

# P10-Pruebas de aceptación con JMeter

## Pruebas de aceptación de aplicaciones Web

Vamos a automatizar pruebas de aceptación de pruebas emergentes no funcionales para una aplicación Web, concretamente, la aplicación JPetStore de prácticas anteriores. Utilizaremos la herramienta JMeter, con la que crearemos los drivers, a partir de unos escenarios de prueba que previamente hemos diseñado. nuestros tests utilizando java y Junit, que nos servirán para evaluar si nuestro sistema soporta una determinada carga de usuarios (pruebas de carga).

## Bitbucket

---

El trabajo de esta sesión también debes subirlo a Bitbucket. Todo el trabajo de esta práctica, tanto los proyectos maven que vamos a crear, como cualquier otro documento con vuestras notas de trabajo, deberán estar en la carpeta **P10** de vuestro repositorio.

## Ejercicios

---

Recuerda que para poder realizar las pruebas previamente tiene que estar desplegada aplicación a probar (aplicación web JpetStore) en el servidor de aplicaciones Glassfish, de forma que está accesible desde la URL:

<http://localhost:8080/jpetstore/actions/Catalog.action>

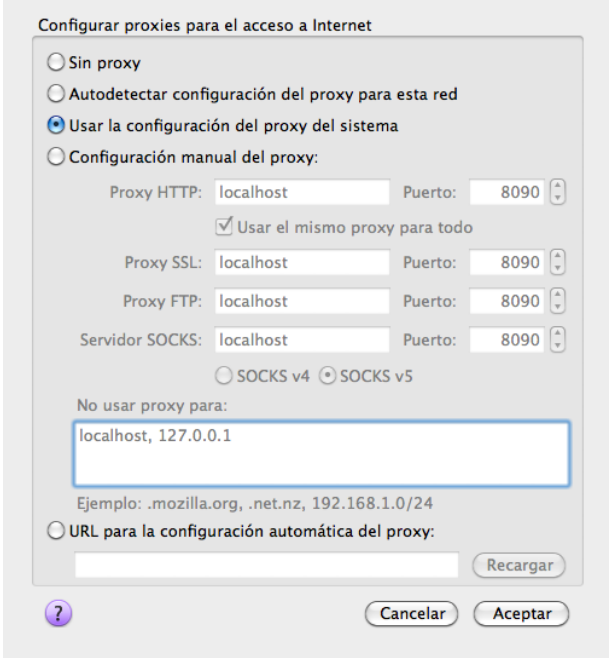
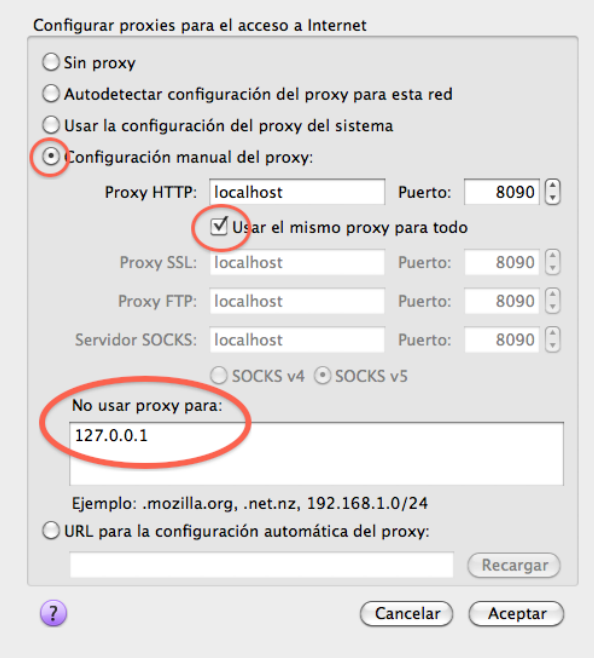
1. **Ejercicio 1.** Antes de comenzar a implementar nuestros tests, vamos a explicar el uso de “proxies” en JMeter para averiguar los parámetros de una petición al servidor, cuando éstos no son visibles en la url correspondiente. La creación de un plan de pruebas con JMeter puede presentar cierta dificultad cuando se ven implicadas *queries* y/o formularios complejos, peticiones https POSTs, así como peticiones javascript, en las que los parámetros que se envían por la red NO son visibles en la URL. JMeter proporciona un **servidor HTTP proxy**, a través del cual podemos utilizar el navegador para realizar las pruebas, y JMeter “grabará” las peticiones http generadas, tal y como se enviarán al servidor, y creará los correspondientes HTTP *samplers*. Una vez que hayamos “guardado” las peticiones HTTP que necesitemos, podemos utilizarlas para crear un plan de pruebas. Vamos a ver cómo utilizar dicho servidor *proxy*.

