Universidad de Valladolid
E.S Ingeniería Informática
Grado en Ingeniería Informática (IS)

Curso 2019/2020

Evaluación y Rendimiento de Sistemas Software
Práctica de Laboratorio: Rend. Lab1 - Pruebas de rendimiento

Máquina Virtual 21
Arenas Guerra, Ignacio
García Asensio, Eduardo
Herruzo Herrero, Juan
Rodríguez Ares, Silvia

Introducción 4

Configuración del entorno de pruebas: servidor, cliente(s), configuración de la monitorización, etc. 4

Servidor 4

Servidor Apache. 4

Cliente. 5

Análisis del rendimiento de un servidor web (LRdto.PT1) 5

Descripción de la prueba 5

Objetivos y restricciones 5

Ajustes en la configuración del entorno de pruebas 6

Diseño de la prueba 6

Métricas (definición y unidades) 6

Especificación de la carga de trabajo 6

Descripción del plan de pruebas. 7

Análisis de resultados. 7

Preprocesado de los resultados 7

Análisis individual 8

Análisis comparativo 12

Conclusiones 14

Análisis de la capacidad de un servidor web. (LRdto.PT2) 14

Descripción de la prueba 14

Objetivos y restricciones 15

Ajustes en la configuración del entorno de pruebas 15

Diseño de la prueba 15

Métricas (definición y unidades) 15

Especificación de la carga de trabajo 15

Descripción del plan de pruebas. 16

Análisis de resultados. 17

Preprocesado de los resultados 17

Análisis 17

Conclusiones 19

Evaluación del rendimiento de una aplicación web. (LRdto.PT3) 20

Descripción de la prueba 20

Objetivos y restricciones 20

Ajustes en la configuración del entorno de pruebas 20

Diseño de la prueba 21

Métricas (definición y unidades) 21

Especificación de la carga de trabajo 21
Descripción del plan de pruebas. 21
Análisis de resultados 23
Preprocesado de los resultados 23
Análisis 24
Conclusiones 27
Conclusiones generales 27
Bibliografía 28
Anexos 29
ANEXO I 29
ANEXO II 31
ANEXO III 33
ANEXO IV 35
ANEXO V 38
ANEXO VI 39
ANEXO VII 40
ANEXO VIII 41
ANEXO IX 42
ANEXO X 45
ANEXO XI 46
ANEXO XII 48
ANEXO XIII 49

1. Introducción

En el siguiente documento se plantean el diseño, análisis y las conclusiones obtenidas tras efectuar diferentes tipos de pruebas de rendimiento sobre los escenarios proporcionados por el profesor para un sistema en concreto también proporcionados por el profesor. Para la ejecución de las pruebas y la obtención de resultados se ha utilizado la herramienta JMeter mientras que para el análisis exploratorio y la representación de los datos se ha utilizado el lenguaje Python y las librerías de análisis estadístico que este proporciona. Como es un caso académico de estudio de un sistema, no se han planteado requisitos de rendimiento sobre este, por lo que se han obviado las fases de análisis del entorno de pruebas y el planteamiento de criterios de aceptación. Las conclusiones han sido extraídas de los datos y su representación, atendiendo a hipótesis que nos hemos formulado basadas en los conocimientos teóricos que se nos han impartido en la asignatura.

2. Configuración del entorno de pruebas: servidor, cliente(s), configuración de la monitorización, etc.

2.1. Servidor

Como servidor, usamos una máquina virtual con las siguientes características:

- Sistema Operativo: Linux virtual 4.15.0-76-generic de 64 bits.
- Procesador: Common KVM
    - Memoria caché: 16MB
    - Nº de cores: 2
    - Velocidad CPU: 2.3GHz.
- Espacio disponible: 3.9GB de 9.8GB
- Memoria RAM: 1.9GB
- Espacio Swap: 2.0 GB

Esta información ha sido obtenida ejecutando una serie de comandos, cuyos resultados se pueden ver en el ANEXO I.

2.1.1. Servidor Apache.

El servidor apache del sistema a probar es un apache versión 2.4.29. Esta configuración la podemos encontrar en el fichero `/etc/apache2/apache2.conf`.De este fichero las configuraciones de interés para este conjunto de pruebas son:

- `Timeout 600`
  Es el tiempo que el servidor va a estar esperando para un envío o recepción, si se supera este tiempo el resultado será un timeout, se mide en segundos.

- `MaxKeepAliveRequests 0`
  Indica la cantidad total de usuarios concurrentes que puede soportar el servidor, si se deja en 0 significa ilimitados.

- `KeepAliveTimeout 120`
  Es el tiempo que el servidor va a tener abierta una sesión con un usuario esperando a una nueva petición, si no se recibe una petición en ese tiempo, se cierra la sesión con el usuario, se mide en segundos.

2.2. Cliente.

En la parte de cliente, usamos jair el cual tiene las siguientes características:

- Sistema Operativo: Linux 5.4.28-gentoo-eii de 64 bits.
- Procesador: Common KVM
  - Nº de cores: 2
  - Velocidad CPU: 2.1 GHz
  - Memoria caché: 16MB
- Espacio disponible: 228GB
- Memoria RAM: 3,8 GB

Esta información ha sido obtenida ejecutando una serie de comandos, estos comandos y su resultado se pueden ver en ANEXO II

3. Análisis del rendimiento de un servidor web (LRdto.PT1)

3.1.Descripción de la prueba

3.1.1. Objetivos y restricciones

El objetivo en esta parte de la práctica será evaluar la productividad del servidor web Apache instalado en nuestra máquina virtual. Para ello realizamos peticiones http a páginas estáticas: una con una imagen de gran tamaño y otra con una imagen pequeña y estudiaremos su comportamiento en situaciones de carga habitual (load test) y cómo varía el rendimiento del sistema en función del tamaño de la página web cargada.

Se puede mencionar que, como restricción, si la práctica no se ejecuta desde jair, debemos restar la latencia que se produce en la conexión desde el pc que lo ejecutemos. En nuestro caso todas las hemos realizado desde jair así que no tenemos este problema. Por otro lado, restringimos la concurrencia de usuarios al diseño de la prueba dado por el profesor. Los

resultados estarán condicionados por el cliente y el servidor en el cual estamos realizando las pruebas, cuyas características ya hemos analizado.

3.1.2. Ajustes en la configuración del entorno de pruebas

El sistema de estudio es un servidor apache versión 2.4.29, el cual venía mayormente preconfigurado cuando se nos dió acceso a él, también la propia configuración del sistema operativo estaba hecha y no hemos necesitado hacer ningún cambio a ninguno de los dos.

Algunos de los ajustes que hemos tenido que hacer han sido la eliminación de ficheros de log y el arranque de la aplicación PerfMon para que conectara con JMeter y enviase información sobre el estado de la máquina.

