



**El futuro
es de todos**

DNP
Departamento
Nacional de Planeación

GUIA PARA CREAR Y EJECUTAR PRUEBAS DE CARGA JMETER

**Oficina de Tecnologías y sistemas de Información
Grupo de Gestión de Sistemas de Información**

**Departamento Nacional de Planeación
Bogotá, 2020**

Control de Versiones

Versión	Fecha	Descripción	Autores
0	20-08-2020	Elaboración del documento	Clara Inés Martínez Pineda

Derechos de Autor: La elaboración de este documento y sus diferentes componentes estuvo a cargo del Grupo de Gestión de Sistemas de Información de la Oficina de Tecnologías y Sistemas de Información del Departamento Nacional de Planeación, DNP, razón por la cual los Derechos de Autor y en lo particular los derechos patrimoniales de este documento y su contenido pertenece exclusivamente al DNP. Por lo tanto, su uso y reproducción por terceros, está sujeto a la autorización expresa de la Oficina de Tecnologías y Sistemas de Información, OTSI del DNP en cumplimiento de la Ley 23 de 1982 y demás que la modifican o adicionan. Siendo así, este documento está protegido por Derechos de Autor y no puede ser copiados, ni reproducidos, ni distribuidos por personas o Entidades diferentes al DNP

Tabla de contenido

1	INTRODUCCION	4
2	OBJETIVO	4
3	ALCANCE	4
4	TERMINOS Y DEFINICIONES	5
5	USO DE PRUEBAS DE RENDIMIENTO	6
6	HERRAMIENTAS.....	6
7	JMETER.....	6
7.1	Instalación	6
7.1.1	Descargue el archivo .zip.....	7
7.1.2	Requisitos	7
7.1.3	Descomprima el archivo.....	7
7.1.4	Carpeta bin	7
7.2	Componentes de Jmeter	8
7.2.1	Test plan	8
7.3	Ejecución de pruebas.....	12
7.3.1	Grabar secuencias desde jmeter	12
7.3.2	Grabar secuencia usando Chrome blazemeter.....	16
7.4	Análisis de resultados de las pruebas de rendimiento	17
8	REFERENCIA.....	18

1 INTRODUCCION

Las pruebas de rendimiento se encargan de comprobar como una aplicación o un recurso sigue funciona bajo una determinada carga de usuarios o transacciones. Los requisitos de rendimiento deben estar de acuerdo a los contratos de nivel de servicio, algunos requisitos son por ejemplo: el tiempo de respuesta promedio de alguna de las funcionalidades, el número de usuarios constantes en determinado tiempo, una vez se tienen los requerimientos no funcionales relacionados con estos requisitos se deben diseñar las pruebas, ejecutarlas y analizar los resultados.

Ya sea que el desarrollo sea interno o externo, es responsabilidad del líder funcional y/o técnico de la dependencia, o del proveedor externo, que las pruebas de rendimiento se planifiquen como cualquier otra tarea, y es importante que desde el desarrollo de la aplicación, se tenga conocimiento de los requerimientos no funcionales y apliquen sus propias pruebas para verificar el cumplimiento de los tiempos de respuesta de cada uno de los desarrollos, esto con el fin de poder detectar temprano los tiempos de respuesta del aplicativo, demostrar si el sistema cumple con los criterios establecidos, y evitar que sea el usuario final quien detecte este tipo de problemas.

2 OBJETIVO

Se debe asegurar que el sistema de información cumplan los hitos o criterios de aceptación de los requerimientos no funcionales, El objetivo de este tipo de pruebas es determinar cuáles son las transacciones más críticas para una posible optimización de las mismas, detectando posibles cuellos de botella y corrigiendo los mismos para mejorar el rendimiento.

Algunos hitos a tener en cuenta son:

- Criterios de tiempos de respuesta del aplicativo y de ciertas funcionalidades.
- Cantidad de usuarios estimados en operación normal.
- Cantidad máxima de usuarios soportados con el criterio de tiempo de respuesta.
- Requerimientos mínimos de hardware y comunicaciones para cumplir con criterio de tiempo de respuesta.

En general cuando se planeen en este tipo de pruebas se deben incluir las funcionalidades más críticas, más usadas, o más consultadas.

3 ALCANCE

Esta guía tiene como propósito dar una breve descripción de manera general sobre cómo crear, configurar y ejecutar pruebas de carga a un sistema de información usando la herramienta de jmeter.

Las pruebas de rendimiento deben ser aplicadas tanto por funcionarios como por contratistas internos o externos relacionados con el DNP y el resultado satisfactorio de las pruebas debe ser un requisito mínimo para que dar el visto bueno para su paso a producción. Es importante en la planificación de este tipo de pruebas que estén comprometidos los grupos de plataforma, arquitectura y los ingenieros de desarrollo y pruebas del sistema a probar.

Las pruebas en ambiente de producción o ambiente de pruebas, se deben ejecutar una vez se han realizado y aprobado satisfactoriamente las pruebas funcionales y para entregas a producción estas deben ser ejecutadas en servidores del DNP.

Si la aplicación está en producción, lo más correcto es hacer las pruebas de rendimiento en el servidor donde está implementada la aplicación, siempre y cuando no se vea afectada la prestación del servicio, para solucionar esto se pueden acordar tiempos para la ejecución y monitoreo de las pruebas. Si no es posible ejecutarlas en servidores de producción se recomienda ejecutar las pruebas de carga en un ambiente lo más parecido a producción.

Generalmente cuando se ejecutan las pruebas de rendimiento se debe tener en cuenta:

- Monitoreo constante de los servidores afectados ya que se debe observar que bajo el Número de usuarios concurrentes no existen consumos excesivos de CPU, memoria y conexiones a Base de Datos.
- Informe de resultados de la herramienta con la que se hizo las pruebas.
- Tener evidencia de cómo se comportó el sistema.
- Interactuar con el aplicativo con pruebas manuales y observar los tiempos de respuesta en cada uno de los procesos.
- finalmente, en la **fase de resultados**, una vez se recopilan los datos de los monitores, evaluar la información y realizar un informe con las conclusiones que hayamos sacado tras el estudio realizado. En dicho informe se debe reflejar aquellos aspectos que influyen de forma negativa en el sistema, de forma que puedan ser corregidos y/o mitigados.

4 TERMINOS Y DEFINICIONES

Jmeter: Es una herramienta de pruebas de rendimiento en donde se registran acciones en un navegador y después puede ejecutarse con varios usuarios.

Tiempo de Respuesta: Es lo que tarda la aplicación en responder a nuestra solicitud, el tiempo de respuesta es el tiempo que tarda el servidor en procesar la solicitud y generar una respuesta al navegador.

Rendimiento: Se puede medir en dos maneras, el número de transacciones, el número de solicitudes.