- A) Podemos añadir un servidor *proxy* para realizar peticiones HTTP, desde el menú contextual del BANCO DE TRABAJO, seleccionando “Añadir→ElementosNoDePrueba→Servidor Proxy HTTP”. En la configuración del *proxy* tendremos que asegurarnos de que el **puerto** no está ocupado con algún otro servicio en nuestra máquina local. En nuestro caso, en el puerto 8080 tenemos ejecutando al servidor de aplicaciones Glassfish, por lo que tendremos que cambiar dicho puerto, por ejemplo al 8090. El campo “Controlador objetivo” indica en qué lugar del plan de pruebas se almacenarán los elementos “grabados”. Dejaremos la opción por defecto: “Utilizar Controlador Recording”. Tendremos que asegurarnos de que el plan de pruebas incluye un controlador de este tipo (explicaremos un poco más adelante cómo hacerlo). Por defecto, JMeter grabará cualquier cosa, incluyendo páginas HTML, ficheros javascript, hojas de estilo csss, imágenes, etc, la mayoría de las cuales no nos serán de mucha utilidad para nuestro plan de pruebas. Para “filtrar” el tipo de contenido que queremos (o no queremos) utilizaremos patrones URL a Incluir (o a Excluir). Por ejemplo, vamos a excluir del proceso de grabación las imágenes y las hojas de estilo. Para ello añadiremos

“.\*\gif” y “.\*\css” como patrones URL a excluir. Para ello primero tendremos que pinchar sobre el botón añadir, y a continuación hacer doble click en la nueva fila en blanco añadida (en color blanco). NO hay que poner comillas al introducir el patrón (necesitarás añadir dos filas, una para cada patrón). Fíjate que no hay ningún botón o similar para aceptar los valores de configuración de los componentes JMeter. Dichos valores quedan registrados en cuanto los tecleamos.

Como ya hemos indicado en la configuración del proxy, JMeter “grabará” todas las acciones que realicemos en el navegador en un controlador de tipo “*recording*”. Por lo tanto, tendremos que incluir un controlador de este tipo en nuestro plan de pruebas. Lo primero que haremos será añadir un grupo de hilos en nuestro plan. Desde el menú contextual del nodo “Plan de Pruebas”, elegiremos *Añadir→Hilos→Grupo de hilos*. A continuación, desde el menú contextual del grupo de hilos que acabamos de añadir tendremos que elegir *Añadir→Controlador lógico→Controlador Grabación*.

- B) A continuación tendremos que **configurar el navegador** (en este caso **Firefox**) para utilizar el puerto en donde actuará el *proxy* JMeter. Para ello iremos al menú de “*Preferencias→Avanzado→Red→Conexión→Configuración→Configurar proxies para el acceso a Internet*”. Desmarcamos la opción actual (“Usar la configuración proxy del sistema”), y marcamos “Configuración manual del proxy”. Como valor de Proxy HTTP, pondremos localhost, y el puerto el 8090. Acuérdate de marcar la casilla “Usar el mismo proxy para todo”. También tenemos que asegurarnos de que en el campo “No usar proxy para” NO no figure “localhost”. La siguiente figura muestra la configuración Inicial de Firefox, y la configuración que debemos usar para utilizar el *proxy* JMeter.