3.2.Diseño de la prueba

3.2.1. Métricas (definición y unidades)

Para este análisis se van a utilizar tres métricas:

- Tiempo de respuesta:
  Intervalo de tiempo desde que el cliente inicia la petición de servicio hasta que el servicio devuelve la respuesta completamente, se incluye el RTT y la latencia, pero al tratarse de que todas las pruebas se ejecutan en una red local, estos valores no serán tenidos en cuenta.
  Se mide en milisegundos (ms)

- Productividad:
  Cantidad de servicios que atiende el servidor por unidad de tiempo, en nuestro caso se tratara de peticiones HTTP por segundo.

- Utilización:
  Fracción del tiempo que un recurso está ocupado sirviendo peticiones, en este caso se analizará el porcentaje de tiempo que un recurso está ocupado si ese recurso solo puede estar ocupado o libre (CPU) o analizaremos el porcentaje de ocupación medio a lo largo de las pruebas (Memoria).
  Esta métrica la utilizaremos para comprobar el estado del sistema, conociendo si se encuentra saturado y que recurso está siendo el cuello de botella.

3.2.2. Especificación de la carga de trabajo

Esta prueba va a contar con tres niveles distintos de concurrencia de usuario, 75, 150 y 300. Para no sobrecargar inicialmente el servidor se activarán los hilos a una velocidad de tres hilos por segundo, una vez activados todos los hilos se mantendrá la carga estacionaria durantes 240 segundos más, este intervalo será el que va a ser medido. Cada una de las pruebas con distinto nivel de concurrencia será ejecutada tres veces en distintos momentos del día. De esta forma se obtienen resultados más independientes del estado momentáneo de la máquina que si solo se ejecutan una vez.

3.2.3. Descripción del plan de pruebas.

En el ANEXO III está la estructura general de este plan de pruebas. Dicho plan de pruebas está constituido por un "*grupo de hilos*" cuyos parámetros "*número de hilos*", "*periodo de subida*" y "*duración*" están configurados con la sintaxis `${__P(<<PARAM>>,300)}`, de esta forma el valor puede ser modificado desde un script, PARAM puede tomar los valores THREADS, RAMP_UP y DURATION para cada parámetro que define las pruebas. El grupo contiene una única petición HTTP a la página web objetivo("*/test/imagenp.html*" o "*/test/imageng.html*" según el recurso web que se desea analizar). Está peticion se hace sobre el servidor virtual.lab.inf.uva.es:31212, el cual ha sido añadido en el plan a través del "*Elemento de configuración por defecto para HTTP*". Por último se añade el receptor "*PerfMon Metrics Collector*" configurado para que recoja información sobre el uso de la CPU, tanto en modo user como en modo system, la ocupación de memoria principal y el número de operaciones I/O en disco.

3.3. Análisis de resultados.

3.3.1. Preprocesado de los resultados

Por cada nivel de concurrencia y página objetivo se han realizado tres repeticiones del test en diferentes momentos del día, lo que conlleva obtener tres resultados por cada combinación. El primer paso es eliminar los periodos iniciales y de parada, para así quedarnos solo con el periodo estacionario. Para ello, utilizamos los parámetros de Jmeter explicados en el apartado anterior y, a mayores, corroboramos que estamos en el estado estacionario aplicando la técnica de truncamiento. [Figura 1 y Figura 2 respectivamente]

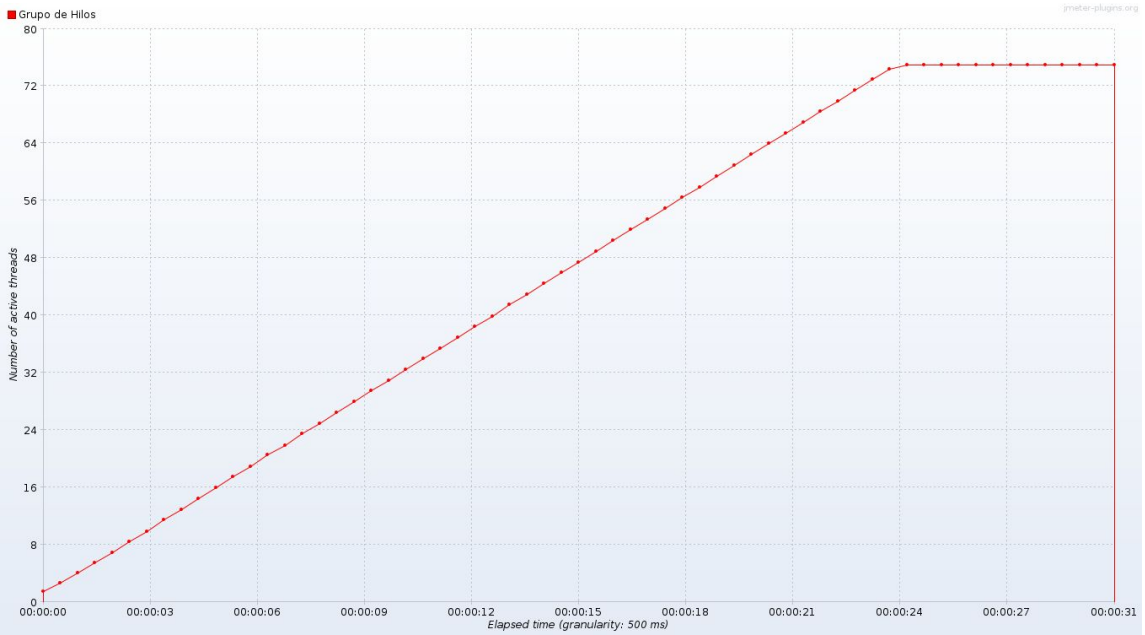

Figura 1 Representación de los hilos activos durante primeros 30 segundos de la prueba