Confiabilidad: Como la aplicación detecta y maneja los errores.

Escalabilidad: De acuerdo a los resultados de las pruebas, se puede definir si se requiere escalabilidad vertical (hardware CPU, RAM), o escalabilidad horizontal (agregar servidores).

Prueba de Humo: Diseñadas para ejecutar una prueba con poca carga, generalmente 1 usuario para probar que la prueba está bien generada.

Prueba de Carga: Diseñadas para monitorear el comportamiento de la aplicación, en ese caso la carga de la aplicación se va incrementando progresivamente hasta cierto punto. Por ejemplo, se van incrementando la cantidad de usuarios que utilizan la aplicación o la cantidad de transacciones que se realizan. La finalidad de este tipo de prueba es la de determinar la **robustez** de una aplicación cuando la carga es extrema facilitando la configuración de las alarmas del sistema cuando se alcancen ciertos límites.

Prueba de Stress Cuando se aplica la carga más allá de lo normal, con el fin de observar hasta qué punto la aplicación permanece estable y receptiva.

Prueba de Resistencia Diseñadas para que la aplicación se somete a una carga dentro de sus límites en un periodo de tiempo prolongado, horas.

5 USO DE PRUEBAS DE RENDIMIENTO

Una prueba de rendimiento apunta a indagar sobre el comportamiento de una aplicación cuando se genera una alta demanda de uso. Durante la prueba de rendimiento se tienen en cuenta diferentes indicadores como: nivel de respuesta, errores, velocidad, escalabilidad y estabilidad, recursos consumidos, entre otros. Por ejemplo, se quiere determinar el comportamiento de una aplicación cuando ingresan en forma simultánea varios usuarios, o se requiere medir su rendimiento cuando se ejecutan algunas operaciones. Una vez ejecutadas las pruebas al finalizar, se analizan todos los datos generados evaluando el desempeño general de la aplicación.

En caso de encontrar anomalías se debe comenzar un trabajo de investigación para detectar los problemas del sistema. Puede tratarse de aspectos referidos a la aplicación misma, a las Bases de Datos, al sistema operativo, fallas en el hardware, o la infraestructura de la red, se debe también revisar si la aplicación corre en servidores compartidos con otras aplicaciones, etc.

6 HERRAMIENTAS

Al día de hoy se solicita que las pruebas sean ejecutadas con jmeter, esto va de acuerdo a la evolución de las herramientas de pruebas que estén dispuestas en el mercado.

A continuación, se detallan mediante algunos ejemplos, como se puede generar un proyecto de carga, la documentación sobre el uso de la herramienta se encuentra en internet, de todas formas al final del documento se anexan algunos link en español que pueden ayudar a entender la configuración de este tipo de pruebas, pero con el ánimo de dejar una guía sobre el uso de estas herramientas, se documenta esta guía y se muestran una serie de actividades de acuerdo a la versión del software de jmeter que está vigente.

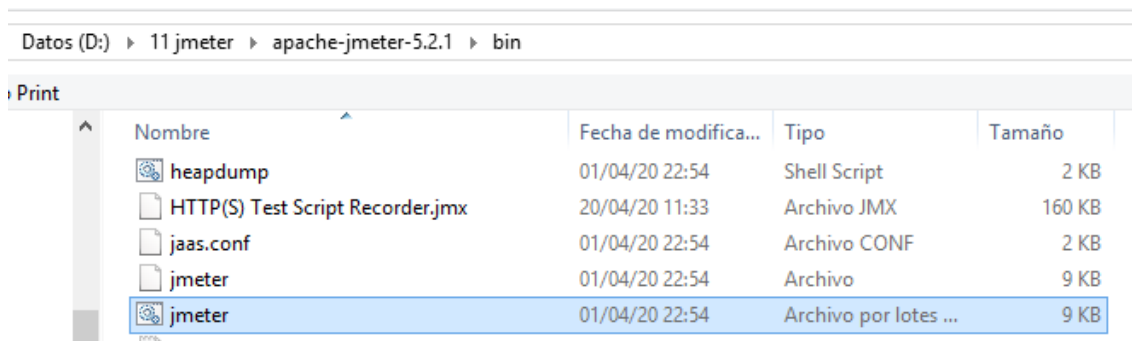
7 JMETER

Apache JMeter está desarrollada 100% en Java y es Open Source. El objetivo es grabar un escenario de pruebas para posteriormente configurarlo con los parámetros como por ejemplo el número de usuarios concurrentes.

A continuación se describen algunos pasos para la creación, ejecución y revisión de resultados.

7.1 Instalación

Para la instalación de jmeter, se descarga el archivo .zip se descomprime el archivo y se ejecuta el archivo jmeter.bat el cual se encuentra dentro de la carpeta bin



7.1.1 Descargue el archivo .zip

En la página de apache, puede descargar el software, en este link puede acceder a las últimas versiones [descarga de Apache JMeter](#).

Binaries

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

Seleccione el archivo .zip para descargar localmente el ejecutable

7.1.2 Requisitos

Dependiendo de la versión que descargó, verifique los [requisitos mínimos de la versión de Java](#)

7.1.3 Descomprima el archivo

Descomprima el archivo .zip en el directorio de su elección. [apache-jmeter-5.2.1.zip sha512 pgp](#)

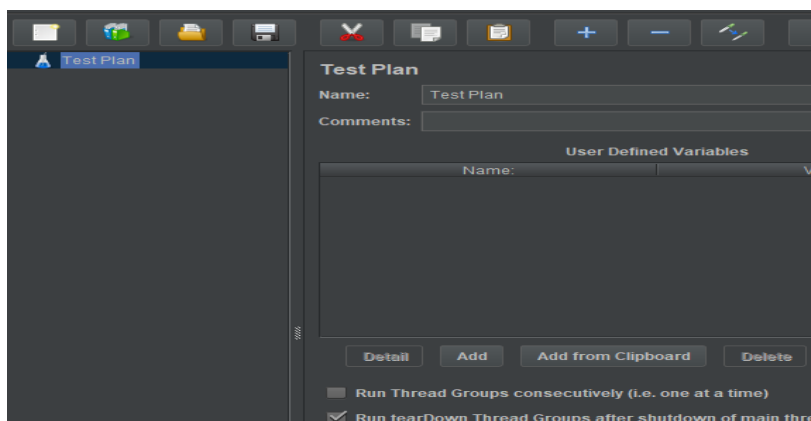
7.1.4 Carpeta bin

Seleccione la carpeta **bin** y seleccione el archivo jmeter.bat

Por ejemplo:

Windows (C:) > desarrollo > jmeter 5.2 > apache-jmeter-5.2.1 > bin				
Nombre	Fecha de modificación	Tipo	Tamaño	
create-mtr-keystore.sh	15/01/2020 12:02 p. m.	Archivo SH	2	
hc.parameters	15/01/2020 12:02 p. m.	Archivo PARAMET...	2	
heapdump.cmd	15/01/2020 12:02 p. m.	Script de comand...	2	
heapdump.sh	15/01/2020 12:02 p. m.	Archivo SH	2	
jaas.conf	15/01/2020 12:02 p. m.	Archivo CONF	2	
jmeter	15/01/2020 12:02 p. m.	Archivo	9	
jmeter.bat	15/01/2020 12:02 p. m.	Archivo por lotes ...	9	

Al ejecutar el archivo jmeter.bat el programa arranca y muestra la pantalla inicial, esta es la interfaz de usuario de JMeter.