	
Configuración inicial Firefox	Configuración Firefox para usar el Proxy

Vamos a utilizar el *proxy* de JMeter para ver cuáles son las peticiones http que tenemos que utilizar para “loguearnos” en el sistema. La petición de *login* es una petición http de tipo POST, y como ya hemos comentado, los parámetros de esta petición NO son visibles en el navegador, por lo que utilizaremos la información que ha grabado el *proxy* para “averiguar” dicha información.

C) Volviendo a JMeter, en la pantalla de configuración del Servidor Proxy vemos que al final de la misma tenemos los botones “Arrancar” y “Parar”. Una forma de asegurarnos de que el navegador está utilizando el *proxy* de JMeter, es, estando “parado” el *proxy*, intentar navegar desde Firefox. Si el navegador sigue sirviendo las páginas es que NO está utilizando el *proxy*. Una vez que nos hemos asegurado de que hemos configurado adecuadamente Firefox para utilizar el *proxy* de JMeter, desde el Servidor Proxy HTTP en JMeter, pulsamos sobre el botón “**Arrancar**” en la parte inferior de la pantalla de configuración del *Proxy* (al arrancar nos aparecerá una alerta indicando que se ha creado un certificado temporal en el directorio JMeter). Cuando “arrancamos” el *proxy* estamos en disposición de “grabar” cualquier “navegación” que hagamos desde Firefox. Vamos a grabar las siguientes acciones:

- ❖ Entramos en la aplicación de la tienda de animales (localhost:8080/jpetstore/actions/Catalog.action), a continuación pinchamos sobre **Sign In**.
- ❖ Utilizaremos el usuario z que ya tenemos creado y pulsamos sobre “Login”. (**Aclaración importante:** Si el usuario z no está creado créalo en este momento, y a continuación haz “Login”. Recuerda que la aplicación JpetStore utiliza una base de datos en memoria, por lo que si paramos el servidor de aplicaciones, se “borrará” cualquier dato almacenado en dicha base de datos).
- ❖ Finalmente pulsamos sobre “Sign Out” y **PARAMOS** la grabación desde el *proxy* de JMeter.

Ahora podemos ver las peticiones http que han quedado grabadas en el “Controlador Grabación” de nuestro plan. Pincha sobre cada una de ellas y observa la ruta y parámetros de petición (**comprueba que cuando hacemos “login” en el sistema se envían 5 parámetros, y que es una petición POST**). Utilizaremos esta información en el siguiente ejercicio para confeccionar nuestro plan de pruebas.

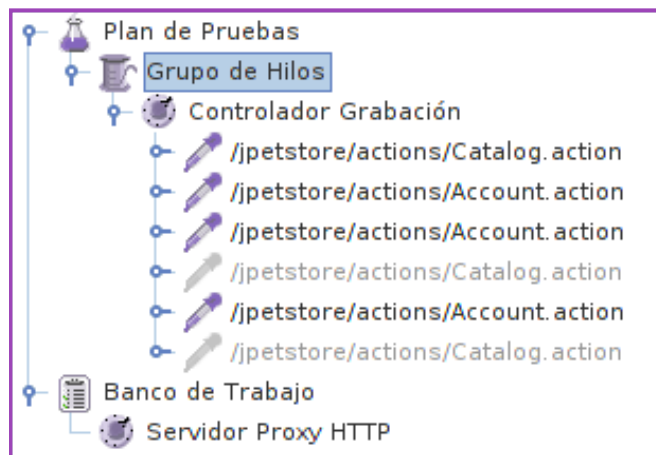
Observarás que hay un par de peticiones que están deshabilitadas (aparecen en gris claro). Se trata de redirecciones realizadas por la aplicación, que también quedan grabadas, pero que no se lanzarán cuando ejecutemos el test.

