

PARTE 2.- Introducción a Apache JMeter

1. Introducción

JMeter¹ es un programa Java diseñado para hacer pruebas de carga de servidores, por ejemplo servidores web. Utilizaremos este software para evaluar experimentalmente las prestaciones de la configuración de Apache presentada en la parte anterior de esta práctica.

2. Ejecutando JMeter en el laboratorio

Desde un terminal puede ejecutar jmeter por ejemplo de la siguiente forma:

```
$ /usr/bin/jmeter
```

Si lo desea, el menú “Options” da la posibilidad de elegir un idioma. Es probable que no todo esté traducido al castellano pero sí bastante. De todas formas, también es recomendable acostumbrarse a usar software en inglés.

Por causas desconocidas, probablemente relacionadas con la instalación de Java en las máquinas del laboratorio, JMeter muchas veces da un error al lanzarse. Reintenten con el comando anterior hasta lograr que arranque.

3. Conceptos básicos sobre JMeter

a) Creando un plan de pruebas con un grupo de hilos

En la Ilustración 1 se puede ver la ventana de JMeter al lanzarlo. El elemento principal en JMeter es lo que se llama un “Plan de Pruebas” o “Test Plan”¹. El plan de pruebas contendrá todos los elementos que definirán cómo se hacen peticiones a un/os servidor/es.

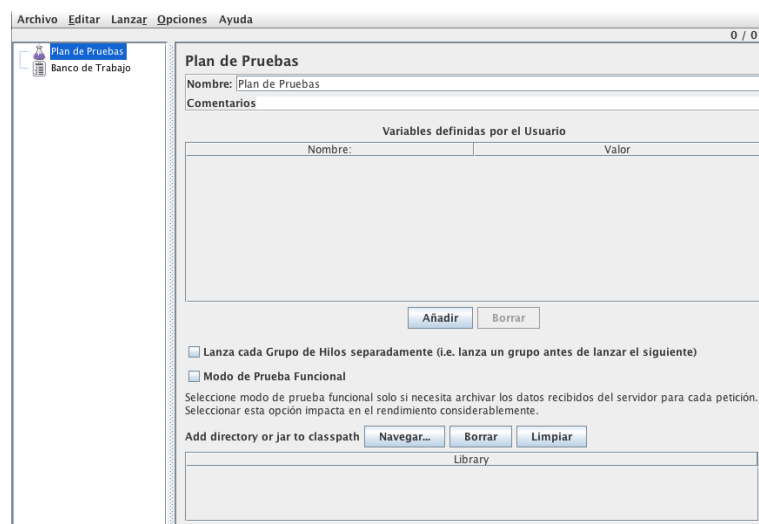


Ilustración 1 - Primera ventana de JMeter

Dentro de ese plan de pruebas añadiremos en primer lugar lo que se llama un “Grupo de hilos” o “Thread Group” mediante el menú que se despliega al pulsar el botón derecho del ratón sobre el plan de pruebas (Ilustración 2).

¹<http://jakarta.apache.org/jmeter/>

¹http://jakarta.apache.org/jmeter/usermanual/test_plan.html



Ilustración 2 - Añadiendo un grupo de hilos

El grupo de hilos indica a JMeter que tendrá que lanzar un número de hilos (“threads”) y qué deberán hacer estos hilos. Todos ellos van a hacer lo mismo y será lo que se indique mediante elementos dentro del grupo de hilos. Lo que se pretende es que uno cualquiera de esos hilos emule el funcionamiento de un usuario o cliente web el cual estaría haciendo peticiones al servidor web, intercaladas con tiempos de descanso en los que se supone el usuario lee/procesa la página descargada antes de pedir la siguiente. Con el grupo de hilos controlaremos cuántos usuarios simultáneos está emulando JMeter (campo “Número de Hilos” en la Ilustración 3).



Ilustración 3 - Propiedades del grupo de hilos

El campo “Periodo de Subida” o “Ramp-up period” indica a JMeter que no lance todos los hilos inmediatamente al iniciar el experimento sino que espacie dicho arranque durante ese tiempo, de forma que sea más improbable que nos encontremos con que nada más empezar el experimento todos los clientes emulados coincidan a pedir una página.

