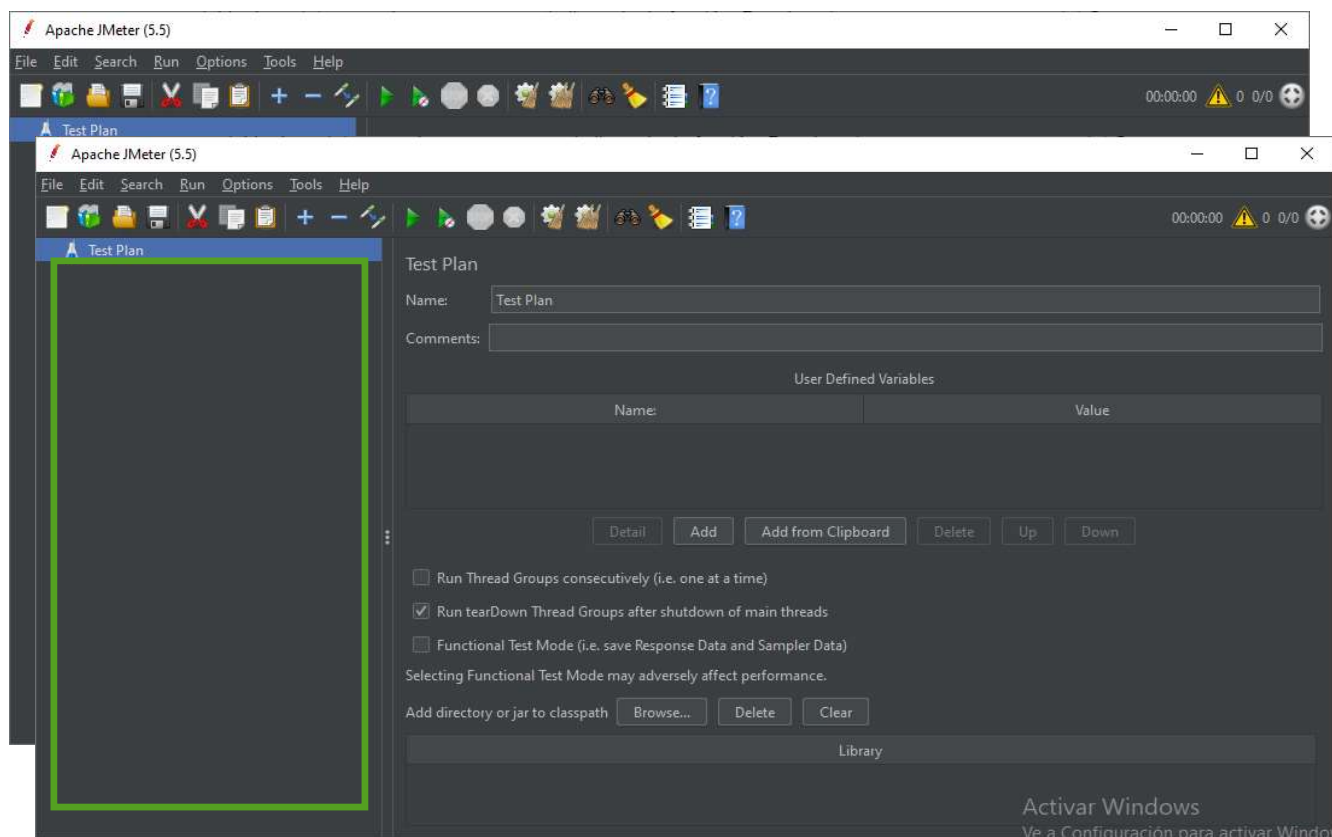


Si está utilizando Windows, simplemente ejecute el archivo **/bin/jmeter** para iniciar JMeter en modo GUI:



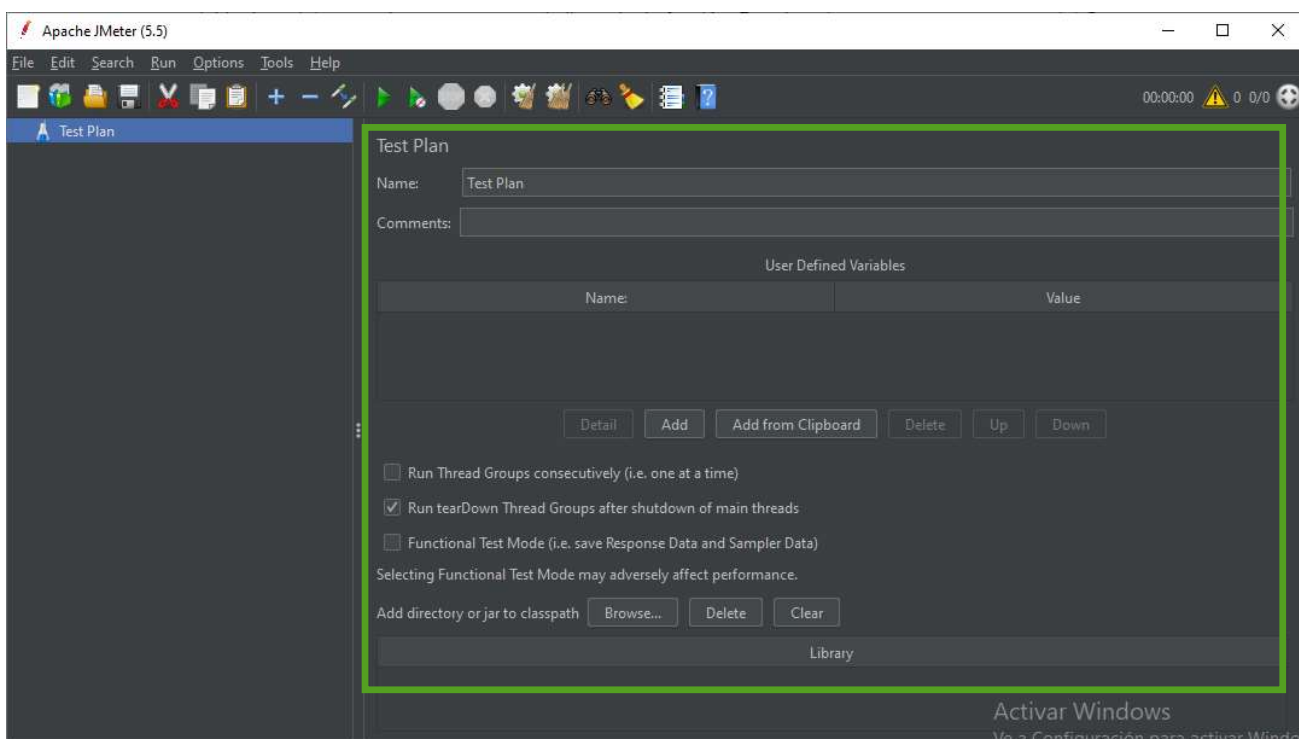
3. Creación del Primer test

Pantalla con la que se inicia en JMeter



En el frame izquierdo se ve los elementos que componen el plan de pruebas.

En el frame derecho, se ve el detalle o la configuración del elemento que se está seleccionando en el frame izquierdo.