Una vez terminado el ejercicio recuerda volver a la configuración inicial de red en Firefox para poder acceder a Internet sin utilizar el *proxy* de JMeter.

**Nota:** Cuando ejecutemos el plan de pruebas, el “Controlador Grabación” no tiene ningún efecto sobre la lógica de ejecución de las acciones que contiene, es como el controlador “Simple”, es decir, actúa simplemente como un nodo que “agrupa” un conjunto de elementos. También es importante tener en cuenta que cuando grabemos nuestro plan de pruebas en el disco duro, **NO** se guardan los elementos del Banco de Trabajo, por lo que si queremos tenerlos almacenados para poder recuperarlos de nuevo, tendremos que “guardar” explícitamente el elemento “Banco de Trabajo” (desde el menú contextual del elemento “Banco de Trabajo”, seleccionaremos “Guardar como”). Algunos componentes JMeter, como por ejemplo, el servidor proxy HTTP sólo pueden añadirse en el banco de trabajo.

Si quisiéramos abrir un banco de trabajo previamente guardado, y un plan de pruebas, podemos hacerlo “abriendo” uno de ellos (el plan o el banco de trabajo), y para “abrir” el segundo elemento deberemos utilizar la opción Archivo→Mezclar. De no hacerlo así, al abrir el segundo elemento, JMeter “cerrará” el primero.

Nuestro plan de pruebas JMeter cuando finalicemos el ejercicio tendrá un aspecto similar a éste (la imagen mostrada se corresponde con una secuencia de acciones que no incluyen la creación del usuario z. Dependiendo del número de acciones que hayas grabado tendrás más o menos elementos hijo del nodo Controlador Grabación):



2. **Ejercicio 2.** Vamos a crear un plan de pruebas para ver cuál es el rendimiento de nuestra aplicación JPetStore. Llamaremos a nuestro plan de pruebas **“Plan JMeter”**. Lo primero que deberíamos hacer sería añadir un **grupo de hilos** a nuestro plan (desde el menú contextual de *Plan JMeter*→*Añadir*→*Hilos(Usuarios)*→*Grupo de Hilos*). En este caso, ya lo hemos añadido en el ejercicio anterior. Como ya hemos visto, JMeter trabaja simulando un conjunto de usuarios concurrentes realizando varias tareas sobre la aplicación. Cada usuario simulado se representa como un hilo. Por lo tanto, el número de hilos representa el número de usuarios simultáneos generados por el grupo de hilos. De momento dejaremos los valores por defecto: un hilo, un periodo de subida de 1 y un bucle con una única iteración.

A) Ahora vamos a añadir dos **Elementos de configuración**. Los elementos de configuración evitan que tengamos que introducir repetidamente alguna información de configuración de los samplers. Por ejemplo, si en el plan de pruebas incluimos varias llamadas a algunas páginas que están protegidas mediante una autenticación http básica, se podría compartir la información de usuario y password, para no tener que repetir estos datos para cada *sampler* dentro del plan de pruebas. Para pruebas de aplicaciones Web, el elemento de configuración más importante es **“HTTP Request Defaults”**. Lo añadimos desde el menú contextual del grupo de hilos (*Añadir*→*Elemento de Configuración* →*Valores por defecto para petición HTTP*), y fijaremos el nombre del servidor a **localhost**, y el número de puerto a **8080**. Otro elemento de configuración muy útil es el **“Gestor de cookies HTTP”**. Éste almacena *cookies* y hace que estén disponibles para subsecuentes llamadas al mismo sitio, tal y como haría un navegador. Dado que cada hilo representa un usuario diferente, las *cookies* no se compartirían entre hilos. Lo añadiremos desde el menú contextual del grupo de hilos (*Añadir*→*Elemento de Configuración*→*Gestor de Cookies HTTP*). Si nos equivocamos al crear cualquier elemento, en cualquier momento podemos borrar/modificar/mover cualquier elemento del plan. Para borrar un elemento lo haremos desde el menú contextual de dicho elemento. Para modificar su configuración simplemente seleccionaremos dicho elemento. Para mover un elemento a otra posición dentro del plan, seleccionaremos y arrastraremos dicho elemento.