Finalmente, el campo “Contador del bucle” indica cuántas peticiones (o iteraciones) debe hacer cada hilo. Si en cada iteración hay configurado por ejemplo una petición al servidor web seguida de un descanso y aquí indicamos 10, entonces cada hilo hará 10 peticiones intercaladas con otros tantos descansos.

En cualquier momento es recomendable seleccionar el plan de pruebas y guardarlo mediante la opción correspondiente del menú “Archivo”.

b) Elementos básicos en el plan de pruebas

Los elementos descendientes del grupo de hilos que nos van a servir para indicar qué peticiones debe hacer un cliente son los “Muestreadores” o “Samplers”, en concreto vamos a añadir un elemento de tipo “Petición HTTP” o “HTTP Request” (Ilustración 4).

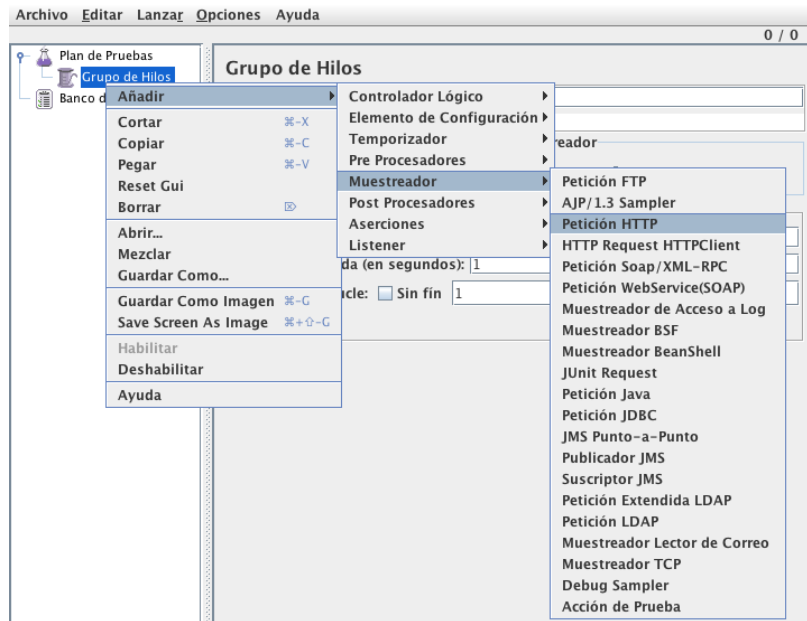


Ilustración 4 - Insertando elemento de petición HTTP

En la Ilustración 5 se ve que hemos indicado el servidor y puerto de nuestro servidor web (asumiendo que corremos Apache HTTPD y JMeter en la misma máquina) así como el camino (Path) al documento a pedir. Esto sería equivalente a pedir en el navegador web: <http://localhost:8087/prueba.html>

En nuestro caso, al estar corriendo Apache HTTPD en la máquina virtual, habrá que indicar la IP de la máquina virtual, una vez la hayamos arrancado.

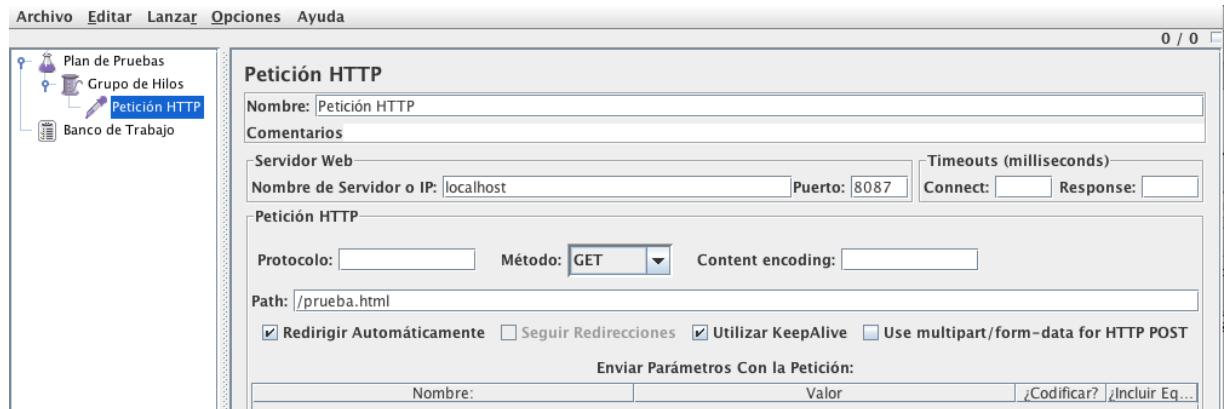


Ilustración 5 - Configurando la petición HTTP de prueba

Para ver los resultados de cada petición añadimos a continuación un “Listener”. Para ello seleccionamos de nuevo el Grupo de Hilos en la columna izquierda y navegamos en el menú desplegable hasta añadir un “Listener” de tipo “Ver Árbol de Resultados” (Ilustración 6).