7.2 Componentes de Jmeter

A continuación se da una breve explicación sobre los componentes de Jmeter, en general se explicaran los componentes más usados, pero de acuerdo a la complejidad de desarrollo y/o a las funcionalidades que se quieran probar, es responsabilidad de los usuarios técnicos del proyecto investigar sobre los demás componentes para implementarlos en su aplicativo (pruebas de rendimiento, s.f.):

- **Test Plan:** Es el tipo de componente que representa la raíz del árbol.
- **Thread Group:** Representa un grupo de usuarios. En JMeter cada thread es un usuario virtual.
- **Controllers (Sampler, Logic Controller):** Los samplers realizan peticiones contra la aplicación y los logic controllers establecen el orden en que se ejecutan éstos.
- **Https Request.** permite enviar peticiones http hacia un servidor.
- **Config Element.** Establecen propiedades de configuración que se aplican a los samplers a los que afectan.
- **Assertion.** Comprueban condiciones que aplican a las peticiones que realizan contra la aplicación los samplers a los que afectan.
- **Listeners.** Recopilan datos de las peticiones que realizan los samplers a los que afectan.
- **Timer.** Añaden tiempo extra a la ejecución de las peticiones que realizan contra la aplicación los samplers a los que afectan

7.2.1 Test plan

El test plan es el punto de partida del proyecto de pruebas de rendimiento. JMeter guarda el plan de prueba en formato .jmx.

Para agregar componentes al plan de prueba, haga clic con el botón derecho en el grupo de prueba y navegue hasta el componente que desea agregar.

JMeter se basa en el concepto de “Elemento” y en una estructura en “Árbol”. Cualquier parte o rama del árbol puede ser guardada de forma independiente, para ser reutilizada en otras pruebas. Los elementos nos van a permitir configurar y definir el plan de pruebas:

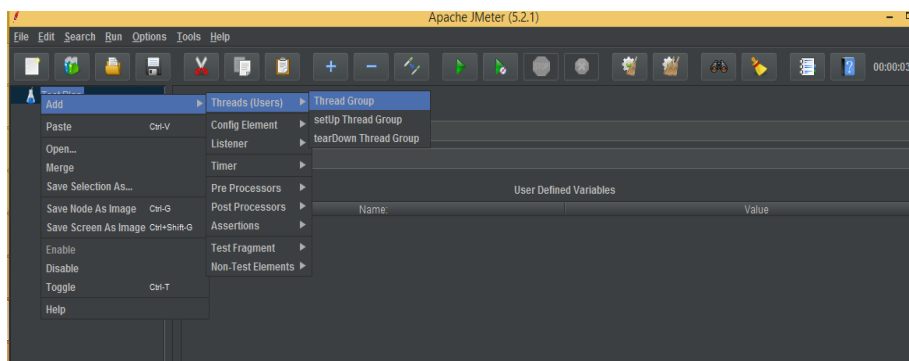
- Elementos jerárquicos:
 - Listeners (elementos en escucha).
 - Config Elements (elementos de configuración).
 - Assertions (afirmaciones).
 - Timers (cronómetros).
- Elementos ordenados:

- Controllers (controladores)
- Samplers (agentes de pruebas).

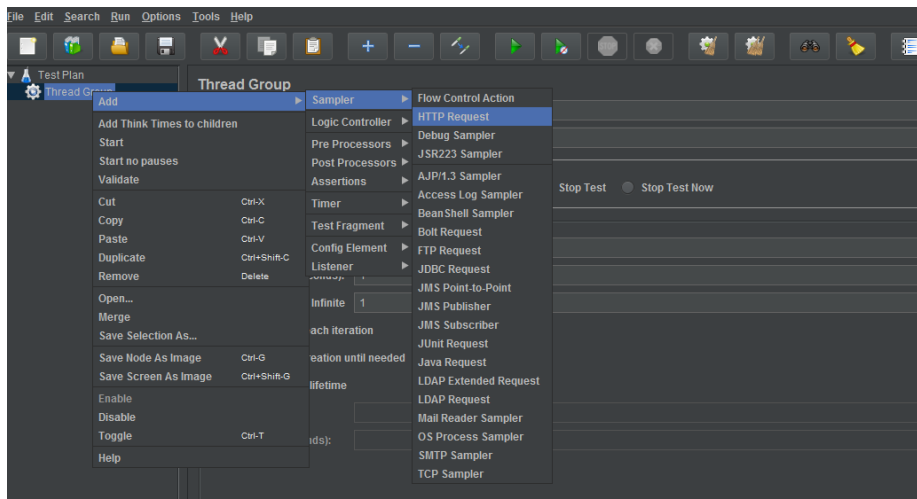
7.2.1.1 Ejemplo usando Peticiones HTTPS

Con esta opción se pueden realizar peticiones a url a sitios, sin tener que modificar el proxy

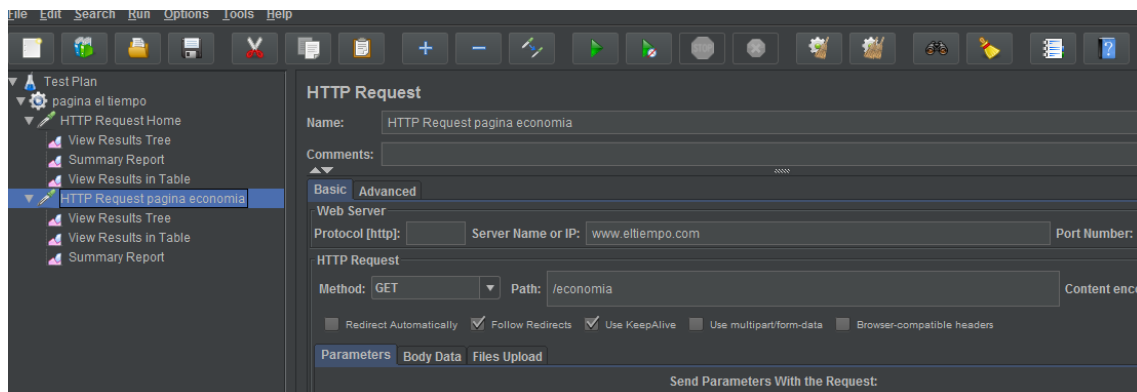
- Lo primero es crear los grupo de hilos, clic derecho en test plan y adicionar thread group