B) Ahora vamos a añadir un **sampler** para hacer peticiones de páginas http a nuestro servidor de aplicaciones y así simular la acción de los usuarios reales (desde el menú contextual del grupo de hilos: *Añadir*→*Muestreador*→*Petición HTTP*). Si, desde Firefox, nos situamos en la página inicial de la aplicación veremos que la URL es *“http://localhost:8080/jpetstore/”*. El servidor y el puerto ya los hemos indicado en un elemento de configuración. Después del contexto inicial, vendría el nombre del recurso, en este

caso, la página html, a la que queremos acceder. Observamos que no aparece esta información (no es necesaria, por defecto se llamará a index.html) . Por lo tanto, lo único que haremos será indicar en el campo **Ruta** el valor `"/jpetstore/"`. Vamos a hacer uso de varios *samplers* http, por lo que será conveniente indicar un nombre adecuado, por ejemplo, para este primer sampler podemos especificar como nombre el valor **"Home"**



- C) Para asegurarnos de que estamos en la página correcta, vamos a añadir una **aserción** (desde el menú contextual de la **petición HTTP Home: Añadir→Aserciones→Aserción de Respuesta**), marcaremos "Respuesta Textual", y el patrón a probar contendrá la cadena "Welcome to JPetStore 6" (El texto anterior no se pone con las comillas. Recuerda que primero tienes que pinchar sobre el botón "Añadir", y hacer doble click en la fila en blanco añadida para editar su contenido).
- D) De nuevo en el navegador, ahora vamos a entrar en la aplicación, pulsando sobre el enlace correspondiente. Esto significa que tenemos que añadir una nueva **petición http** (desde el grupo de hilos), en este caso, puesto que la página mostrada nos muestra el catálogo de animales, podemos asignarle como nombre **"Catalog"**. Utiliza el navegador y pincha en entrar en la tienda para ver la URL de la nueva petición. En este caso la ruta tendrá el valor `"/jpetstore/actions/Catalog.action"`. para verificar que entramos en la página correcta añadiremos una aserción igual que en el caso anterior con el patrón a probar "Saltwater".
- E) Desde la página anterior, vamos a acceder al menú de Reptiles (por ejemplo pinchando sobre la imagen correspondiente). Tendremos, por lo tanto, que añadir otra **petición http** (le pondremos el nombre **"Reptiles"**). Podemos comprobar que la URL del navegador, es la siguiente: `"http://localhost:8080/jpetstore/actions/Catalog.action?viewCategory=&categoryId=REPTILES"` En este caso, vemos que la petición tiene dos parámetros (los parámetros se especifican con pares *nombre=valor*, separados por '&'): `viewCategory` (que en este caso no tiene valor asociado, pero que es necesario incluir), y `categoryId` (cuyo valor es REPTILES). Por lo tanto, indicaremos, además de la ruta (`/jpetstore/actions/Catalog.action`), los parámetros anteriores con el valor correspondiente para `categoryId`. Acuérdate de que primero tienes que pulsar el botón añadir, y hacer doble click en las entradas añadidas para introducir los valores. No tienes que poner comillas). Añade una aserción de respuesta textual con el patrón "Reptiles".