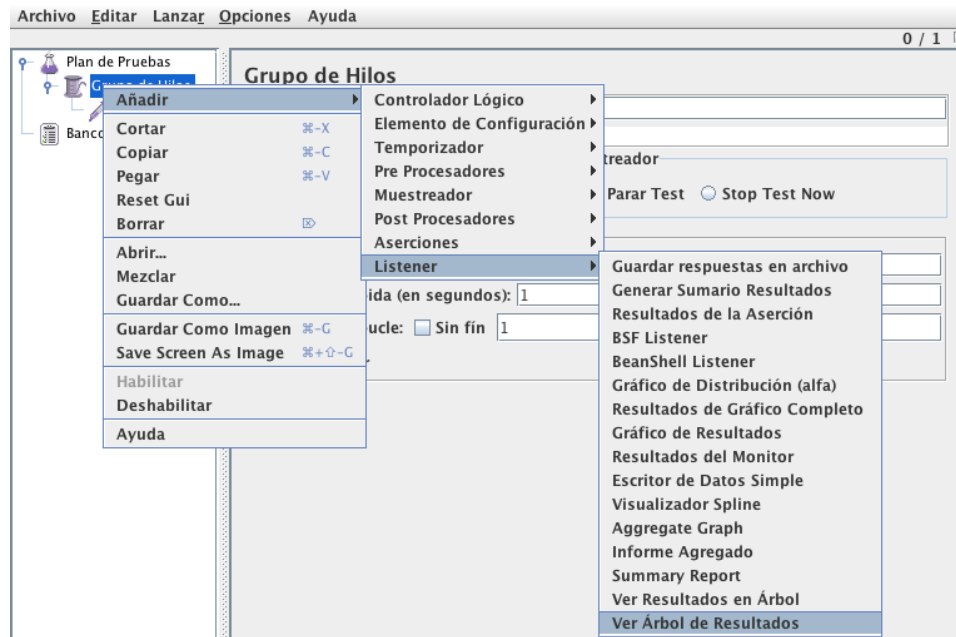


Ilustración 6 - Añadiendo un Listener

Podemos poner en marcha ahora el experimento mediante la opción “Arrancar” del menú “Lanzar”. Como tenemos configurado que se lance un solo hilo y haga una sola iteración obtendremos algo similar a la Ilustración 7.