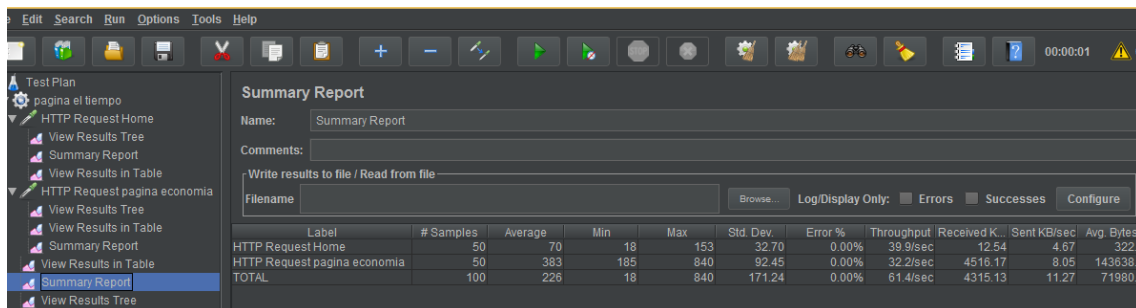
- Se puede adicionar al grupo de hilos las peticiones https



- Se pueden crear varias peticiones a diferentes funcionalidades, en este ejemplo se crea una petición al servidor del tiempo y otra petición a página de economía del tiempo.

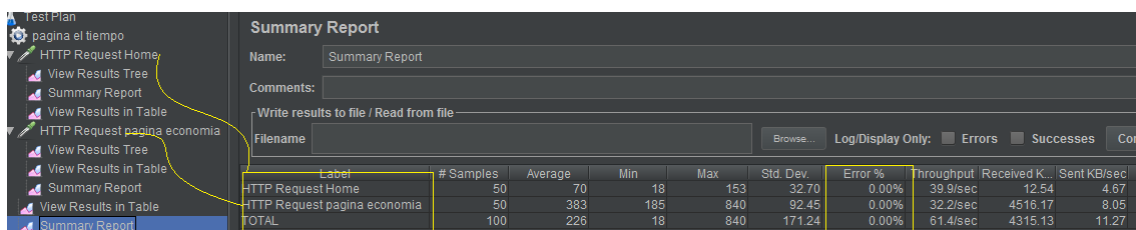


Una vez se adicionan los listener (componentes que permiten procesar la información capturada por JMeter), se ejecuta la prueba y se obtienen los resultados de la ejecución.



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received K...	Sent KB/sec	Avg. Bytes
HTTP Request Home	50	70	18	153	32.70	0.00%	39.9/sec	12.54	4.67	322.0
HTTP Request pagina economia	50	383	185	840	92.45	0.00%	32.2/sec	4516.17	8.05	143638.9
TOTAL	100	226	18	840	171.24	0.00%	61.4/sec	4315.13	11.27	71980.5

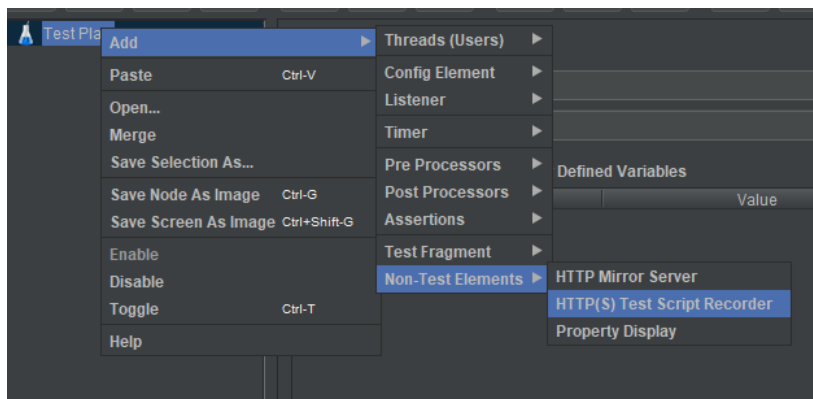
En el listener de “summary report” se pueden observar los resultados de la ejecución, es importante revisar que no se presenten errores en la ejecución de las pruebas. Para el ejemplo se enviaran peticiones para 50 usuarios



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received K...	Sent KB/sec	Avg. Bytes
HTTP Request Home	50	70	18	153	32.70	0.00%	39.9/sec	12.54	4.67	322.0
HTTP Request pagina economia	50	383	185	840	92.45	0.00%	32.2/sec	4516.17	8.05	143638.9
TOTAL	100	226	18	840	171.24	0.00%	61.4/sec	4315.13	11.27	71980.5

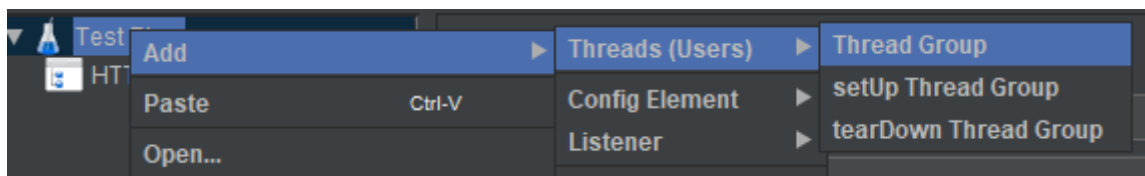
7.2.1.2 HTTP(s) Test Script Recorder

Este componente se usa para grabar las acciones del navegador usando un servidor proxy.