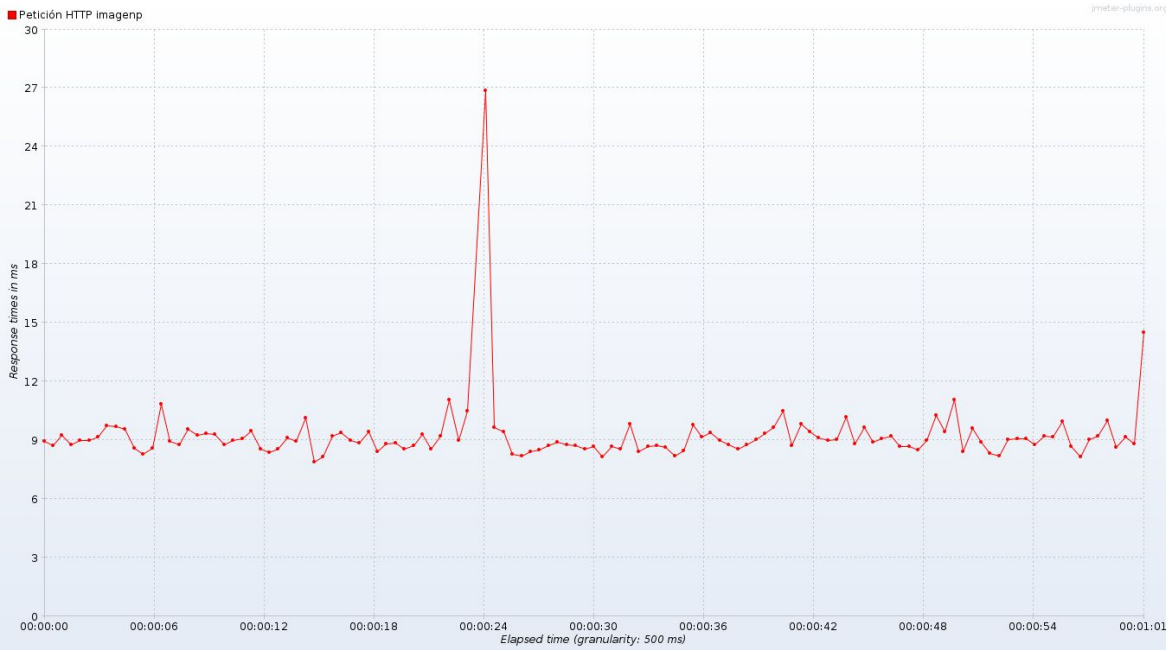

Figura 2 Representación del periodo estacionario

3.3.2.Análisis individual

Este escenario requiere tres repeticiones por nivel de concurrencia, por lo tanto al final de las ejecuciones tendremos tres archivos de resultados para cada nivel.

Para poder sacar la media global necesitamos primero sacar las medias de cada uno de estos ficheros, y luego sacar una vez más la media de estas medias. A parte de la media obtenemos también la varianza y el intervalo de confianza para la media con una confianza del 95% como viene explicado en [7]. Como sacamos las métricas de tiempo de respuesta y productividad, este proceso se realiza dos veces, una para cada métrica.

Nuestro método para analizar los datos extraídos de las pruebas en el lenguaje Python basada en las librerías "Pandas"[8], "Matplotlib"[9] y "Numpy"[10] se centra en el procesado de los archivos con extensión "csv" obtenidos a través del uso de los receptores de JMeter. Posteriormente se filtran los datos manteniendo sólo las columnas que nos interesen utilizando las funciones proporcionadas por la librería "Pandas" que luego visualizamos utilizando los métodos incluidos en la librería "Matplotlib.pyplot". La librería "Numpy" es necesaria para poder trabajar con las dos mencionadas previamente. Los resultados extraídos de este procedimiento se encuentran en el ANEXO IV.

Sobre estos valores lo primero que podemos concluir es que el incremento del tiempo de respuesta crece linealmente con la cantidad de usuarios concurrentes doblándose cada vez que se doble la cantidad de usuarios concurrentes. Este incremento es independiente del tamaño del recurso solicitado como se puede ver en [Figura 3 y 4].

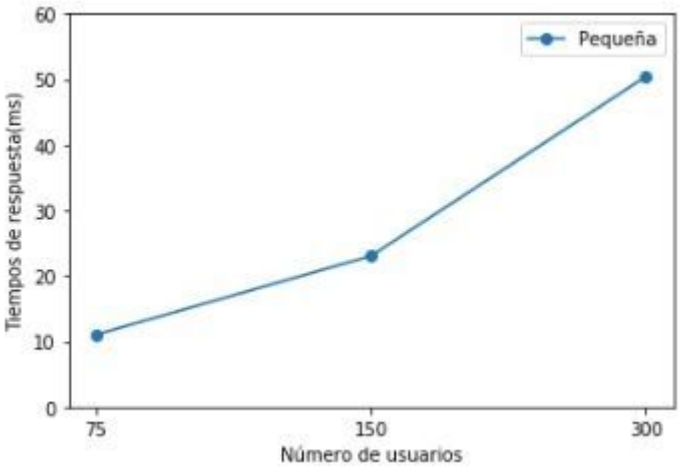

Figura 3 Media del tiempo de respuesta para la imagen pequeña en función de número de usuarios concurrentes

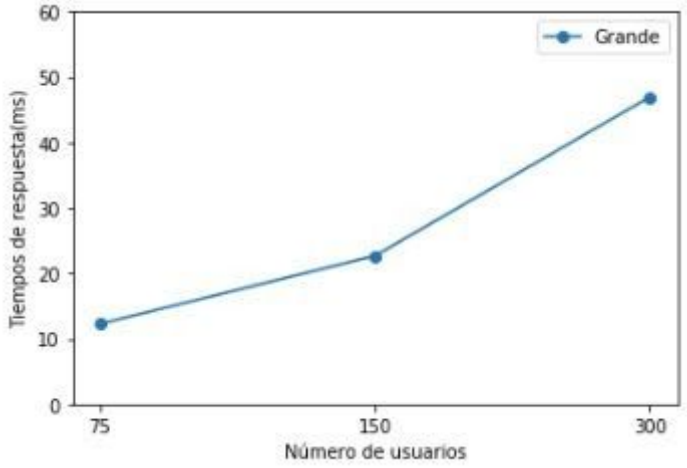

Figura 4 Media del tiempo de respuesta para la imagen grande en función de número de usuarios concurrentes

Las productividades en ambos casos se mantiene prácticamente estables [Figura 5 y 6] el ligero decremento se puede deber a la mayor carga que tendrá el sistema empaquetando y enrutando las peticiones y no al tamaño del recurso web solicitado.

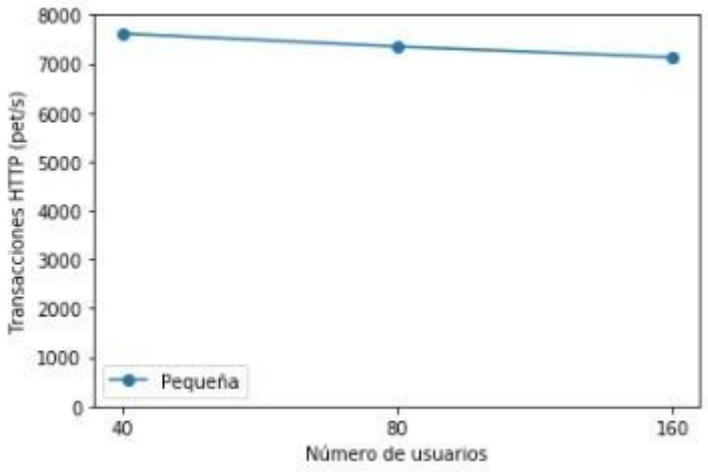

Figura 5 Media de la productividad para la imagen pequeña en función de número de usuarios concurrentes