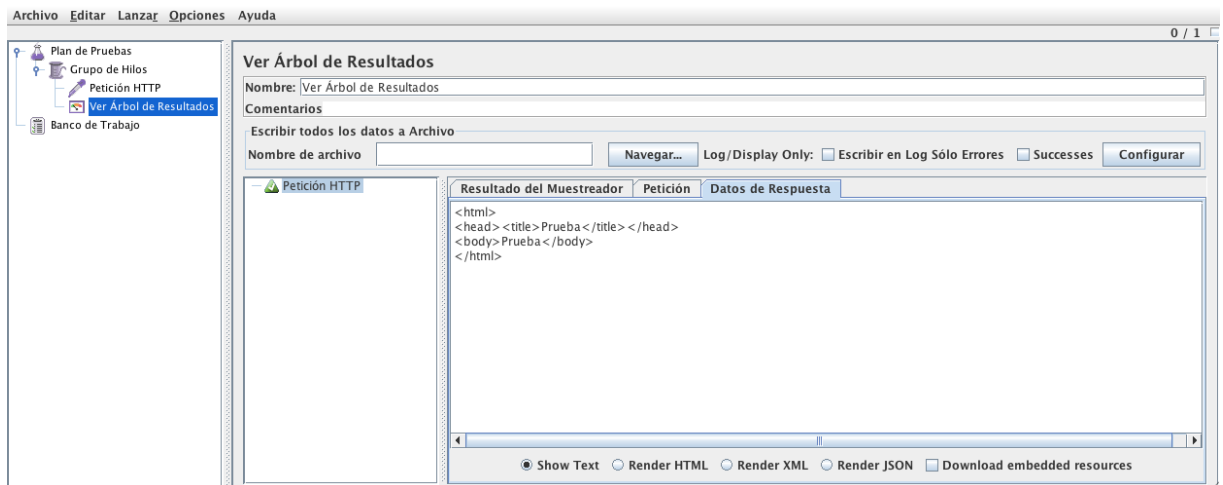


Ilustración 7 - Resultado de una petición

Vea en el fichero *access_log* de apache que ha recibido esta petición.

Si se indica un número mayor de 1 en el “Contador del bucle” del Grupo de Hilos, al terminar de recibir la respuesta a la petición HTTP y por lo tanto terminar la iteración pasaría el hilo a hacer una nueva petición. Esto no tiene en cuenta el tiempo que el usuario pasa procesando el recurso obtenido (o sea, leyendo la página web por ejemplo). Para tener en cuenta este punto incluiremos un “Temporizador” o “Timer”, añadiéndolo al grupo de hilos mediante su menú desplegable. Seleccionamos por ejemplo un temporizador constante (Ilustración 8).

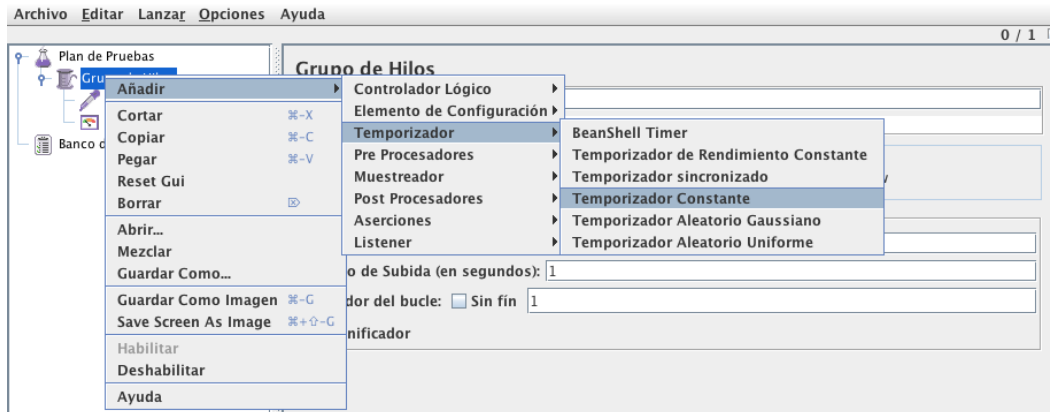


Ilustración 8 - Añadiendo un temporizador

En los parámetros de dicho temporizador se puede indicar la cantidad de tiempo (en milisegundos) que debe esperar el hilo (Ilustración 9).

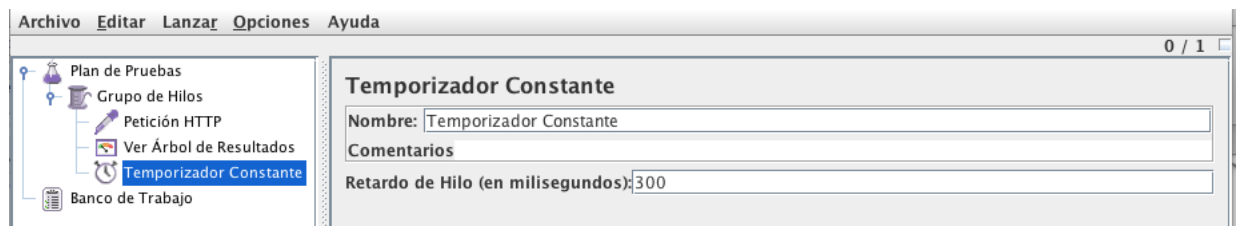


Ilustración 9 - Parámetro de retardo en temporizador constante

Hay que tener en cuenta que independientemente del orden en que estén colocados los elementos dentro del grupo de hilos, en cada iteración todos los temporizadores se procesan antes que los muestreadores. Es decir, en el ejemplo anterior, en cada iteración primero se bloquearía el hilo durante 300ms y a continuación haría la petición HTTP. Al terminar de recibir la respuesta del servidor habría concluido la iteración y si quedaban más iteraciones por llevar a cabo comenzaría de nuevo con el bloqueo de 300ms.