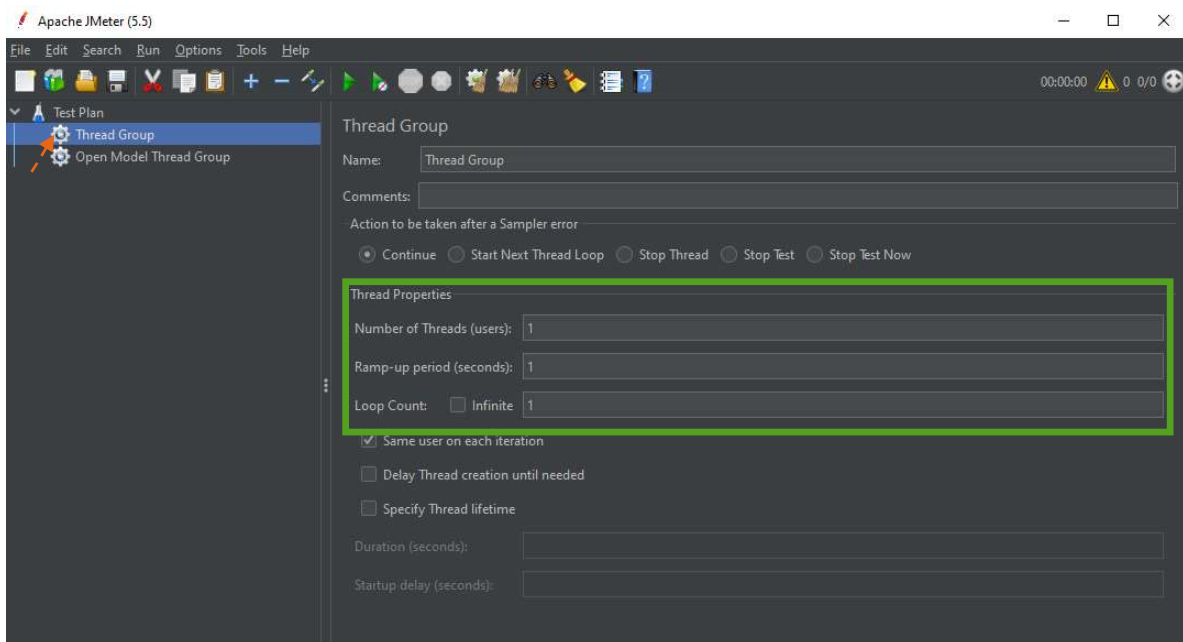
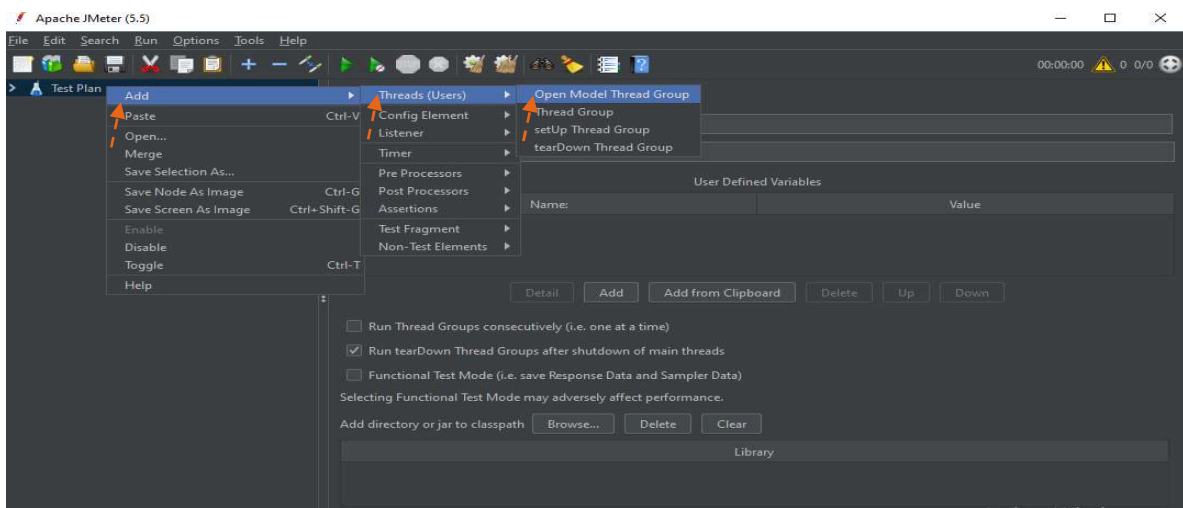
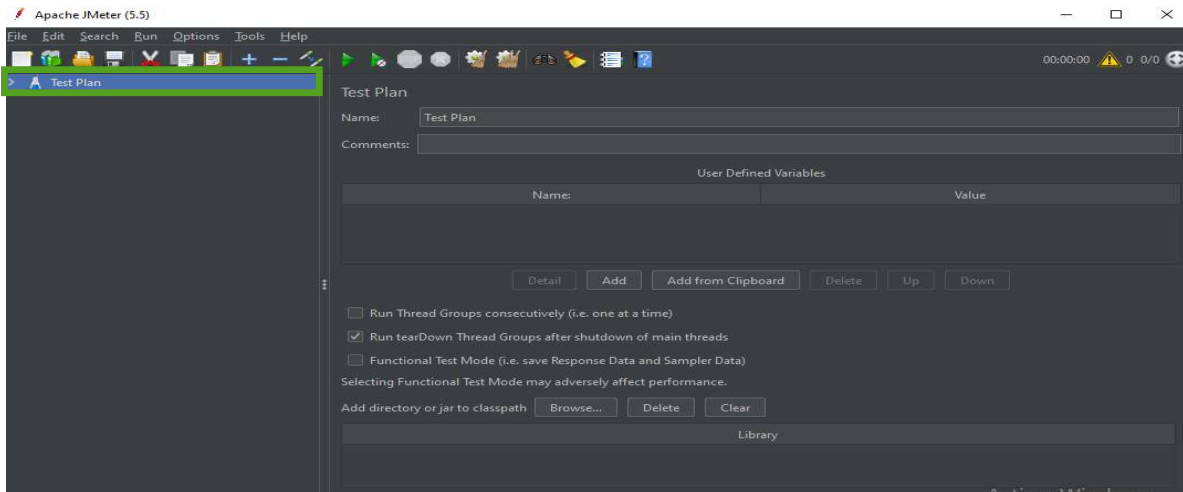


Test Plan: Es como un contenedor que almacena los elementos necesarios para ejecutar una prueba. Es la raíz de nuestro script.

3.1 Agregar un Thread Group

Un Thread Group es el conjunto de usuarios virtuales, (threads) que serán enviados al servicio que se está probando. Son semejantes a varios robots de Selenium.

Se da click derecho en Test Plan y se selecciona Add//Threads (Users)//Thread Group



Thread Properties

Number of Threads (Users)

Es el total de usuarios virtuales que quiere enviar durante la prueba.

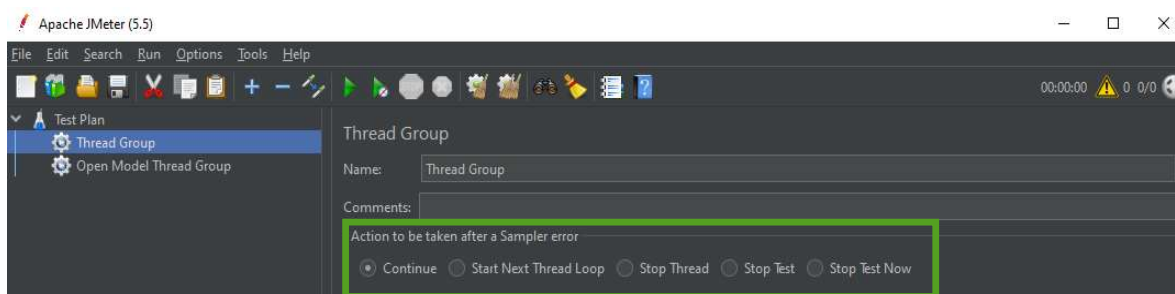
Ramp-up Period (in seconds)

Es una rampa que determina en cuánto tiempo se ha enviado el total de usuarios virtuales al servicio que se está probando

Loop Count:

Es la cantidad de ciclos en los que se puede estar enviando el Thread Group actual.

Action to be taken after a Sampler error



Continue: Continuar, aunque aparezca algún error durante la ejecución de algún Thread.

Start Next Thread Loop: Inicia el siguiente loop (ciclo) si encuentra un error durante la ejecución del loop actual.

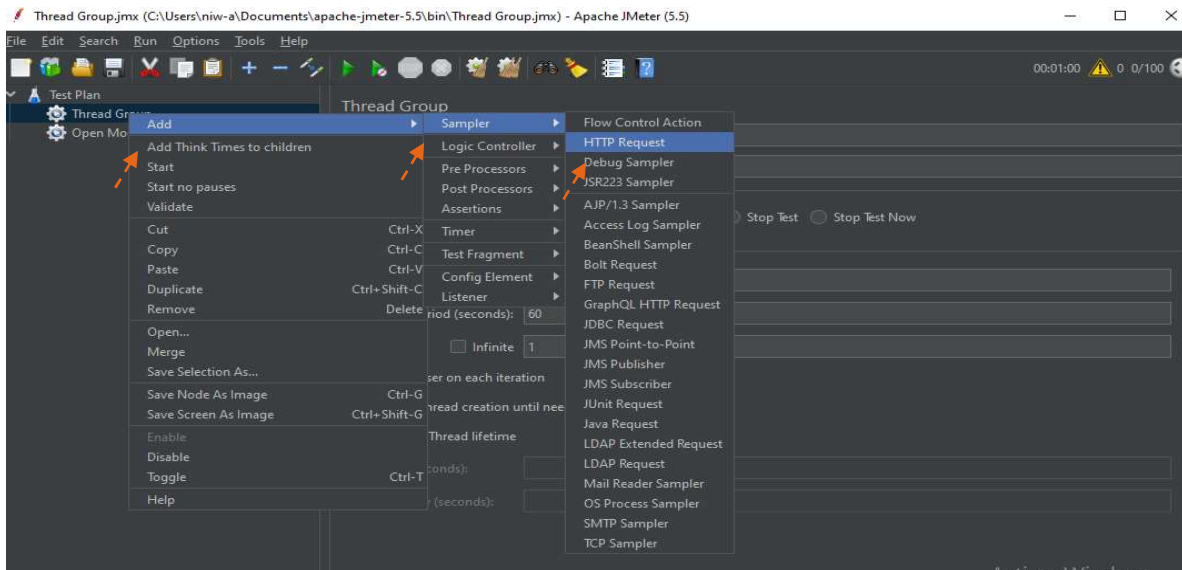
Stop Thread: Detiene la ejecución del loop actual y no inicia los demás ciclos, todo el Thread Group se detiene.

Stop Test: Se detiene la prueba en total, los threads que estén en ejecución se completan antes de detener la prueba.