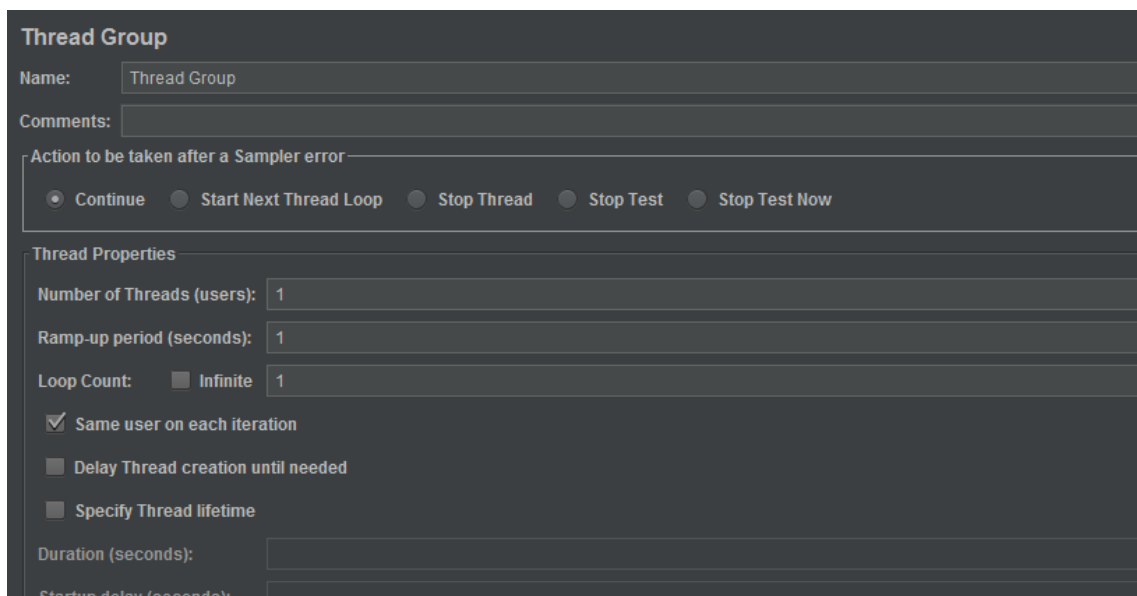


7.2.1.3 Configuración de Threads (usuarios)

Los threads representan un grupo de usuarios. En JMeter cada thread es un usuario virtual. Este tipo de componente permite representar grupos de usuarios



A continuación se explican algunos conceptos de las propiedades del thread



Un Thread Group: Es el punto de partida de cualquier plan de prueba de Jmeter. Es la parte más alta del árbol y todos los elementos de un plan de prueba deben definirse debajo de él.

Number of Thread : Define el número de usuarios que desea simular para la ejecución. En Jmeter, cada thread es un usuario virtual que abre la conexión a nuestro back-end y comienza a ejecutar solicitudes.

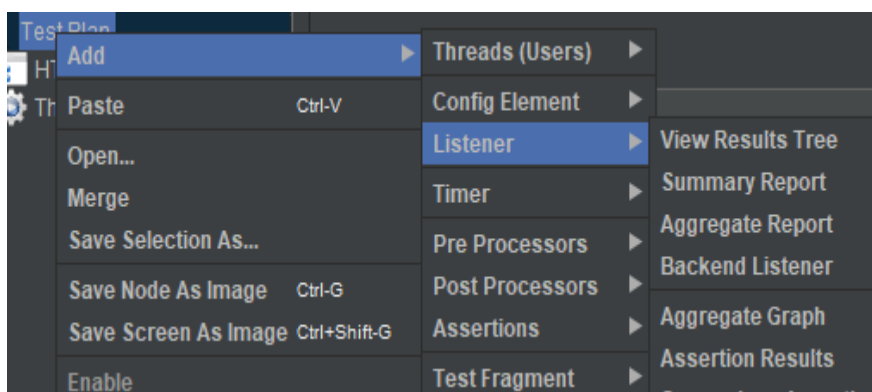
Ramp –up periodo : La aceleración es la cantidad de tiempo que Jmeter debería tomar para obtener todos los hilos enviados para la ejecución. Por ejemplo, si el número de threads es 10 y el ramp-up period es de 100 segundos, Jmeter tardará 100 segundos en poner en funcionamiento los 10 threads. El primer thread se enviará el segundo 0 y luego cada thread se ejecuta después de 10 segundos (100/10)

7.2.1.4 Listener: (oyentes)

Estos componentes se encargan de recopilar datos de las peticiones que realizan los samplers a los que afectan. Los oyentes le permiten guardar los resultados de la prueba, ver la ejecución de la prueba, etc. Por ejemplo:

View Results Tree: Puede ver la solicitud / respuesta de los muestreadores y si están marcados como PASS (color verde) / FAIL (color rojo) por JMeter.

Informe agregado: Puede guardar los resultados de la prueba en formato CSV. Debemos eliminar los escuchas durante la prueba, ya que consume una gran cantidad de recursos del sistema.



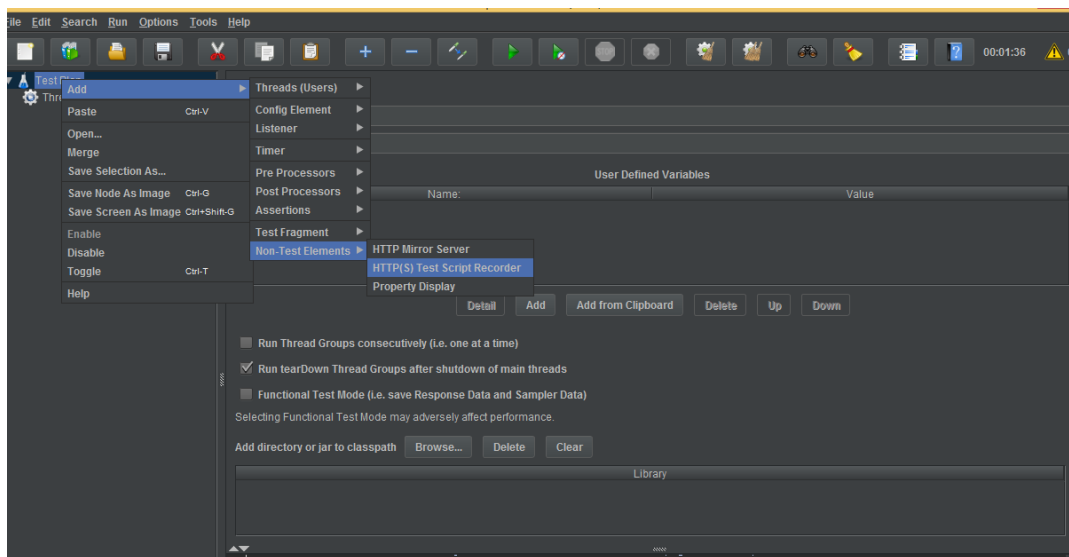
7.3 Ejecución de pruebas

A continuación se explican dos formas para captura de secuencias

7.3.1 Grabar secuencias desde jmeter

Abrir la aplicación, configurarla para indicar qué tipo de pruebas queremos realizar, para este ejemplo, realizaremos una prueba a la página principal de “El Tiempo”. Para ello,