Aumente la pausa del temporizador a 2 segundos y el número de iteraciones a 5 (manteniendo un solo hilo) y compruebe que funciona como se ha descrito.

Con la configuración anterior aumente a 3 el número de hilos, con un periodo de subida de 3 segundos (de forma que el lanzamiento de los hilos estará espaciado en ese tiempo). Verifique con el log del servidor que el funcionamiento es el esperado.

Cambiamos a continuación que el elemento de petición HTTP en vez de pedir */prueba.html* solicite */randwait.php*. Para pasarle los parámetros al mismo podemos utilizar el botón de “Añadir” y rellenar “Nombre” y “Valor” de cada parámetro (Ilustración 10). También pueden poner el URL incluyendo los parámetros en al campo “Path” igual que lo harían en un navegador.

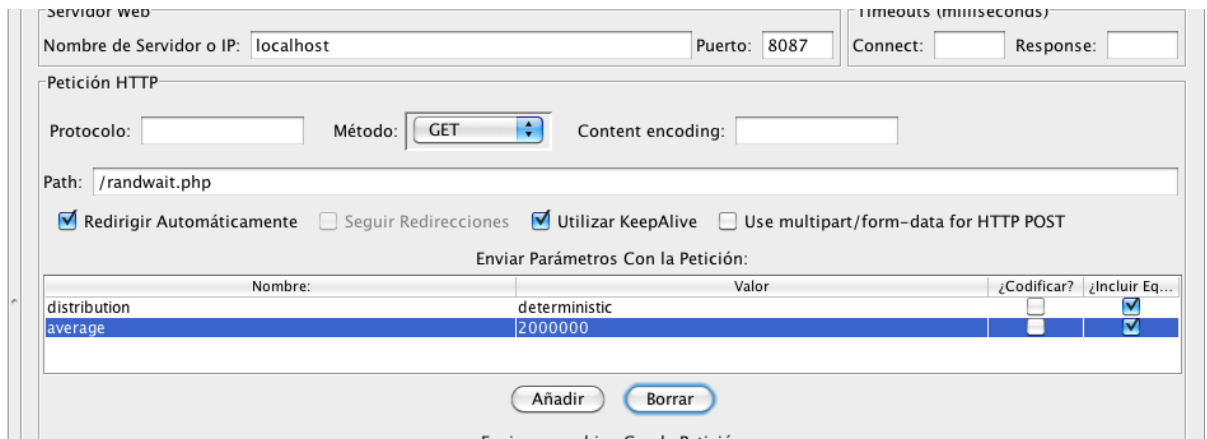


Ilustración 10 - Parámetros en la petición HTTP

Haga que JMeter pida a randwait.php una espera exponencial de media 2.5s.

Asociado al grupo de hilos puede haber varios muestreadores y varios temporizadores. Por ejemplo, podríamos añadir a continuación un temporizador de tipo aleatorio uniforme. Este temporizador lleva a que el hilo, en cada iteración, se bloquee un tiempo aleatorio que proviene de una distribución uniforme. De esta forma, ahora, en cada iteración, el hilo se bloquearía un tiempo determinista dado por el temporizador constante y un tiempo aleatorio uniforme dado por este temporizador; a continuación pasaría a ejecutar los muestreadores (en este ejemplo solo el de petición HTTP) y con ello completaría la iteración.

Cree un plan de pruebas en el que 2 hilos hagan peticiones a randwait.php indicando una espera del servidor en el servicio exponencial de media 1.5s con tiempos de espera del cliente entre peticiones según un timer uniforme entre 0 y 2s.

Emplee wireshark o tcpdump para calcular el tiempo que aproximadamente tarda en establecerse la conexión TCP entre el cliente (JMeter) y el servidor (HTTPD). Una estimación de este valor nos será útil posteriormente.

c) Captura a fichero

En prácticamente cualquier Listener se puede especificar un nombre de fichero en el que guardar los resultados. Todos los Listeners ofrecen las mismas opciones respecto a lo que guardar en ese fichero (botón “Configurar”). Por defecto guardan los resultados en formato XML, desmarque la opción “Guardar como XML” para que lo guarde en formato CSV (*Comma Separated Values*) (Ilustración 11).