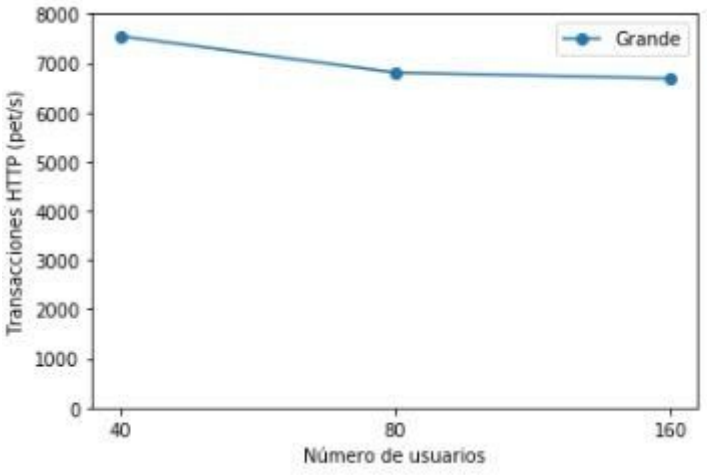

Figura 6 Media de la productividad para la imagen grande en función de número de usuarios concurrentes

Este cambio en los tiempos de respuesta y productividad es coherente; tenemos una situación en la cual la carga de trabajo se dobla y como la utilización de la CPU del sistema es elevada hasta el punto de estar prácticamente saturado [Figuras 7 y 8], la productividad no puede aumentar por lo que el tiempo de respuesta se ve incrementado el doble.

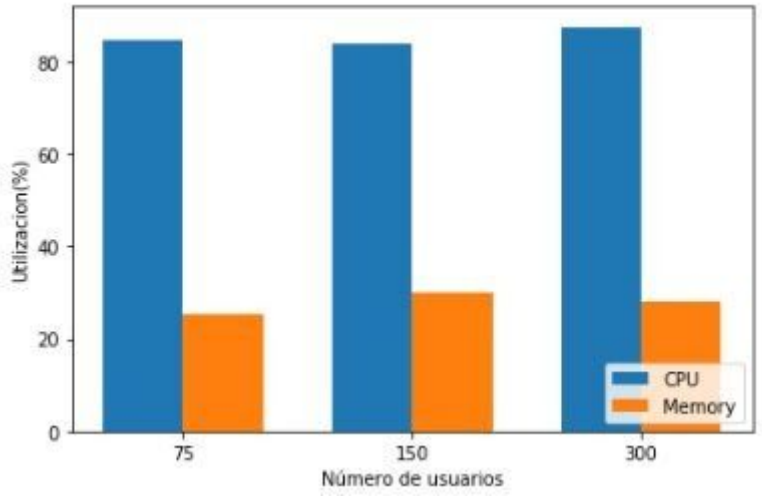

Figura 7 Media de la utilización de CPU y memoria para la imagen pequeña en función de número de usuarios concurrentes

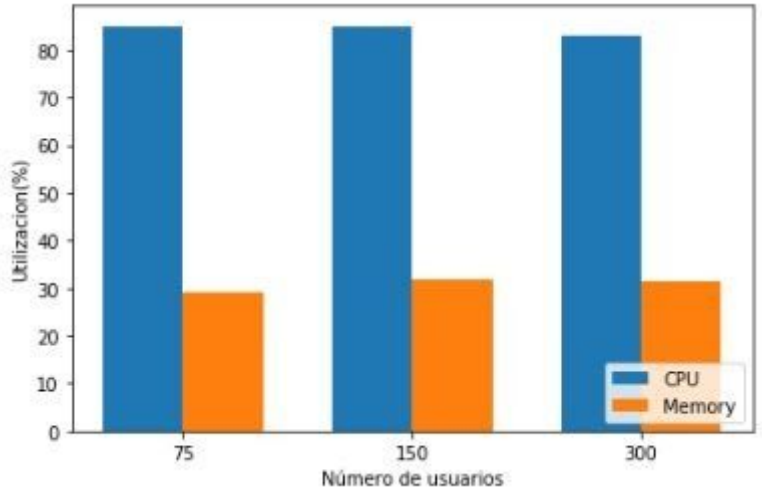

Figura 8 Media de la utilización de CPU y memoria para la imagen grande en función de número de usuarios concurrentes

3.3.3. Análisis comparativo

En este apartado vamos a comparar las diferencias que puedan existir en función del tamaño del recurso solicitado. Los tamaños de los recursos que se solicitan en cada prueba son:

- Prueba imagen pequeña: 236 KB
- Prueba imagen grande: 2.5MB

Por lo tanto el tamaño del recurso de la prueba de imagen grande es 10.6 veces más grande que el de la imagen pequeña.

Para poder comprobar si existe diferencia significativa entre los tiempos de respuesta hay que calcular la diferencia entre medias, para ello podemos seguir el procedimiento explicado en [8 página 95].

Al tener media, varianza y número de repeticiones, solo faltaría: (la explicación está hecha con el nivel de 75 usuarios concurrentes)

- La de calcular la diferencia entre medias

$$\overline{x} = \overline{x_{pequeña}} - \overline{x_{grande}} = 11.043659 - 12.269352 = -1.225693$$

- Las desviaciones estándar para cada para la resta.

$$s = \sqrt{\sigma_{pequeña}/n_{pequeña} + \sigma_{grande}/n_{grande}} = 1.38477$$

- Los grados de libertad.

$$df = 1.705$$

Ahora podemos sacar el intervalo de confianza al 95% el cual es:
$(-7.183; 5.958)$
Como este intervalo contiene al 0 se puede decir que la diferencia no es significativa al 95% para 75 usuarios. El resto de intervalos se pueden encontrar en el ANEXO V aunque en todos la diferencia no es significativa al 95%. Por lo tanto podemos decir que los tiempos de respuesta son iguales entre sí e independientes del tamaño del recurso solicitado. [Figura 9]

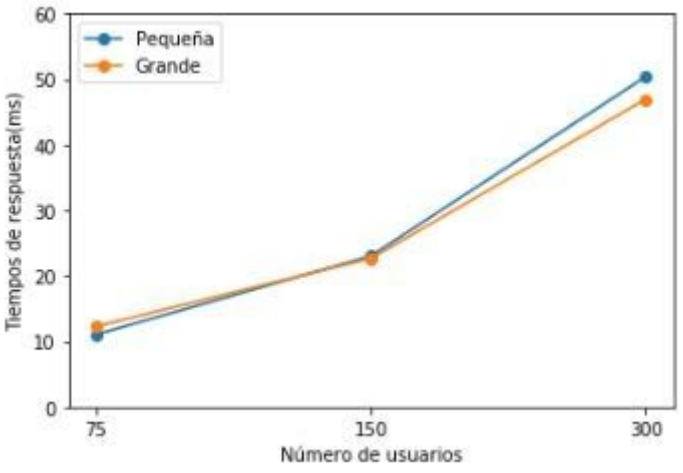

Figura 9 Comparación de tiempo de respuesta entre la imagen pequeña y la imagen grande