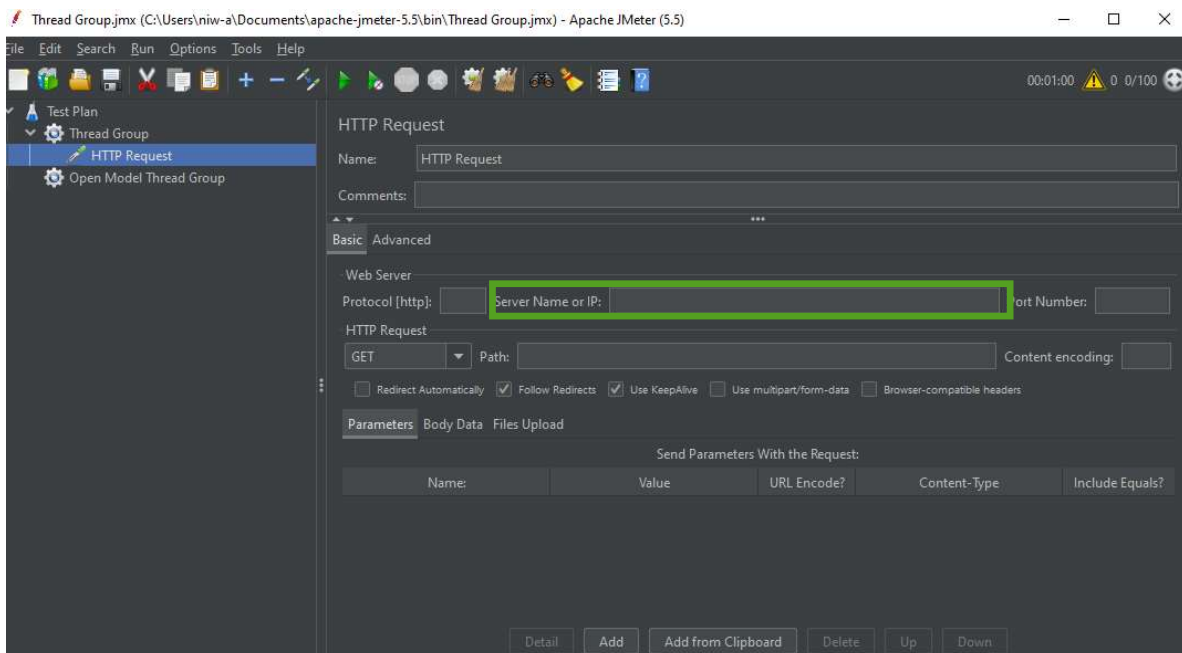
Stop Test Now: Se detiene la prueba de forma inmediata, es un stop más agresivo y no espera a los threads en ejecución.

3.2 Agregar un Sampler

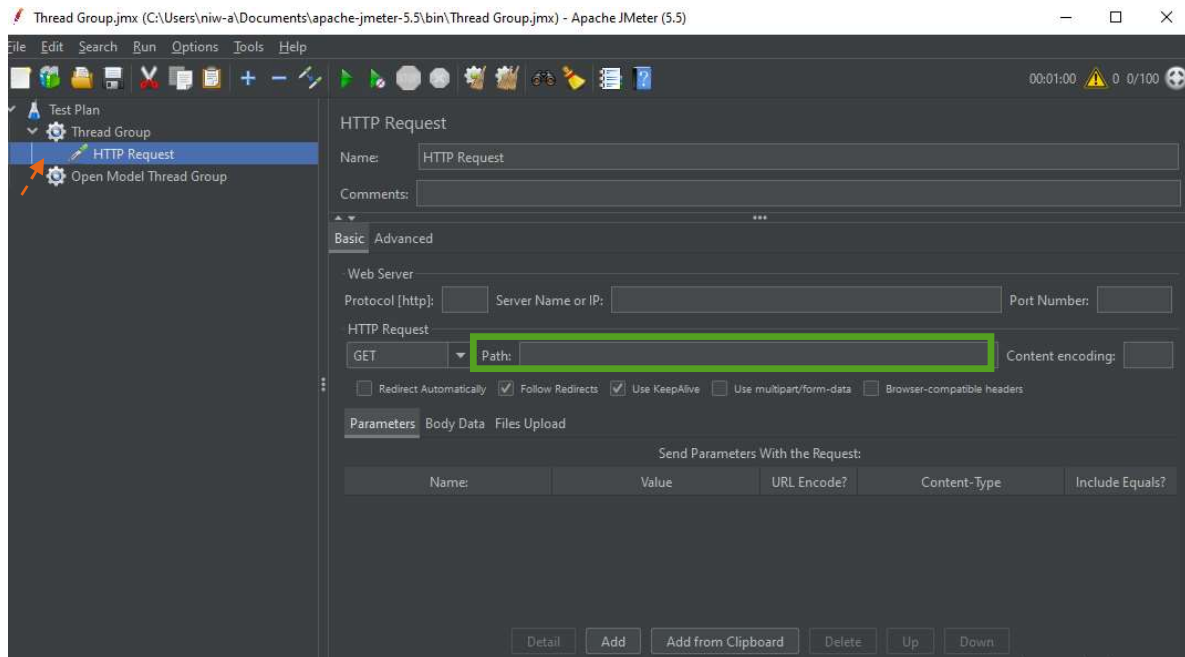
Sampler es el que va a tomar los threads y los va a enviar hacia el aplicativo deseado, para agregar un sampler (en este caso un HTTP Request) se da clic derecho en el Thread Group agregado y continúan la ruta Add//Sampler//HTTP Request.



Con el Sampler agregado, se procede a seleccionar y a escribir el nombre o la IP del servidor que se va a probar



Dado que la prueba se va a realizar en el home page, se agrega un / a la casilla Path:

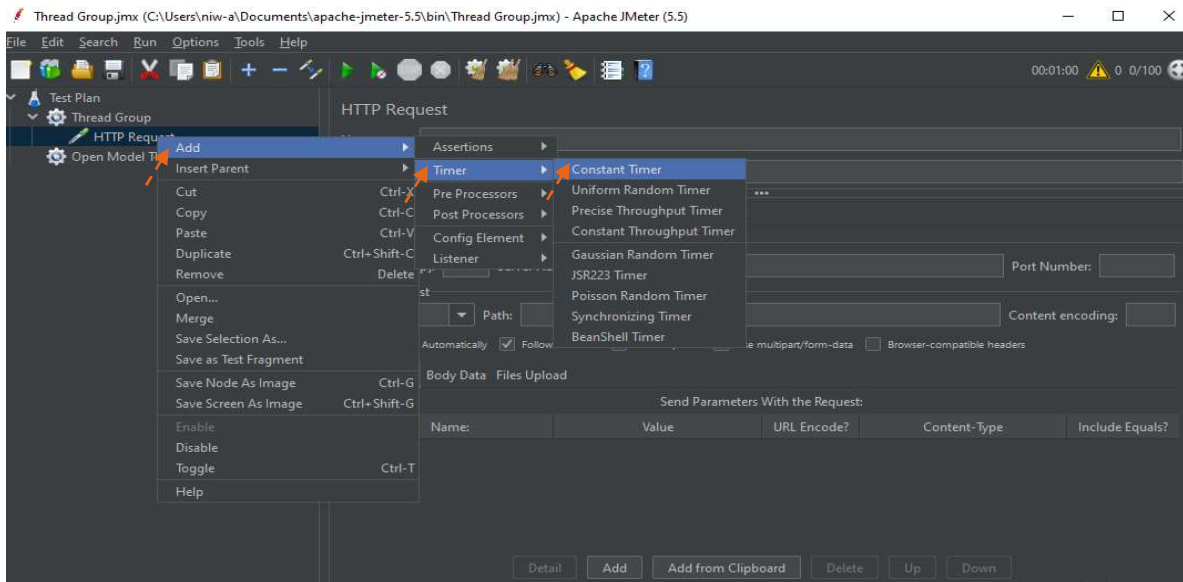


Agregar un Timer

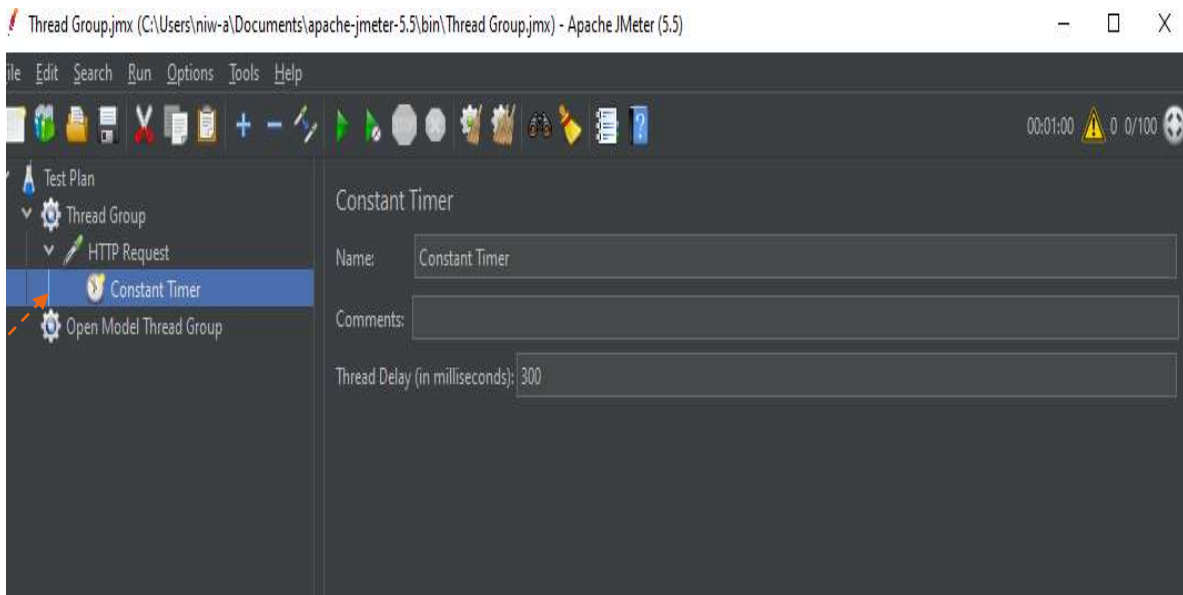
Los temporizadores le dan realismo a las pruebas, ya que los threads no deberían ejecutar acciones de forma inmediata, como clics con diferencias de milisegundos.