**Nota:** Como acabamos de ver, cuando utilizamos *samplers* http, para averiguar la "ruta" y parámetros de la petición URL podemos fijarnos en la ruta que aparece en el navegador al ejecutar la aplicación. Esta aproximación funciona bien cuando se trata de peticiones "simples" y parámetros "sencillos" de "manipular". Páginas que contengan formularios grandes pueden requerir docenas de parámetros. Además, si se utilizan peticiones HTTP POST (como por ejemplo el envío de login y password), los valores de los parámetros no aparecerán en la URL, por lo que tendremos que descubrirlos de alguna otra forma. Una aproximación consiste en utilizar un **Proxy JMeter**, que grabará el caso de prueba por nosotros, tal y como hemos visto en el ejercicio 1, y que luego podremos modificar según nuestras necesidades,.

- F) Antes de continuar vamos a **grabar** nuestro plan de pruebas (como medida de precaución es recomendable grabar a menudo). Lo podemos hacer desde Archivo→Guardar, y podemos ponerle el nombre **"Plan-Ejercicio2"** (también podemos guardar nuestro plan desde el menú contextual del plan de pruebas, eligiendo la opción "Guardar como"). Dado que podemos ejecutar el plan en cualquier momento, vamos a hacerlo ahora, pero antes nos aseguraremos de que en las propiedades del grupo de



hilos solamente se genera un hilo, con un periodo de subida de 1 segundo y una única vez. Para poder “ver” lo que ocurre al ejecutar las pruebas, resulta muy útil (por no decir imprescindible) añadir uno o varios “Listeners” que nos muestren (y/o graben los resultados de las pruebas en algún fichero). Utilizaremos, de momento, dos de ellos. Desde el menú contextual del plan de pruebas elegimos Añadir→Receptor→**Resultados de la aserción**. También añadiremos el “Listener” (Receptor) **Ver árbol de resultados**. El primero de ellos nos mostrará en pantalla los resultados de las aserciones de nuestro plan, para cada uno de los *samplers* del plan. El segundo nos muestra las peticiones y respuestas de cada uno de los muestreadores. Para **ejecutar el plan** tienes que ir al menú de JMeter y seleccionar Lanzar→Arrancar (también puedes utilizar el icono con forma de triángulo verde). En el extremo derecho de la barra de herramientas de JMeter verás un pequeño cuadrado. Durante la ejecución del plan, verás que el cuadrado se muestra de color verde. Cuando acaba la ejecución vuelve a estar gris. Selecciona cualquiera de los *listeners* que hemos añadido y observa los resultados. Si ejecutas el plan con un *listener* seleccionado, verás cómo se va actualizando durante la ejecución. Realiza varias ejecuciones para familiarizarte con la información que proporcionan estos dos *listeners*. (si todas las aserciones se evalúan a “true” verás que el *listener* correspondiente no muestra ninguna información. Prueba a cambiar la aserción para que de “fallo”, y podrás observar que el *listener* muestra el error producido). Verás que los *listeners* muestran los resultados de la ejecución “a continuación” de los resultados de ejecuciones anteriores. Podemos “limpiar” el panel de resultados pinchando sobre el icono correspondiente. A continuación mostramos el aspecto del plan de pruebas hasta este momento, así como la imagen del icono para *limpiar* los resultados de los *listeners* de ejecuciones anteriores.

	
Plan de pruebas	Icono para “limpiar” los resultados de los <i>listeners</i>

- G) Ahora “estructuraremos” nuestro plan de pruebas añadiendo **controladores lógicos** para “alterar” el comportamiento secuencial del plan. Vamos a incluir un “Controlador Interleave” desde el menú contextual del Grupo de Hilos: Añadir→Controlador Lógico→Controlador Interleave. Este controlador “alterna” la ejecución de uno de sus hijos en cada iteración de cada uno de los hilos. Podéis consultar el funcionamiento de este elemento y de cualquier componente de JMeter en ([http://jmeter.apache.org/usermanual/component\\_reference.html](http://jmeter.apache.org/usermanual/component_reference.html)). Situaremos el controlador justo después del elemento Catalog, e incluiremos como nodos hijo tres peticiones http. Una de ellas es la que hemos denominado Reptiles. Las otras dos son el acceso a los submenús Dogs, y Birds, respectivamente. Puedes crearlos rápidamente “duplicando” el nodo Reptiles y editando los elementos correspondientes.