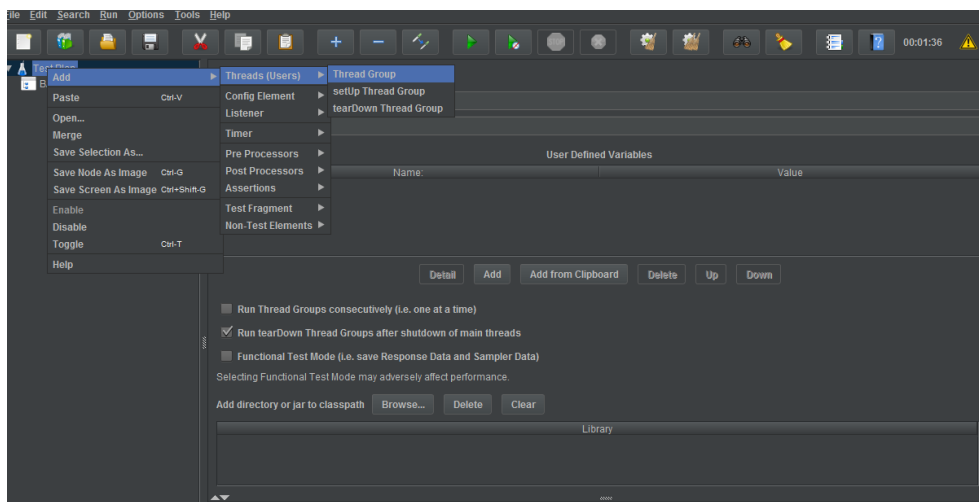
- Agregar un **servidor proxy** al **test plan**



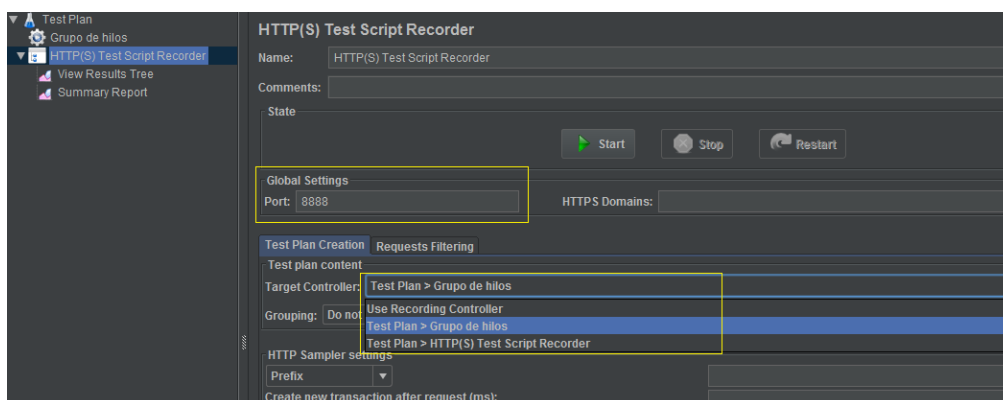
- **Configurar el navegador Mozilla Firefox:** Ir al menú herramientas y pulsar en Opciones. El navegador mostrará una ventana en la que tendremos que seleccionar la pestaña Avanzado. Una vez allí, pulsar el botón Configuración que se encuentra dentro de la pestaña Red. En esta nueva ventana le indicaremos al navegador que utilice el servidor proxy creado en Apache JMeter.



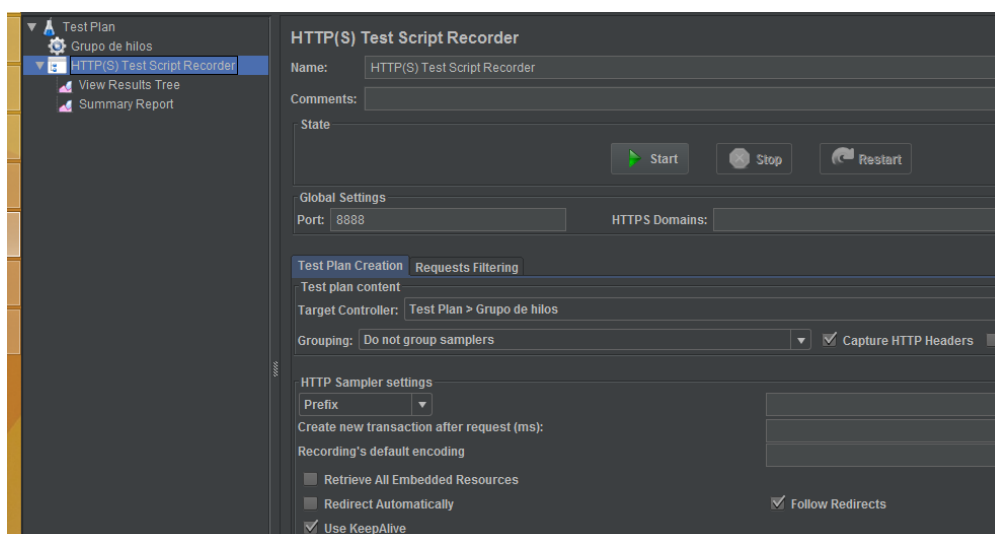
- A continuación hay que crear un **grupo de hilos** en el cual se almacenarán todas las peticiones detectadas por el servidor proxy.



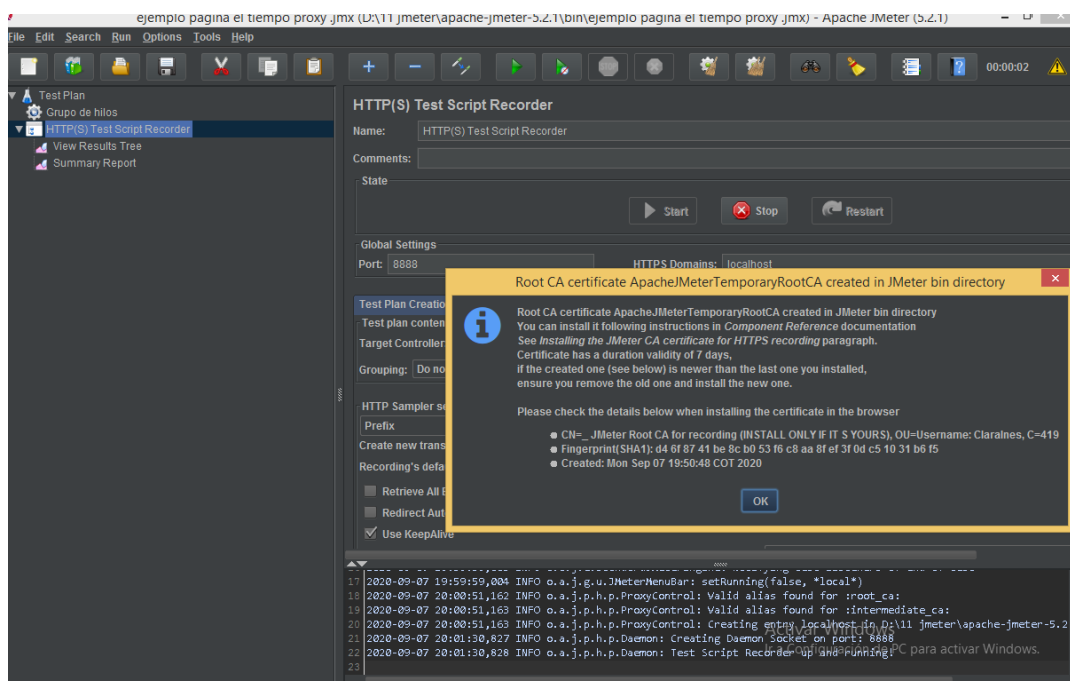
- Una vez creado el grupo de hilos hay que configurar el servidor proxy para que almacene las peticiones en él.