Constant Timer

Para agregar un temporizador constante al HTTP Request, se da clic derecho sobre el sampler y se selecciona Add//Timer//Constant Timer.

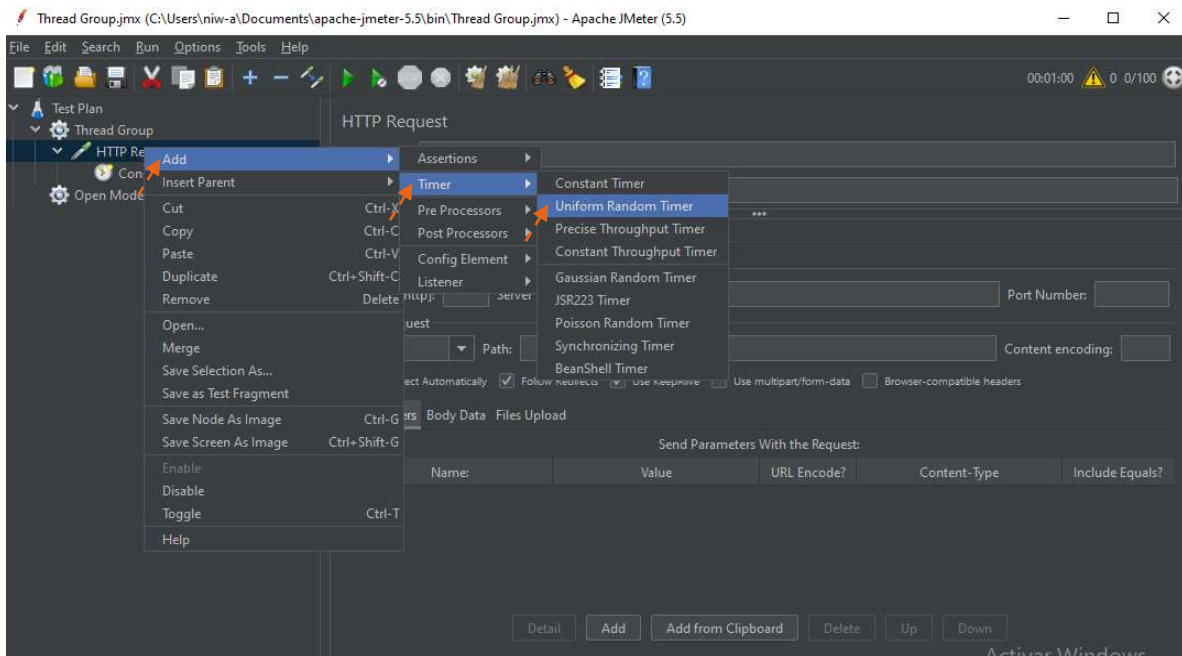


Agregando el constant timer al sampler, se aplica un delay a cada thread que se envíe al servidor, probando que el valor del delay puede ser cambiado en la casilla señalada.

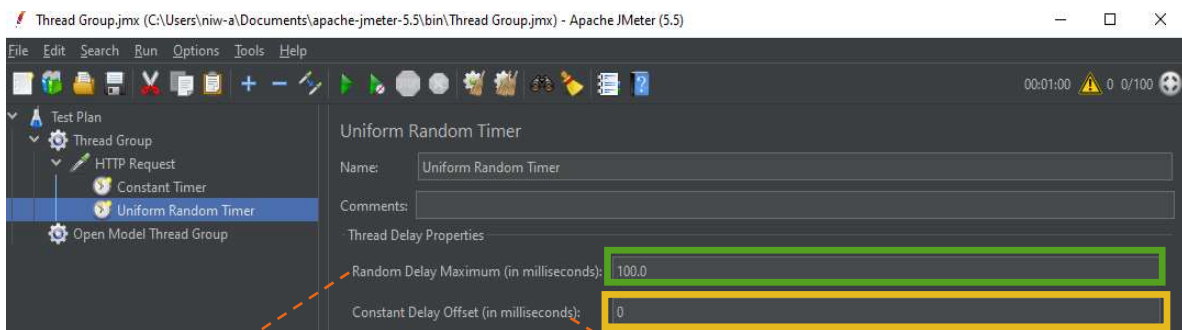


Uniform Random Timer

El temporizador agrega un delay constante + un delay variable, para añadir un uniform random timer, se selecciona Add//Timer//Uniform Random Timer.



Los campos que se deben configurar están señalados en la imagen

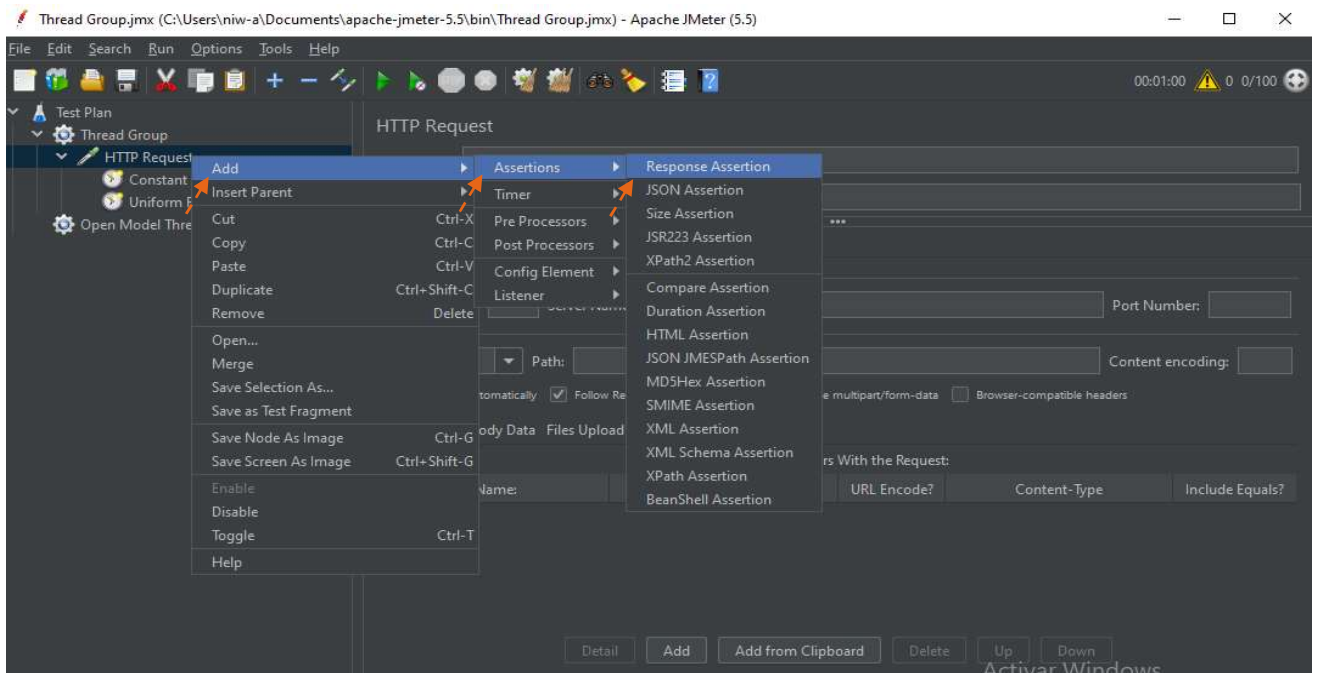


El Random Delay Maximum es el valor máximo de delay variable que se agregará a la ejecución de cada thread

Constant Delay Offset es el valor de delay constante que tendrá la ejecución de cada thread