Comparando las productividades entre imagen pequeña y grande, observamos que para 75 usuarios son casi idénticas, siendo el servidor solo un 0.87% más productivo para la imagen pequeña. Sin embargo, con 150 y 300 usuarios se produce un cambio significativo en las productividades, siendo para la imagen pequeña un 7.81% y un 6.86% mayores respectivamente. Aunque esta diferencia es poco notable y si comprobamos ahora las productividades utilizando la misma técnica que para los tiempos de respuesta basada en los

Intervalos de Confianza hemos llegado a la conclusión de que también son iguales, los intervalos están descritos en ANEXO VI. [Figura 10]

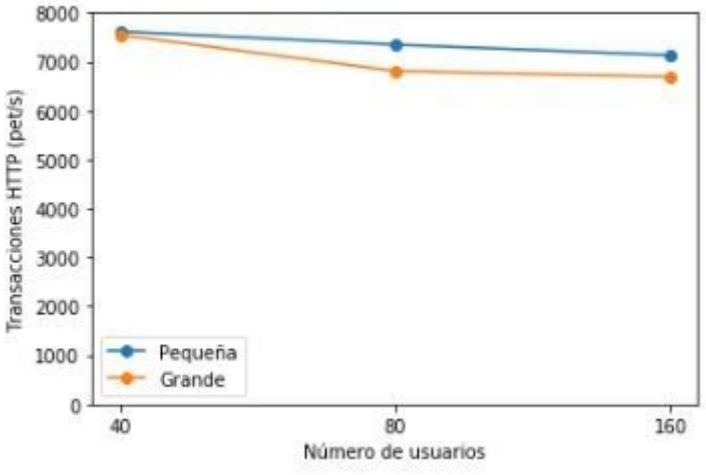

Figura 10 Comparación de productividad entre la imagen pequeña y la imagen grande

3.4. Conclusiones

Una vez analizadas los resultados de las pruebas se puede reconocer que el sistema actualmente trabaja saturado siendo el cuello de botella la CPU , esto hace que la productividad no pueda aumentar y por lo tanto el tiempo de respuesta aumenta a la vez que aumenta el número de usuarios concurrentes. Además no existe correlación significativa entre el tamaño de la página web y la productividad del sistema o los tiempos de respuesta, podemos denotar que es un factor que no afecta al rendimiento del sistema. Por lo tanto el factor principal en la productividad y el tiempo de respuesta del sistema va a ser la cantidad de usuario concurrentes. Esto hace que en la situación habitual de trabajo de sistema el tiempo de respuesta sea proporcional al número de usuarios concurrentes teniendo constante la productividad, esto lo podemos decir con una confianza del 95%

4. Análisis de la capacidad de un servidor web. (LRdto.PT2)

4.1. Descripción de la prueba

Esta prueba busca evaluar la capacidad del servidor para procesar peticiones http. Se irá aumentando la carga de trabajo del servidor, para así acotar el momento en el cual aumenta el tiempo de respuesta y se disparan los errores de forma exponencial.

Como herramienta principal se va a usar, de nuevo, JMeter para hacer las pruebas, y una vez obtenidos los datos se usarán herramientas como Excel y bibliotecas de Python para procesar los datos y así poder analizarlos, siguiendo los procedimientos explicados en el apartado anterior.

4.1.1.Objetivos y restricciones

El objetivo principal de esta prueba es evaluar cuál es la carga máxima de trabajo que puede soportar nuestro servidor en circunstancias excepcionales. Por ello, como objetivos parciales podemos marcar los siguientes:

- Encontrar unos niveles de usuarios concurrentes que escalen de forma progresiva.
- Encontrar un último nivel de usuarios concurrentes, para el cual los tiempos de respuesta crezcan de forma anómala, o bien produzca una gran cantidad de errores por timeout.

Como objetivos de aprendizaje podemos destacar:

- Aprender a manejar JMeter para realizar una prueba de capacidad.
- Aprender a manipular datos, para transformarlos y poder analizarlos.
- Obtener conclusiones a partir del análisis de datos.

Hay que mencionar ciertas restricciones que tenemos que tener en cuenta:

- Limitaciones de manejo y recursos del servidor por ser una máquina virtual.
- Uso de JMeter en modo comando para lanzar pruebas pesadas que consumen gran cantidad de recursos.
- Desconocimiento de factores externos que pueden alterar los resultados, como puede ser el tráfico existente en la red en el momento de lanzar las pruebas.

4.1.2.Ajustes en la configuración del entorno de pruebas

En esta prueba no se han realizado ajustes extras a los realizados en la Prueba 1 comentadas en el apartado 3.1.2

4.2.Diseño de la prueba

4.2.1. Métricas (definición y unidades)

Las métricas a utilizar serán las mismas que en la prueba anterior explicadas en el apartado 3.2.1

4.2.2. Especificación de la carga de trabajo

El objetivo de esta prueba es encontrar el límite de capacidad del sistema, por lo tanto nos basaremos en hacer una prueba de carga creciente dividida en 6 niveles de concurrencia, estos niveles son 15, 30, 60, 120, 240 y 480 usuarios. El número de niveles lo hemos determinado atendiendo a los criterios de obtener una cantidad significativa de intervalos sobre los que trabajar y no excederse en el número de estos ya que aumentan el trabajo que hay que realizar en cuanto al procesado.

El periodo de subida es calculado respecto a la constante de activar 2 usuarios por segundo por lo tanto los tiempos para cada nivel respectivamente son 7,15,30,60, 120 y 240 segundos.

La razón por la que decidimos bajar la constante de tres usuario por segundo de la anterior prueba dos es porque la demanda de esta prueba es superior y en los primeros intentos ejecutando con los tres usuarios originales saturaron el sistema y aparecia un gran pico en los tiempos de respuesta en el periodo inicial.

El periodo estacionario objetivo es de 120 segundos, inicialmente utilizábamos este mismo tiempo entre nivel y nivel, pero en esta prueba el sistema tarda en estabilizarse y pasar al periodo estacionario, por lo que el periodo estacionario se veía reducido en algunos casos hasta ser de solo un minuto. Por esto el tiempo final utilizado entre cada carga es de 200 segundos, de los cuales se analiza un tramos de 120 segundos que se encuentra en el periodo estable.

Por último el tiempo de bajada es de 0, es decir, una vez se acaba el tiempo el sistema cliente no mandará ninguna petición más, pero si que esperará a las activas a que acaben.

4.2.3. Descripción del plan de pruebas.

En el ANEXO III está la estructura general de este plan de pruebas.