- Se pueden adicionar los listeners para ver los resultados.

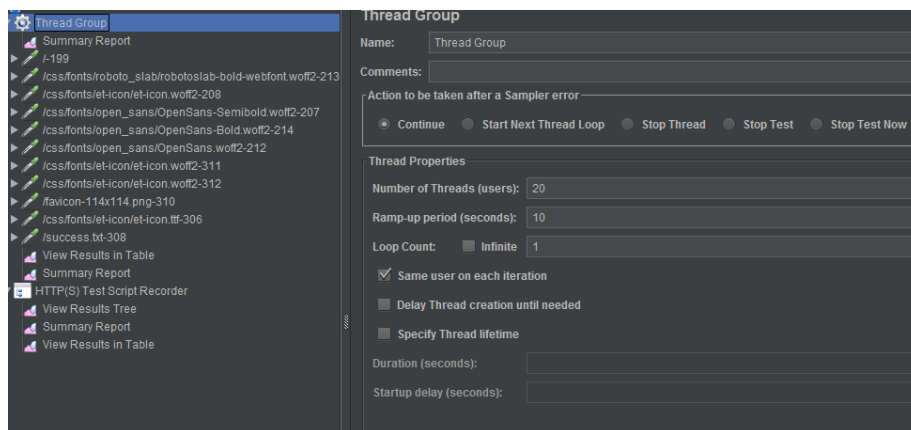


- Iniciar la ejecución en el navegador mozilla, Tras iniciar el servidor proxy se guardarán en el grupo de hilos creado todas las peticiones que realice el navegador, por lo que accedemos a la aplicación (en nuestro caso sólo para el ejemplo vamos a acceder a la página de inicio de www.eltiempo.com), una vez cargada la aplicación, el elemento Grupo de Hilos tendrá una lista de peticiones, que corresponden con las peticiones que realiza la página de inicio de la aplicación
- Haga clic en el botón 'Inicio', que se encuentra en la parte inferior de la página "HTTP(S) Test Script Recorder", y vaya a través del flujo de trabajo de la aplicación web que desea probar.
- Cuando regrese a JMeter, debería ver todas las solicitudes capturadas desde su navegador bajo el "Controlador de grabación"

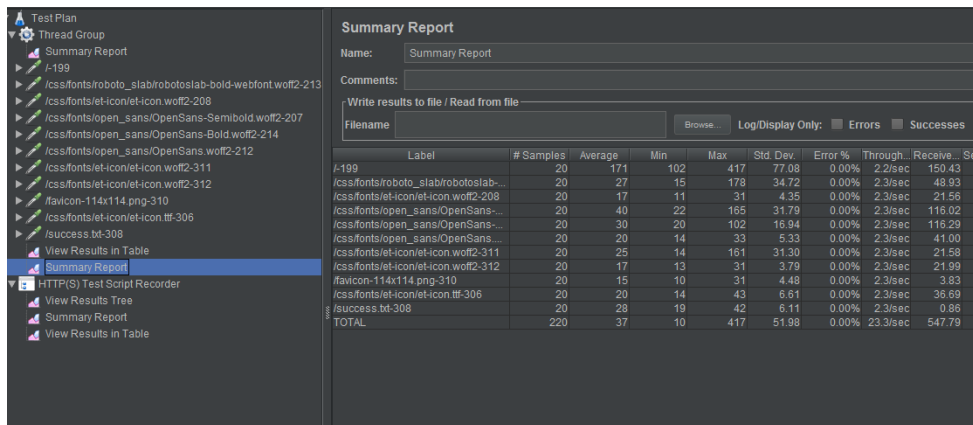


Las peticiones se van guardando en nuestro **Test Plan**

- Se observa como en jmeter se tiene la grabación de las peticiones y se configura la cantidad de usuarios que se quieren para la prueba, por ejemplo , se configuran 20 usuarios con un periodo de carga de 10 segundos



Salida de datos



Summary Report										
Name: Summary Report										
Comments:										
Write results to file / Read from file										
Filename										
/-199	20	171	102	417	77.08	0.00%	2.2/sec	150.43		
/css/fonts/roboto_slab/roboto_slab-bold-webfont.woff2-213	20	27	15	178	34.72	0.00%	2.3/sec	48.93		
/css/fonts/et-icon/et-icon.woff2-208	20	17	11	31	4.35	0.00%	2.3/sec	21.56		
/css/fonts/open_sans/OpenSans-Semibold.woff2-207	20	40	22	165	31.79	0.00%	2.3/sec	116.02		
/css/fonts/open_sans/OpenSans-Bold.woff2-214	20	30	20	102	16.94	0.00%	2.3/sec	116.29		
/css/fonts/open_sans/OpenSans.woff2-212	20	20	14	33	5.33	0.00%	2.3/sec	41.00		
/css/fonts/et-icon/et-icon.woff2-311	20	25	14	161	31.30	0.00%	2.3/sec	21.58		
/css/fonts/et-icon/et-icon.woff2-312	20	17	13	31	3.79	0.00%	2.3/sec	21.99		
/favicon-114x114.png-310	20	15	10	31	4.48	0.00%	2.3/sec	3.83		
/success.bt-308	20	20	14	43	6.61	0.00%	2.3/sec	36.69		
/success.bt-308	20	28	19	42	6.11	0.00%	2.3/sec	0.86		
TOTAL	220	37	10	417	51.98	0.00%	23.3/sec	547.79		

7.3.2 Grabar secuencia usando Chrome blazemeter

Hasta ahora hemos cubierto las formas básicas de registrar escenarios de prueba. Pero una de las maneras más rápidas y fáciles de grabar sus scripts de rendimiento, que también es gratis, es usar la extensión [Chrome de BlazeMeter Recorder](#). Estas grabaciones se pueden ejecutar en JMeter o en BlazeMeter.