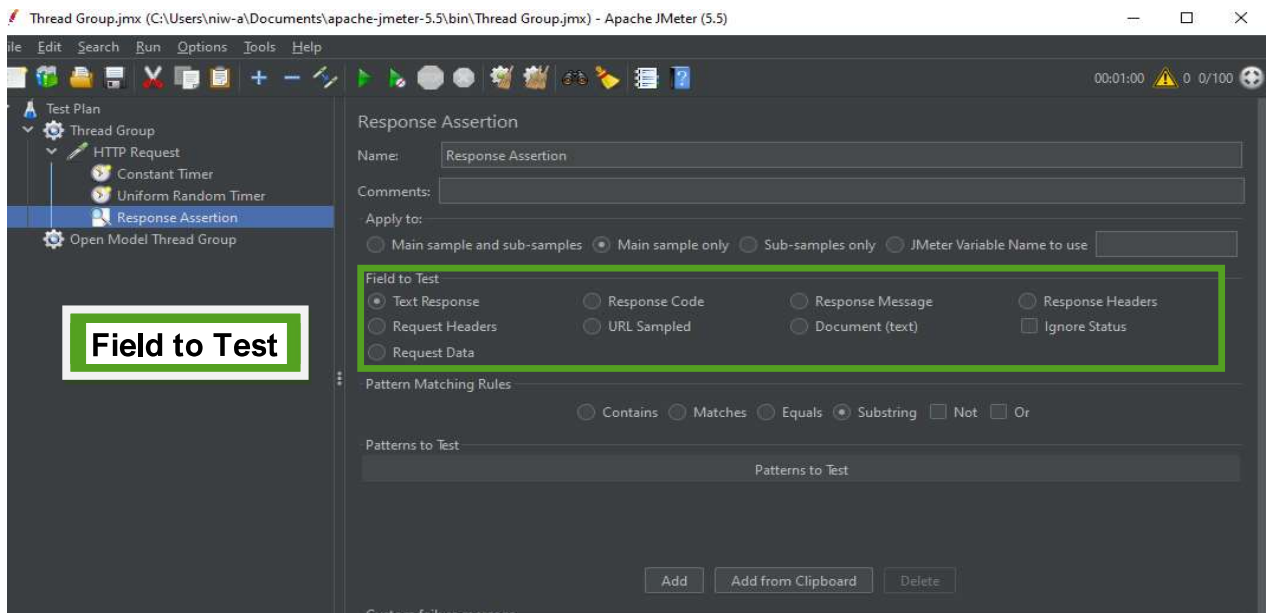
3.3 Agregar un Assertion

Con los asserts validamos que se están recibiendo las respuestas que se esperan por parte del servidor, también sirven para validar lo que está siendo dirigido a la página correcta. Para agregar un assert, se da clic derecho en el sampler y se selecciona Add//Assertions/Response Assertion.



En el Response Assertion, se identifica lo siguiente en el área de opciones **Field to Test**

- **Text Response:** se valida el contenido que recibe en las etiquetas body del código HTML.
- **URL Sampled:** se valida la URL a la que se dirige.
- **Response Code:** en él se valida el número de respuesta recibida. 200 si está OK, 404 si no fue encontrado, 500 si hay un error en el servidor, 302 se redirige a otra página.
- **Response Message:** Validar el mensaje que el servidor envía en el Response, en el caso de recibir un 200 en el Response Code, el Response Message sería un OK por parte del servidor.
- **Response Headers:** se valida Los headers del Response.
- **Document (text):** se analiza metadata o el contenido de un documento, como un archivo de Word, powerpoint, txt, etc.



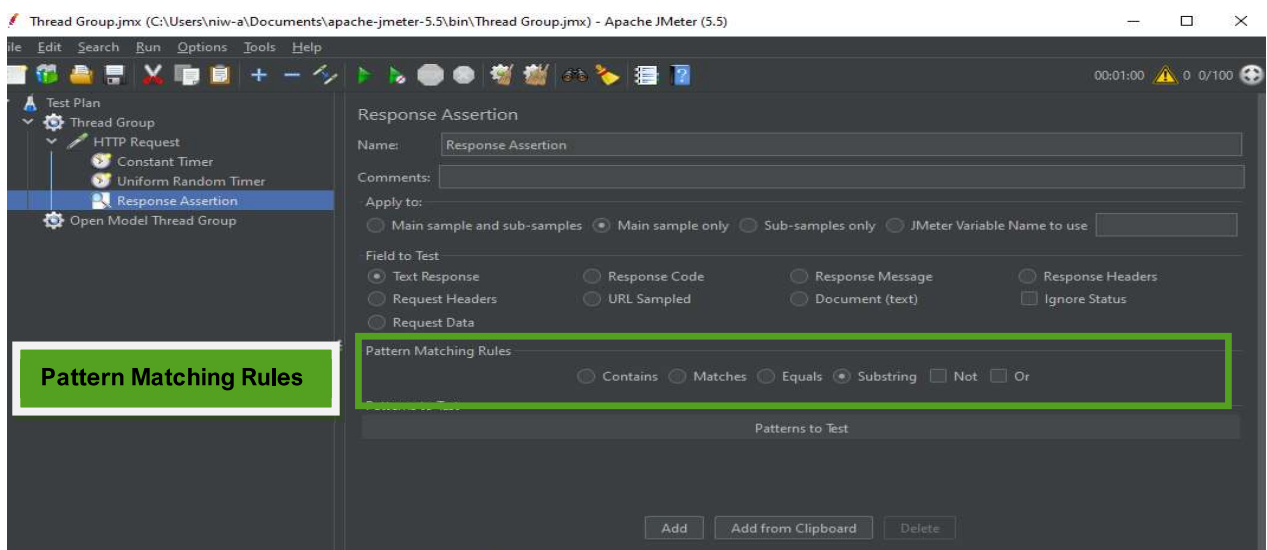
En el área de **Pattern Matching Rules**, identificamos lo siguiente:

Contains: Utiliza expresiones regulares (RegEx) validar que el texto que se recibe por parte del servidor contiene el texto que se indica en el assert.

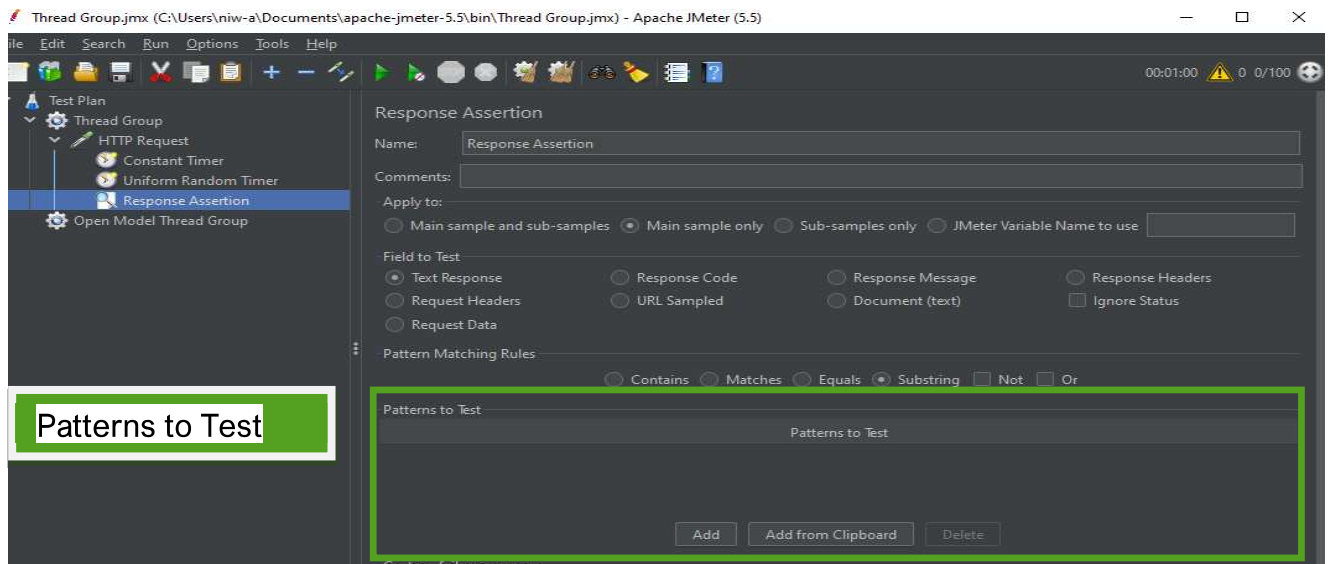
Matches: Usa expresiones regulares (RegEx) es similar a Contains, con la diferencia de que el texto que se recibe por parte del servidor debe ser completamente igual al que se indica en el assert.

Equals: Usa texto plano, su validación es igual a la de Matches.

Substring: Usa texto plano, su validación es igual a la de Contains.



En el área Patterns to Test, se agrega el texto que se quiere validar, en ese caso se agrega la palabra Welcome, junto a la configuración del Response Assertion presentada en la imagen.