Al tratarse de solo una prueba, este plan está creado con los valores de subida, ramp up y duración integrados en él, sin necesidad de usar la sintaxis `${__P(<<PARAM>>,300)}`.A diferencia de con la prueba anterior, esta usa un grupo de hilos llamada "Ultimate Thread Group" el cual ha de ser instalado adicionalmente [4]. Este está configurado de la forma explicada en el apartado 4.2.2 y los datos concretos en el ANEXO VIII.

Este grupo contiene una única petición HTTP la cual está dirigida a "*/test/phptestotal.php*", la página objetivo de esta prueba.El resto de partes, como el servidor al que hacer las peticiones como la recogida de datos están configurado de la misma manera que la explicada en el apartado 3.2.3

4.3. Análisis de resultados.

4.3.1. Preprocesado de los resultados

Al igual que con la primera parte, lo primero que intentamos fue sacar el período estacionario de cada nivel de concurrencia. Inicialmente el método para obtener los periodos es la mísma que la utilizada en el apartado 3.3.1, con la diferencia de que como esta prueba tiene los 5 niveles de concurrencia juntos, al tiempo de subida de cada nivel hay que sumarle todo el tiempo de la prueba que ya ha pasado.

Esté método no pudo ser utilizado porque en esta prueba el periodo de subida no coincida exactamente con el ramp up al tener los tiempos de respuesta tan elevado.

Esto implica que la forma de sacar el período estacionario para cada nivel de concurrencia se hará por truncamiento, los cinco períodos estacionarios pueden ser encontrados en el ANEXO VII

4.3.2. Análisis

Seguimos utilizando las herramientas del ANEXO IV para evaluar cada una de las pruebas que hemos realizado y obtener unos datos que nos caractericen la muestra.

Podemos observar, viendo los datos de la ANEXO VII que con cada aumento en el número de usuarios concurrentes los tiempos de respuesta se incrementan en el mismo orden.
En nuestro caso, vamos duplicando la cantidad de usuarios y los tiempos de respuesta se duplican igualmente excepto en la última prueba [Figura 11]. Esto se debe a que el sistema no es capaz de responder a las peticiones de todos los usuarios, comienzan a producirse fallos y los tiempos de respuesta de las peticiones fallidas (si estas llegasen a completarse) alcanzarían unos valores de un orden mucho mayor. Esto puede ser observado en [Figura 12]. También podemos destacar que a medida que aumenta el número de usuarios, es más frecuente la presencia de outliers, que hacen que las desviaciones típicas aumenten.

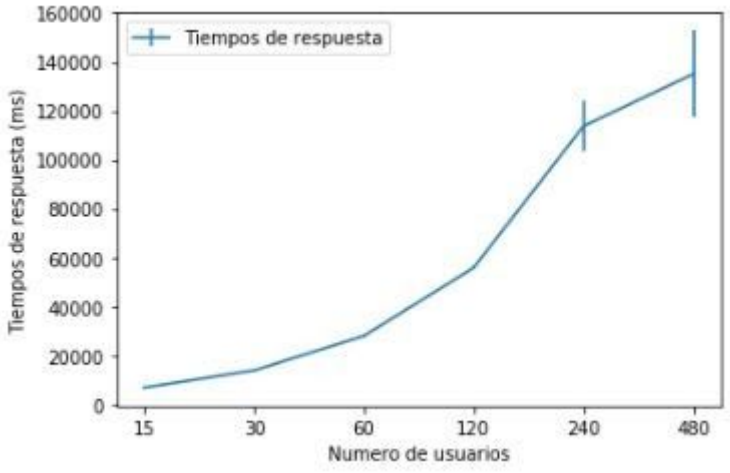

Figura 11 Tiempo de respuesta para cada nivel de concurrencia. Añadida la desviación típica como intervalo.

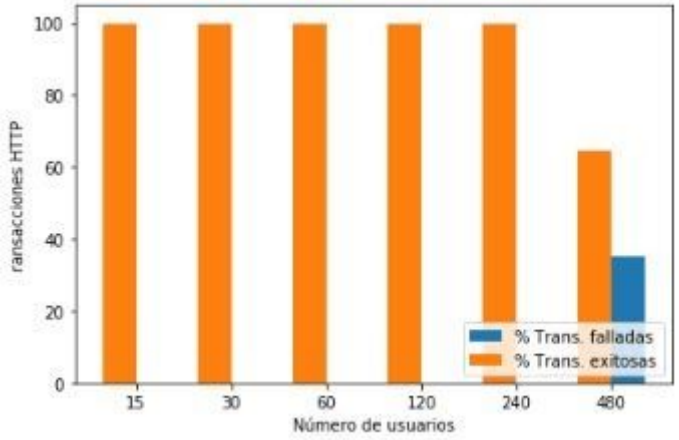

Figura 12 Porcentaje de transiciones exitosas y fallidas respecto al nivel de concurrencia.

En esta prueba nos encontramos con el sistema saturado por CPU [Figura 13] por lo cual la productividad en este sistema no va a poder incrementar, lo que conlleva que con los aumentos de usuarios concurrentes las diferencias en las métricas se verán en el tiempo de respuesta principalmente, como pasa con 480 usuarios concurrentes cuando el sistema corta conexiones y empieza a fallar.

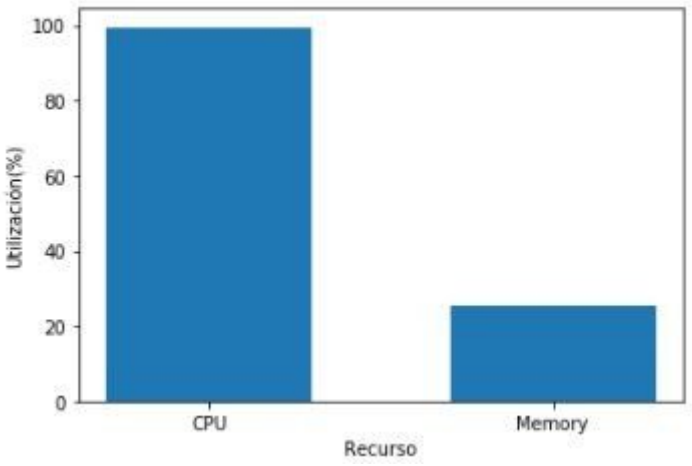

Figura 13 Utilización media de los recursos del sistema en todo el transcurso de la prueba, los valores de donde se obtiene están en ANEXO X

4.4. Conclusiones

Utilizando la información dada por los tiempos de respuesta y las de productividades, podemos concluir que el límite de capacidad del servidor web se encuentra entre los 240 y los 480 usuarios concurrentes, en el cual se dispararían los tiempos de respuesta si el propio sistema no rechazase parte de las peticiones al mismo.

Extrapolando, si observamos los porcentajes de aciertos y fallos con 480 usuarios concurrentes (64.76% y 35.24% respectivamente), podemos estimar que el sistema podría atender concurrentemente sin dar fallo a un total de 310 usuarios concurrentes. En este caso, cada usuario tendría un tiempo de respuesta similar al obtenido con los 480.