- H) A continuación añadiremos las **peticiones http** para “loguearnos” en el sistema (como hijos del grupo de hilos), y las llamaremos “**Sign in form**”, “**Login**” respectivamente. La primera es la petición que realizamos cuando “pinchamos” sobre el enlace “Sign in” en el navegador, y la segunda corresponde con la introducción del login y password y pinchar sobre el enlace “Login” en el navegador Firefox. Estas acciones las tenemos “grabadas” en el controlador Grabación del ejercicio anterior. Identifícalas y cópialas (o muévelas) a continuación del controlador *Interleave*. Seguidamente introducimos una petición para mostrar el catálogo (podemos, por ejemplo duplicar el elemento “Catalog” que ya teníamos antes (fíjate que, en el navegador Firefox, después de “pinchar” sobre “Login”, se llama de nuevo a la página que muestra el catálogo general de especies). El nuevo elemento “**Catalog**” tendrá asociada dos aserciones de Respuesta: una con el patrón “Sign Out”, y una segunda con el patrón “Welcome z”. (**IMPORTANTE:** JMeter es sensible a las mayúsculas/minúsculas, asegúrate de que el patrón coincide exactamente con el mensaje que aparece en la página). Añade otra petición Http con el nombre “**My Account**” que es la petición que se genera cuando en el navegador “pinchamos” sobre “My Account”. Asociaremos una aserción de respuesta con el patrón “User Information”. Finalmente añadimos una última petición para salir de nuestra cuenta (utiliza la petición que tenemos grabada) a la que llamaremos “**Sign off**”, seguida de otra petición “**Catalog**”, esta vez con una aserción de respuesta con el patrón “Sign In”.
- I) Vamos a borrar el controlador Grabación, ya que ya no nos hace falta, simplemente lo hemos utilizado para “averiguar” la configuración de las peticiones http que teníamos que realizar. Guardamos el plan y ejecutamos (Todas las peticiones y aserciones correspondientes tiene que aparecen “en verde”. Si no es así, corrige los posibles errores en tu plan). Podemos ver los resultados en los “listeners” Resultados de la Aserción y Ver árbol de resultados. El plan resultante debe tener este aspecto:



- J) Finalmente vamos a **añadir dos listeners más** (desde el nodo de nuestro plan): un **informe agregado**, que proporciona un resumen general de la prueba; y un **gráfico de resultados**, que muestra el rendimiento de la aplicación. El informe agregado muestra una tabla con una fila para cada petición, mostrando el tiempo de respuesta, el número de peticiones, ratio de error, ... , entre otra información. El gráfico de resultados muestra el rendimiento de la aplicación (*throughput*) como el número de peticiones por minuto gestionadas por el servidor. Vuelve a ejecutar el plan y observa los resultados mostrados por los nuevos *listeners* que hemos añadido. Reduce al máximo el número de aplicaciones “abiertas” en el ordenador cuando estés ejecutando JMeter, ya que afectarán a los datos obtenidos por JMeter. En un caso de prueba “real” solamente deberíamos tener en ejecución la aplicación JMeter.
- K) Ahora vamos a suponer que está prevista una carga de 25 usuarios concurrentes. Nuestras especificaciones estipulan que con 25 usuarios concurrentes, el tiempo de respuesta debe ser inferior a 1 segundos por página. Prueba a cambiar los parámetros del grupo de hilos de nuestro plan para verificar si nuestra aplicación cumple con los requisitos de rendimiento acordados con el cliente. Explica qué valores necesitas utilizar para el número de hilos, el periodo de subida y el número de iteraciones en el bucle. Ten en cuenta que en cualquier momento podemos parar la ejecución del plan.