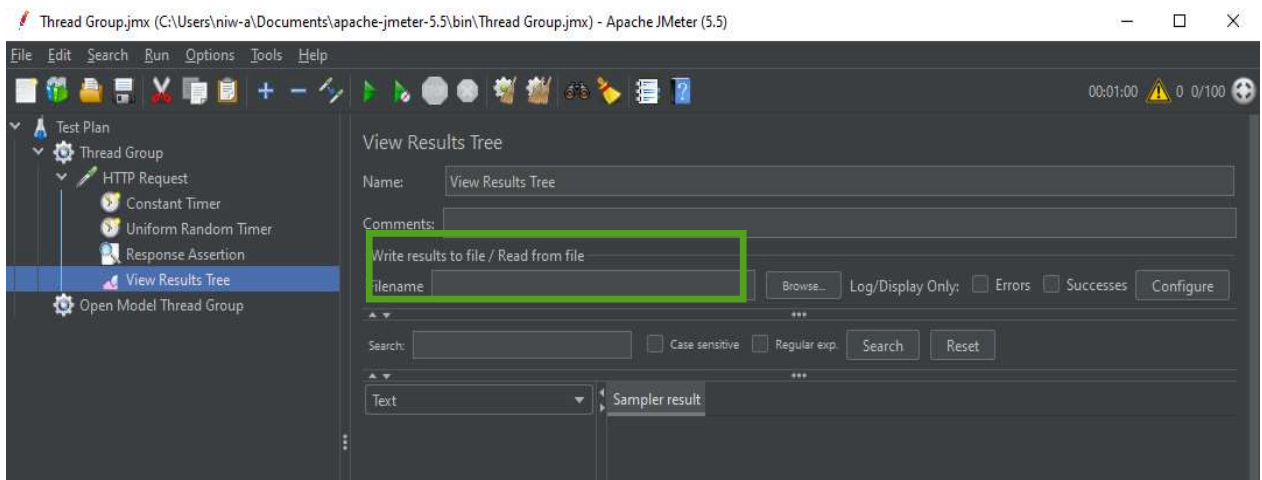
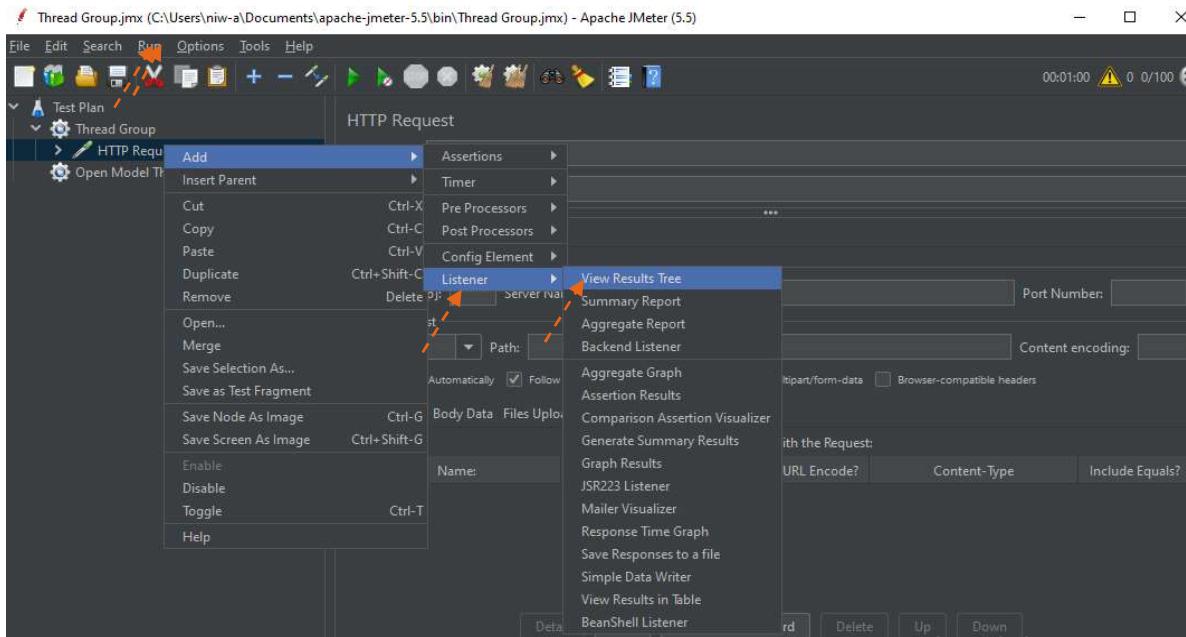


3.4 Agregar un Listener

Un Listener es aquel que está pendiente de lo que sucede durante la ejecución de una prueba

Para agregar un View Results Tree, se selecciona Add//Listener//View Results Tree.

Un View Results Tree, permite ver los resultados del Test como si estuviera siendo ejecutado en modo debug, en el cuadro naranja se especifica un archivo donde se quiere que JMeter guarde los resultados, o se especifica un archivo que contenga resultados de pruebas anteriores.

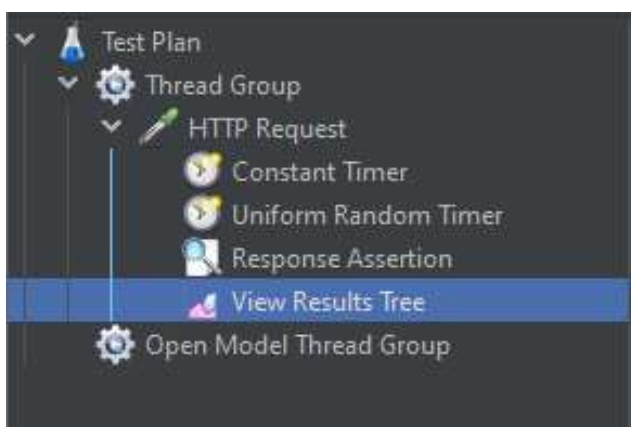


The screenshot shows the Apache JMeter 5.5.3 application window. The title bar indicates the file path: "Thread Group.jmx (C:\Users\niw-a\Documents\apache-jmeter-5.5\bin\Thread Group.jmx) - Apache JMeter (5.5)". The main menu includes File, Edit, Search, Run, Options, Tools, and Help. The left sidebar shows a tree structure with "Test Plan" expanded, containing "Thread Group" and "HTTP Request". The "View Results Tree" node is selected. The main panel displays the "View Results Tree" view, showing a table with columns for Name, Response Code, Response Message, and Response Data. The "Sample Result Save Configuration" dialog box is open, allowing users to configure which data to save. The dialog has a title bar with a close button (X). It contains a list of checkboxes for various save options, organized in three columns. The "Configure" button in the background is highlighted with an orange arrow.

| Sample Result Save Configuration | | |
|--|--|--|
| <input type="checkbox"/> Save As XML | <input checked="" type="checkbox"/> Save Field Names (CSV) | <input checked="" type="checkbox"/> Save Time Stamp |
| <input checked="" type="checkbox"/> Save Elapsed Time | <input checked="" type="checkbox"/> Save Label | <input checked="" type="checkbox"/> Save Response Code |
| <input checked="" type="checkbox"/> Save Response Message | <input checked="" type="checkbox"/> Save Thread Name | <input checked="" type="checkbox"/> Save Data Type |
| <input checked="" type="checkbox"/> Save Success | <input checked="" type="checkbox"/> Save Assertion Failure Message | <input checked="" type="checkbox"/> Save received byte count |
| <input checked="" type="checkbox"/> Save sent byte count | <input checked="" type="checkbox"/> Save Active Thread Counts | <input checked="" type="checkbox"/> Save URL |
| <input type="checkbox"/> Save Response Filename | <input checked="" type="checkbox"/> Save Latency | <input checked="" type="checkbox"/> Save Connect Time |
| <input type="checkbox"/> Save Encoding | <input type="checkbox"/> Save Sample and Error Counts | <input type="checkbox"/> Save Hostname |
| <input checked="" type="checkbox"/> Save Idle Time | <input type="checkbox"/> Save Request Headers (XML) | <input type="checkbox"/> Save Sampler Data (XML) |
| <input type="checkbox"/> Save Response Headers (XML) | <input type="checkbox"/> Save Response Data (XML) | <input checked="" type="checkbox"/> Save Sub Results |
| <input checked="" type="checkbox"/> Save Assertion Results (XML) | | |

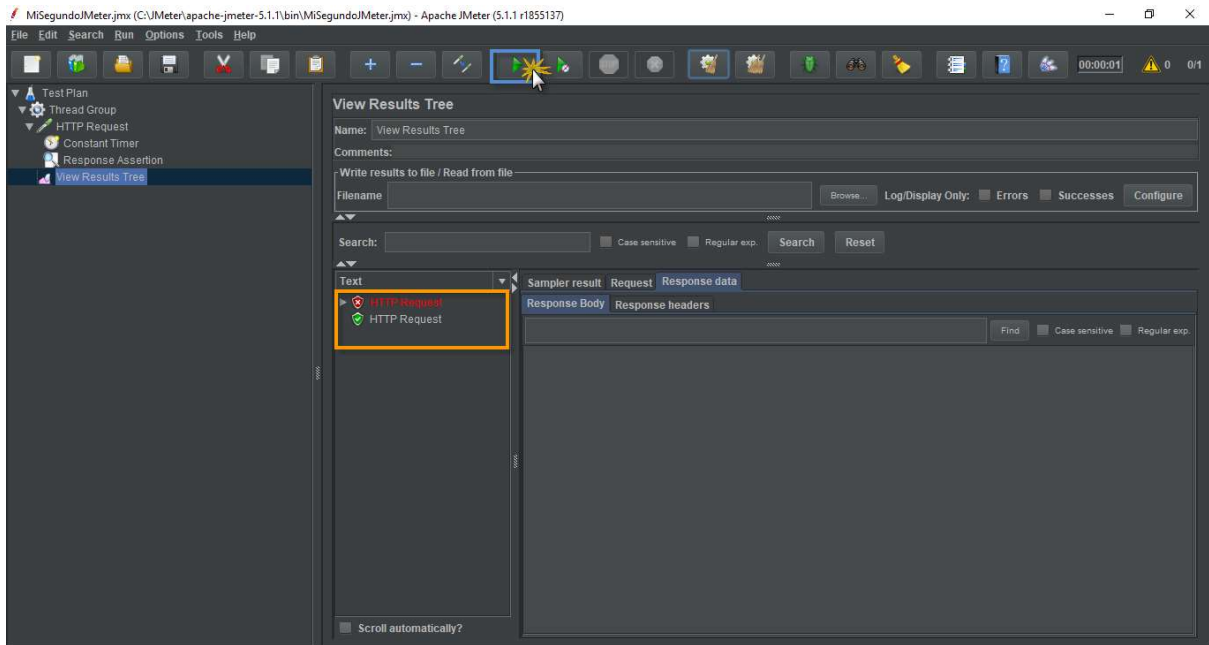
3.5 Visualización de Resultados

El Test Plan debería tener la siguiente estructura.