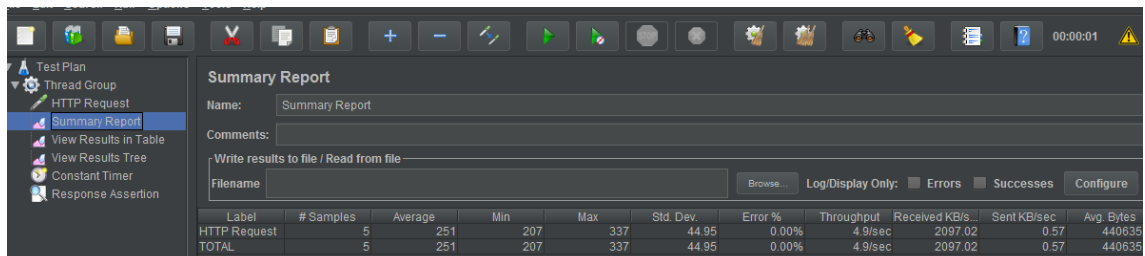
La razón por la que la extensión es tan útil es que le permite grabar scripts de rendimiento desde su navegador sin tener que configurar su proxy.

A continuación se explican los pasos para crear una nueva secuencia de comandos de rendimiento usando blazemeter

- Abrir la grabadora desde el navegador Chrome
- Ingresar un nombre de prueba en el campo superior
- Comience a grabar haciendo clic en el botón de grabación, y realice las acciones web que desea grabar. Todas sus peticiones serán capturadas. Blazemeter Chrome Extensión también admite la grabación de tráfico HTTPS.
- Después de terminar de grabar, haga clic en el botón de detener, en forma de un cuadrado. También puede pausar su grabación y luego reanudarla, así como editarla, en formato .jmx o JSON.
- Exporte la grabación: para ejecutar la prueba en JMeter, exporte al formato .jmx haciendo clic en el botón .jmx



7.4 Análisis de resultados de las pruebas de rendimiento



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/s	Sent KB/sec	Avg. Bytes
HTTP Request	5	251	207	337	44.95	0.00%	4.9/sec	2097.02	0.57	440635
TOTAL	5	251	207	337	44.95	0.00%	4.9/sec	2097.02	0.57	440635

Los resultados de la ejecución de las pruebas se pueden observar en los 'Listeners' que añadimos anteriormente.

Después de ejecutar la Prueba, haga clic en el elemento "Summary Report (Reporte Resumen)" para consultar los resultados.

Los datos obtenidos se interpretan de la siguiente manera:

- **# Samples / # Muestras:** Indica el número total de threads que se ejecutaron en el intervalo de tiempo indicado en el "Thread Group".
- **Average / Media:** Indica el tiempo promedio de respuesta en Ms (milisegundos) de todos los Threads ejecutados en el intervalo de tiempo, pertenecientes al "ThreadGroup".
- **Min:** Indica el tiempo de respuesta (milisegundos) más corto de todo el grupo de threads ejecutado.
- **Max:** Indica el tiempo de respuesta (milisegundos) más largo de todo el grupo de threads ejecutado.

- **Std.Derivation / Desviación Estándar:** Indica la distancia promedio que hay entre los diferentes tiempos de respuesta de todos los Threads ejecutados respecto al tiempo promedio. Una menor distancia indica que los Threads se respondieron relativamente en el mismo tiempo promedio, lo cual indica que los servicios se ofrecen en las mismas condiciones para todos).
- **Error %:** Indica la proporción de Threads cuyas peticiones no fueron atendidas por el servidor ya que éste estaba ocupado o indisponible.
- **Throughput / Rendimiento:** Indica el número de threads ejecutado por segundo.
- **KB/Sec:** Número de Kilobytes por segundo enviados al servidor.
- **Avg Bytes:** Promedio de Kilobytes por segundo enviados al servidor.
- **Latencia:** tiempo que se tarda en hacer la petición al servidor y lo que se demora el servidor en responder, el tiempo siempre el tiempo es más baja que el tiempo de muestra
- **Tiempo de muestra:** Es el tiempo desde que se hizo la petición hasta que termine y se obtenga respuesta.

Teniendo en cuenta la información anterior, es posible definir la capacidad máxima del software, para esto hay que tener en cuenta los requerimientos de calidad. Por ejemplo si en la definición de requerimientos de rendimiento se estableció que:

- El tiempo de respuesta para todos los usuarios debe ser menor a 1 segundo y Todas las peticiones deben ser respondidas por el servidor.

Se puede verificar estos requerimientos en los resultados de la prueba, y en adelante comenzar a repetir esta prueba variando la cantidad de usuarios en el mismo límite de tiempo hasta encontrar la cantidad que no cumpla los umbrales.

8 REFERENCIA

Internet 1 Cómo hacer pruebas de rendimiento a un servicio web con Apache JMeter - Idioma : Español https://www.youtube.com/watch?v=KpqeqIA7yS0&feature=emb_rel_pause

Internet 2 Prueba del rendimiento de una aplicación web mediante Azure Portal - Idioma : Español https://www.youtube.com/watch?v=KpqeqIA7yS0&feature=emb_rel_pause

Internet 3 Empezando con Apache Jmeter - Idioma : Español <https://riptutorial.com/es/jmeter>

Internet 4 Pruebas de rendimiento con JMeter. Ejemplos básicos Idioma : Español <https://sdos.es/blog/pruebas-de-rendimiento-con-jmeter-ejemplos-basicos>

Internet 5 BlazeMeter + Jmeter for Performance testing <https://folderit.net/itech/es/blazemeter-jmeter-for-performance-testing-es/>

Internet 6 jmeter Beginner Tutorial 6 - Jmeter How to record login test – Idioma ingles <https://www.youtube.com/watch?v=zn1DSUZ6t64>

Internet 7 Aprendiendo JMeter paso a paso Idioma: Español <https://www.youtube.com/watch?v=pxqxCTVtESA>

Internet 8 Aprendiendo JMeter paso a paso Idioma : Español

<https://www.youtube.com/watch?v=pxqxCTVtESA>

Internet 9 introducción a jmeter . conceptos básicos - Idioma : Español

<http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/388>

Internet 10 Install and run JMeter and Taurus testing tools. Idioma ingles

<https://marketplace.visualstudio.com/items?itemName=AlexandreGattiker.jmeter-tasks&ssr=false#overview>

Internet 11 curso en español de jmeter Idioma : Español

https://www.youtube.com/watch?v=Erl2EWfE1Is&list=PLWeJLT_CmHoyvD6U9OTHW9ywWo4aQA0YL&index=1

Internet 12 conceptos en español de jmeter Idioma : Español

<https://www.blazemeter.com/blog/csv-data-set-config-in-sharing-mode-made-easy>

Internet 13 Monitorización básica de tus servidores desde JMeter.

<https://jmeterenespanol.org/blog/2020-04-01-monitoring-delvis/> - <https://blog.intive-fdv.com.ar/testing-distribuido-con-jmeter-y-aurus/> - <https://www.youtube.com/watch?v=T19My7nDM6M>