Otra conclusión obtenida consiste en que podemos prever que, si se aumenta el número de usuarios concurrentes, los tiempos de respuesta se mantendrán relativamente estables. Por contra, aumentará la proporción de fallos.

5. Evaluación del rendimiento de una aplicación web. (LRdto.PT3)

5.1. Descripción de la prueba

Esta práctica busca evaluar el rendimiento de una aplicación de gestión de inventarios basada en web [5], observando los tiempos de respuesta y la productividad, obtenidos de realizar una sesión personalizada. Esta sesión se realizará con varias intensidades de carga, las cuales se indicarán en el siguiente apartado 5.2.2.

5.1.1. Objetivos y restricciones

El objetivo principal de esta práctica es evaluar el rendimiento de una aplicación en ejecución mediante pruebas de carga. Por tanto, como objetivos parciales podríamos destacar los siguientes:

- Grabar un escenario válido para la prueba.
- Preparar los lanzamientos de las pruebas con JMeter usando los parámetros predefinidos en la prueba, y decidiendo los que se dejan a libre elección.
- Analizar las métricas de rendimiento propuestas en los siguientes apartados a fin de obtener una descripción real del comportamiento del sistema en términos de rendimiento.
- Identificar que elemento de la sesión cuenta con los mayores tiempos de respuesta, es decir, encontrar el elemento determinante de la actuación del sistema.

Como objetivos de aprendizaje mencionaremos los siguientes:

- Aprendizaje de UML2 para describir el diagrama de actividades del escenario de prueba.
- Utilización de JMeter para crear un escenario de prueba.

Como restricciones, a mayores de las mencionadas en puntos anteriores podemos destacar:

- El uso de Firefox debido a que facilita la realización del escenario de prueba.

5.1.2.Ajustes en la configuración del entorno de pruebas

Para llevar el desarrollo de la prueba a buen término, ha sido necesario, por un lado, modificar la configuración anterior de JMeter, añadiendo un Proxy y su puerto correspondiente, y por otro lado, instalar un certificado en Firefox. Tras la grabación del escenario, se ha requerido hacer una limpieza de todas las transacciones grabadas, manteniendo sólo aquellas relevantes para el mismo.

5.2.Diseño de la prueba

5.2.1. Métricas (definición y unidades)

Las métricas a utilizar serán las mismas que en la prueba anterior. Estas están explicadas en el apartado 3.2.1.

5.2.2. Especificación de la carga de trabajo

Es esta práctica tenemos que hacer un análisis de carga estacionaria, para ello el escenario para la aplicación Stock elaborado por el grupo se ejecutará con los siguientes niveles de usuarios concurrentes, 40, 80 y 160.

Se fijará un tiempo de pensar de 2 segundos entre cada petición HTTP simulando el comportamiento natural de un cliente que pudiese utilizar los servicios de la página.

Para el tiempo de subida utilizamos una cantidad de tres usuarios por segundo. Por lo tanto los tiempos de subida quedarían en 13, 26 y 52 segundos para cada nivel de usuarios.

La duración del periodo de medida será de 600 segundos excluyendo el periodo de subida.

Para cada nivel, se ejecutará la prueba tres veces en momentos de tiempo distintos durante el día, para así evitar posibles interferencias debidas al estado del sistema.

5.2.3.Descripción del plan de pruebas.

Estas pruebas han sido generadas utilizando la plantilla de recording [6].

El escenario utilizado en estas pruebas es el expuesto en el enunciado de la práctica, el cual es:

1. El usuario se conecta al sistema (login)
2. El sistema muestra el panel de control principal
3. El usuario selecciona “Productos”
4. El sistema muestra un listado de los productos disponibles
5. El usuario selecciona “Editar” para un producto
6. El sistema muestra la ficha del producto
7. El usuario modifica el precio y el stock del producto y confirma el cambio
8. El sistema guarda los cambios y confirma que el cambio ha sido realizado.
9. El usuario selecciona Inicio y se desconecta del sistema (logout)
10. El sistema muestra la página principal de la aplicación

Siendo recogido este escenario en el siguiente diagrama de casos de uso [Figura 14]

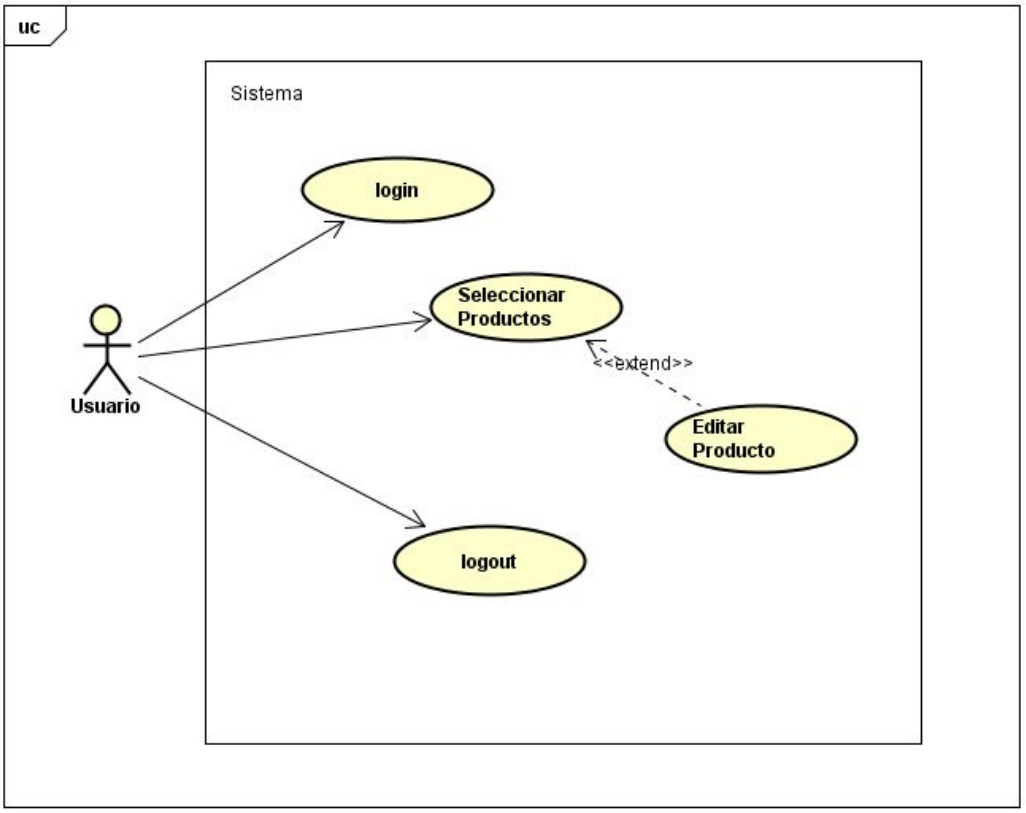

Figura 14 Diagrama de casos de uso del escenario Actualización.