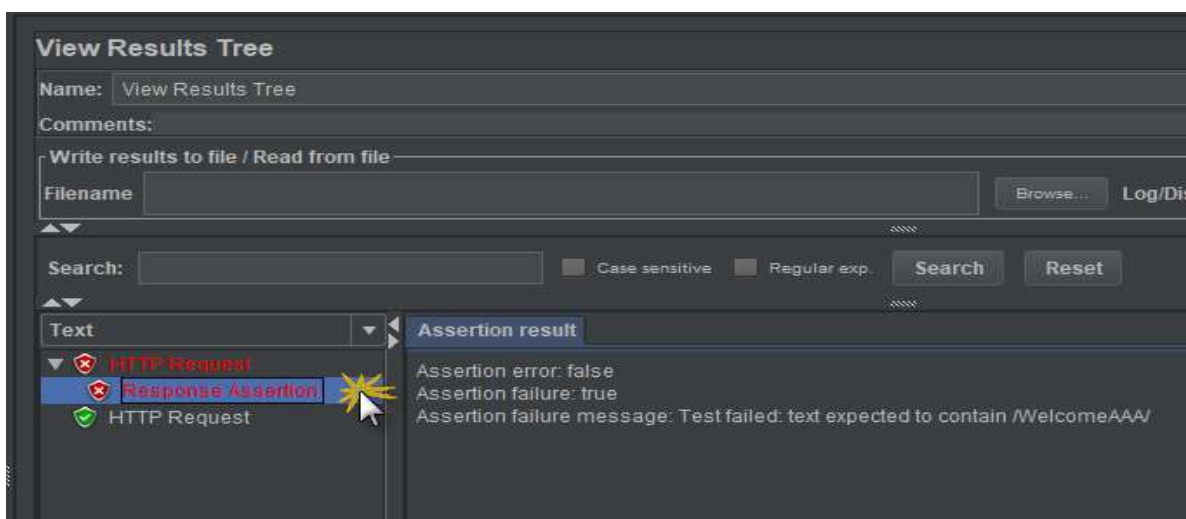


En el View Results Tree, se puede observar el resultado de las ejecuciones; cada vez que se presiona el botón Verde Start ubicado en la parte superior central, se ejecuta el Test Plan, en ese caso el botón Start se presionó dos veces, donde en la primera ejecución se alteró el valor del Assertion para que falle (el assert esperaba

un mensaje que contenga WelcomeAAA) y en la segunda ejecución se corrigió el valor esperado en el assert para que la prueba no falle.



Para observar un error se expande el resultado del HTTP Request fallido y se da clic en la opción Response Assertion, el mensaje que se recibe indica que acciones tomar.



3.6 Instalación del PerfMon Listener en JMeter

Instalación de Plugins Manager

Con la herramienta Plugins Manager es más fácil gestionar los plugins de JMeter, para descargarlo ingrese al siguiente link

<https://jmeter-plugins.org/wiki/PluginsManager/>

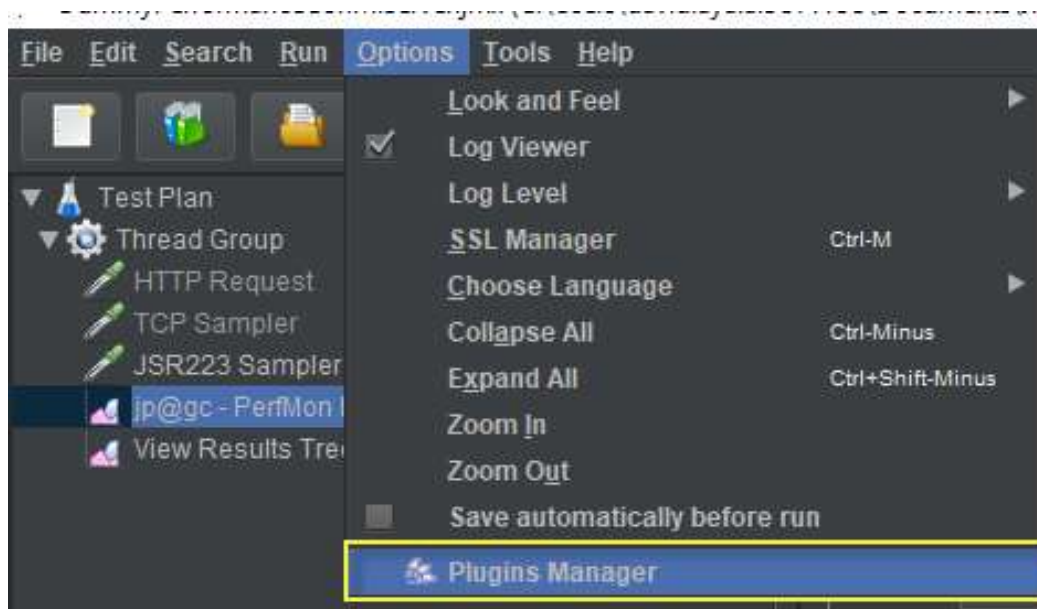
The screenshot shows the JMeter Plugins Manager website at <https://jmeter-plugins.org/wiki/PluginsManager/>. The page title is "JMeter Plugins Manager" and it includes a description of the tool and its benefits. A red box highlights the text: "Download the Plugins Manager JAR file and put it into JMeter's lib/ext directory. Then start JMeter and go to 'Options' menu to access the Plugins Manager." Below this, a JMeter Options dialog box is shown with the "Options" tab selected, and the "Function Helper Dialog Look and Feel" option is visible. A red box highlights the "Options" menu item in the JMeter application window.

Cuando el jar esté descargado, agregarlo en la ruta lib/ext de JMeter.

The screenshot shows a Windows file explorer window displaying the contents of the JMeter lib/ext directory. The path shown in the address bar is "Este equipo > Documentos > JMeter > apache-jmeter-5.1.1 > lib > ext". The file list includes various JMeter components and plugins. A red box highlights the file "jmeter-plugins-manager-1.3.jar", which is the file to be installed.