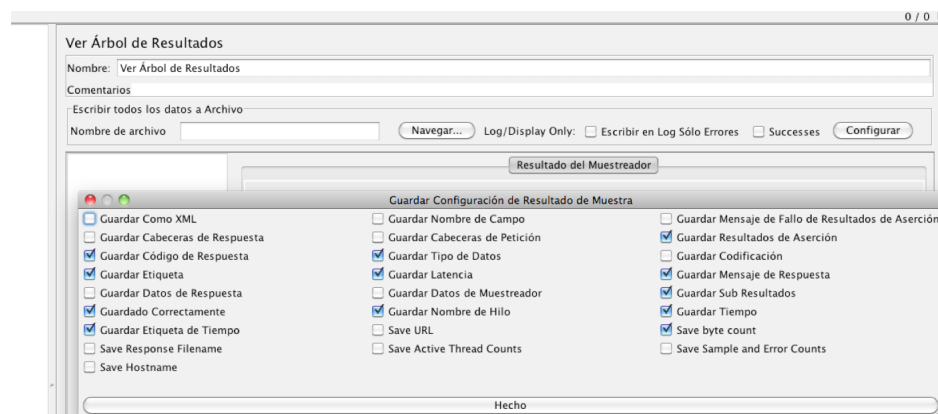


Ilustración 11 - Opciones de datos a fichero

Prueben con diferentes configuraciones el tipo de información que es capaz de guardar JMeter en ese fichero.

d) **Peticiones rechazadas**

Con la configuración llevada a cabo del elemento “Petición HTTP”, cuando se llenan las posiciones de la cola de conexiones, una nueva petición de conexión TCP por parte de JMeter no es aceptada pero como tampoco recibe una notificación de cierre o aborte, la implementación de TCP del cliente reenvía al poco tiempo el mensaje de SYN hasta conseguir establecer la conexión. A partir de ahora queremos que las conexiones que no se completen en el primer intento no se completen mediante retransmisiones sino que directamente se cuente la petición como rechazada por el servidor, de forma que efectivamente el servidor actuaría como un sistema con cola igual al tamaño especificado con la directiva *ListenBacklog*.

Emplearemos para ello el parámetro “Timeout” de “Connect” que se ve en la Ilustración 5. Si transcurre el tiempo especificado ahí sin que se haya terminado de establecer la conexión TCP entonces JMeter abortará la petición HTTP, marcándola como rechazada y pasando al siguiente elemento o iteración. Debemos configurar aquí un valor que sea mayor que el tiempo que suele tardar en establecerse la conexión pero poco más grande de forma que no se dé tiempo a que haya una retransmisión del SYN y pueda establecerse la conexión porque haya quedado libre el servidor o alguna posición de memoria en la cola.

Cree un plan de pruebas en el que se produzcan rechazos de las peticiones y capture en fichero suficiente información como para calcular qué porcentaje de las peticiones son rechazadas.

Pruebe otros “Listeners” como por ejemplo el “Summary Report”.

e) **Log de JMeter**

Si mira el manual del JMeter² verá que no suele sacar pop-ups o avisos gráficos para los errores o avisos de configuración ya que esto podría interferir con el correcto funcionamiento de los tests. Todos estos mensajes los saca a un fichero de logs. Busque en la documentación su PATH.

De todas los avisos hay uno que nos interesa mucho, uno sobre el “SampleResults TimeStamps”.

¿Para qué sirve el “SampleResults TimeStamps”?

¿Qué indica la configuración del JMeter?

Busque en la ayuda del JMeter cómo lanzar el JMeter indicando que el “SampleResults TimeStamps” sea el tiempo de inicio no el final y compruebe que le hace caso.

Por último de la misma manera que al lanzar el JMeter le ha indicado el tipo de timestamp que desea, también se le puede indicar cuantos reintentos HTTP puede hacer por cada petición HTTP.

Busque en la ayuda del JMeter cómo lanzar el JMeter indicando que el número de reintentos sea 0.

Use el plan de prueba que creó en el subapartado d) pero dejando vacío el parámetro “Timeout” de “Connect. ¿Obtiene resultados parecidos en el porcentaje de pérdidas?

²<http://jakarta.apache.org/jmeter/usermanual/get-started.html>