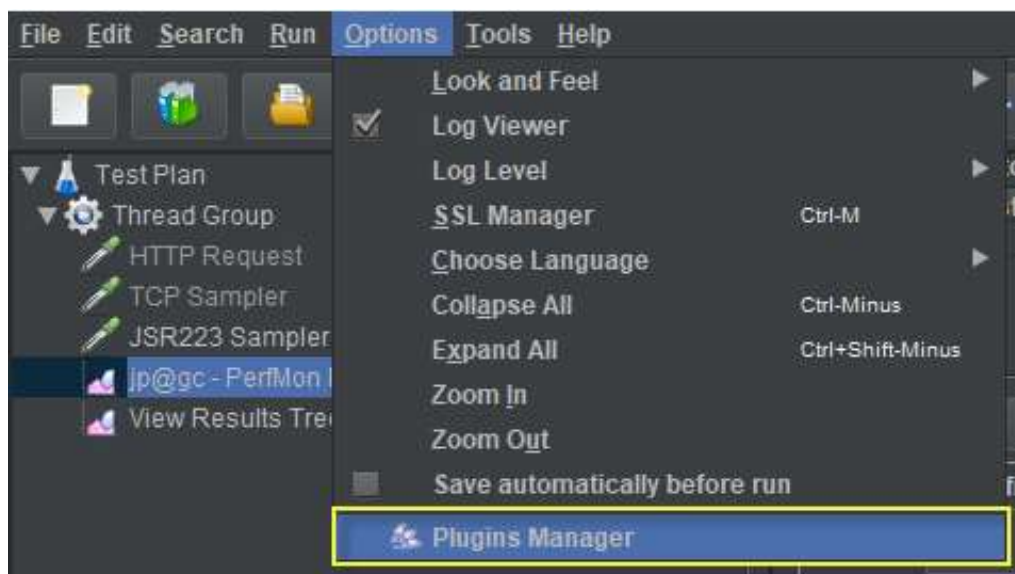
| Nombre | Fecha de modifica... | Tipo | Tamaño |
|-----------------------------------|----------------------|---------------------|----------|
| ApacheJMeter_components | 10/03/2019 9:58 | Executable Jar File | 683 KB |
| ApacheJMeter_core | 10/03/2019 9:58 | Executable Jar File | 1,641 KB |
| ApacheJMeter_ftp | 10/03/2019 9:58 | Executable Jar File | 15 KB |
| ApacheJMeter_functions | 10/03/2019 9:58 | Executable Jar File | 113 KB |
| ApacheJMeter_http | 10/03/2019 9:58 | Executable Jar File | 453 KB |
| ApacheJMeter_java | 10/03/2019 9:58 | Executable Jar File | 45 KB |
| ApacheJMeter_jdbc | 10/03/2019 9:58 | Executable Jar File | 56 KB |
| ApacheJMeter_jms | 10/03/2019 9:58 | Executable Jar File | 95 KB |
| ApacheJMeter_junit | 10/03/2019 9:58 | Executable Jar File | 23 KB |
| ApacheJMeter_ldap | 10/03/2019 9:58 | Executable Jar File | 48 KB |
| ApacheJMeter_mail | 10/03/2019 9:58 | Executable Jar File | 59 KB |
| ApacheJMeter_mongodb | 10/03/2019 9:58 | Executable Jar File | 26 KB |
| ApacheJMeter_native | 10/03/2019 9:58 | Executable Jar File | 14 KB |
| ApacheJMeter_tcp | 10/03/2019 9:58 | Executable Jar File | 30 KB |
| jmeter-plugins-casutg-2.9 | 18/06/2019 8:35 | Executable Jar File | 69 KB |
| jmeter-plugins-dummy-0.3 | 18/06/2019 8:35 | Executable Jar File | 13 KB |
| jmeter-plugins-ffw-2.0 | 18/06/2019 8:35 | Executable Jar File | 15 KB |
| jmeter-plugins-fifo-0.2 | 18/06/2019 8:35 | Executable Jar File | 18 KB |
| jmeter-plugins-functions-2.1 | 18/06/2019 8:35 | Executable Jar File | 29 KB |
| jmeter-plugins-graphs-basic-2.0 | 18/06/2019 8:35 | Executable Jar File | 9 KB |
| jmeter-plugins-manager-1.3 | 18/06/2019 8:34 | Executable Jar File | 873 KB |
| jmeter-plugins-perfmon-2.1 | 18/06/2019 8:35 | Executable Jar File | 41 KB |
| jmeter-plugins-tst-2.5 | 18/06/2019 8:35 | Executable Jar File | 17 KB |
| readme | 10/03/2019 8:36 | Documento de tex | 1 KB |

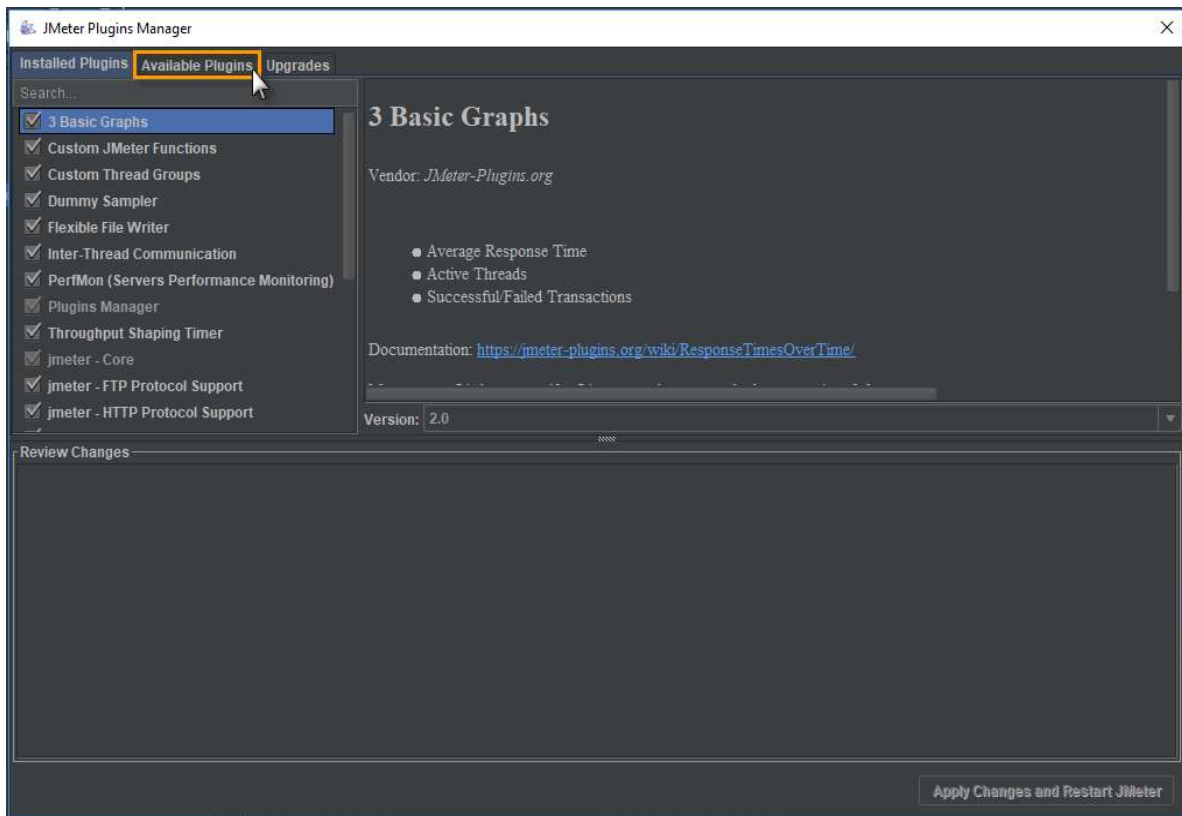
Si tiene JMeter abierto, se cierra y se vuelve a abrir, dirijase al menú y seleccione la lista Options.



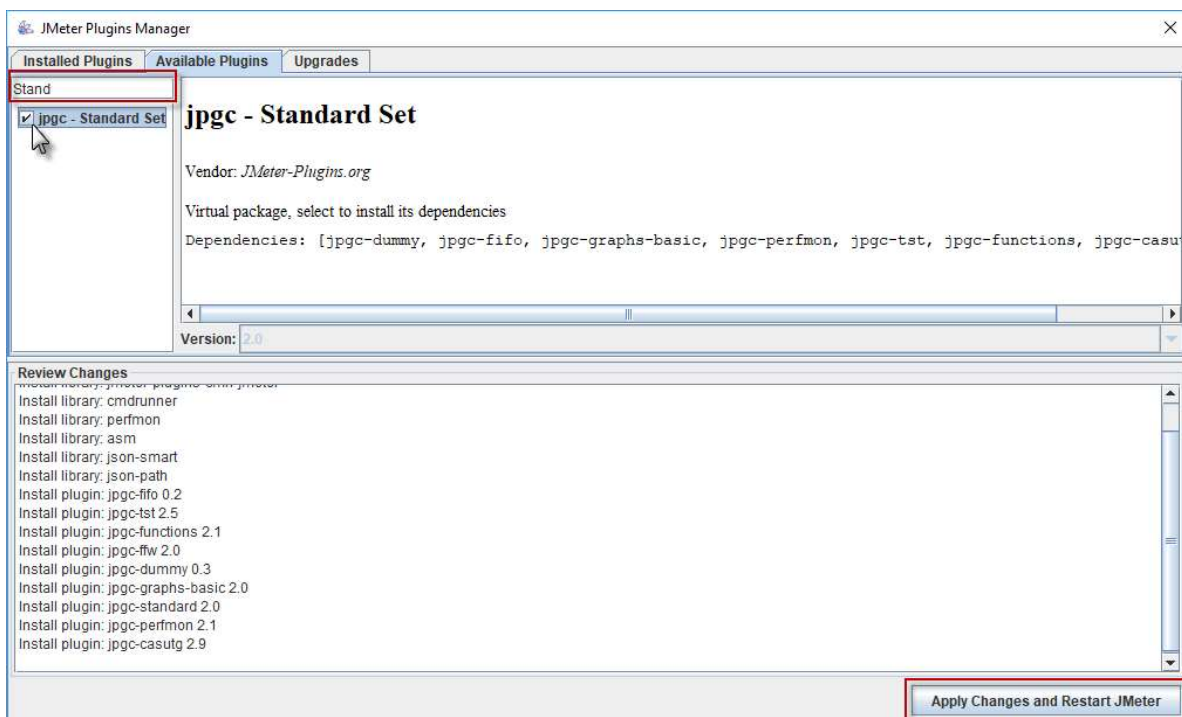
Instalación del plugin Standard Set en JMeter

Abrir el Plugins Manager





Buscar el Standard Set seleccionarlo, y hacer clic sobre el botón 'Apply Changes and Restart JMeter'.



Cuando JMeter se reinicie, tendrá disponibles nuevas opciones para armar sus pruebas de rendimiento, en ese caso el que se usará es el listener PerfMon Metrics Collector. Este recibirá los datos enviados por el PerfMon Server Agent y los verificará o guardará.