La descripción detallada de cada caso de uso la encontramos en las figuras del ANEXO XI.

Este escenario se convierte en un total de 11 peticiones HTTP. En el ANEXO III está la estructura del plan de prueba de JMeter, aun así a continuación será explicada la estructura y su configuración.

Este plan parte de una configuración estándar, definiendo el nombre del servidor, el protocolo, y la utilización de cookies, estas últimas necesarias para mantener iniciada una sesión y así poder realizar modificaciones en el stock.

Lo siguiente que encontramos es el grupo de hilos; en este caso volvemos a utilizar el grupo de hilos básico con la configuración basada en la sintaxis `${__P(<<PARAM>>,300)}` ya explicada en el apartado 3.2.3.

Dentro de este grupo de hilos nos encontramos con todas las peticiones HTTP necesarias para llevar a cabo el escenario. Para facilitar la obtención de los tiempos de respuesta, podemos dividir estas peticiones en tres bloques de controladores de transacción:

- Login
  Tiene solo una petición HTTP la cual es el inicio de sesión pasando el nombre y la contraseña adecuados.
- Transaction
  Compone el grueso del plan de pruebas. Tiene ocho peticiones HTTP, las cuales comprenden de los puntos 2 al 8 del escenario expuesto al inicio de este apartado.
- Logout
  Contiene las dos últimas peticiones HTTP las cuales son cerrar sesión y la posterior carga de la página de iniciar sesión.

Por último, en el grupo de hilos tenemos un temporizador, en este caso "Temporizador Constante", que asigna un tiempo de pensar entre cada petición.

Para finalizar el plan de pruebas está añadido el receptor de PerfMon que permite recabar medidas sobre el estado del sistema de prueba durante la ejecución. Las métricas que usaremos son las mismas que las explicadas en el apartado 3.2.3

5.3.Análisis de resultados

5.3.1.Preprocesado de los resultados

Nos encontramos en una situación parecida con el primer escenario, tenemos tres repeticiones para cada prueba, por lo tanto volvemos a aplicar el mismo procedimiento (apartado 3.3.1) hasta llegar a tener la media, varianza e intervalo de confianza tanto de productividad como de tiempo de respuesta para cada nivel de concurrencia.

Una diferencia con el método explicado en el apartado 3.3.1 es que en estas pruebas tenemos varias transacciones, por lo tanto, para cada nivel de concurrencia se sacarán la media, desviación estándar e intervalo de confianza de:

- Sesión:
  Es la suma de los controladores de Login y Logout descritos en el apartado anterior.
- Transacción:
  Es el controlador Transaction descrito en el apartado anterior.
- Total:
  Es la suma de los tres controladores

Los tiempos de respuesta para un caso en concreto de los componentes de Transacción se pueden ver en ANEXO XII (Figura 39).

5.3.2. Análisis

Analizando los resultados obtenidos de la ejecución de este escenario, podemos visualizar el comportamiento de la web de stock a través de los siguientes gráficos y quedan recogidos los datos en las tablas del ANEXO XIII.

La [Figura 15] nos muestra el tiempo de respuesta total medio por nivel de usuario concurrentes, aumentando un 16.94% en el segundo nivel respecto al primero, y un 546% del segundo nivel al tercero (datos sacados de las tablas 8, 9 y 10 del ANEXO XIII).

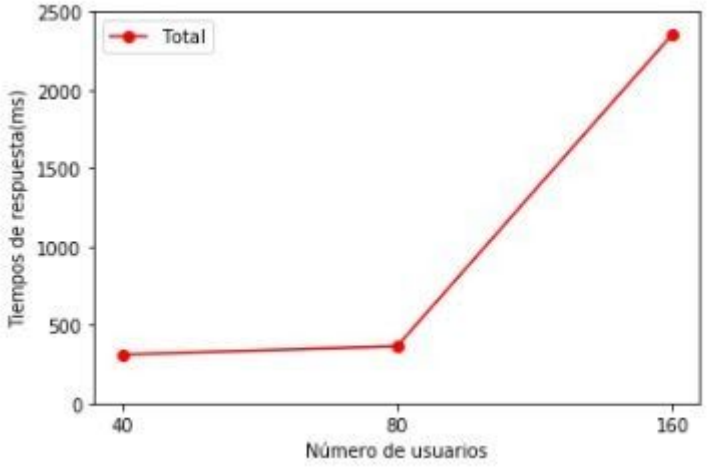

Figura 15 Tiempo de respuesta total medio por nivel de usuarios concurrentes

Este incremento de los tiempos de respuesta al alcanzar los 160 usuarios nos puede indicar que, pese a que la CPU está siendo utilizada alrededor del 60% del tiempo y no esté saturada [Figura 16], puede que sea el gestor de la base de datos el que esté llegando a saturarse cuando alcanzamos un número de peticiones entre los niveles 2 y 3. Aunque no tenemos hechos que nos permitan comprobar esta hipótesis.
También cabe decir que el crecimiento de la utilización tanto de CPU como de memoria es prácticamente lineal respecto al nivel de concurrencia.

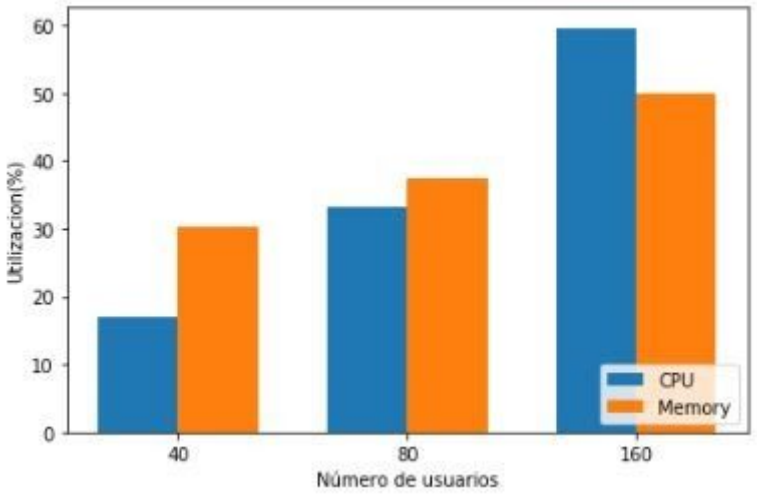

Figura 16 Utilización recursos del sistema para cada nivel de concurrencia

Observando la gráfica de la [Figura 17] vemos como las transacciones aumentan en cada nivel de concurrencia, aunque no aumentan al mismo ritmo que la cantidad de usuarios concurrentes. Esto junto con el incremento que pueden observarse en los tiempos de respuesta de la [Figura 15] denota que el sistema tiende a un punto en el que no puedan satisfacerse las peticiones en un tiempo de respuesta razonable.

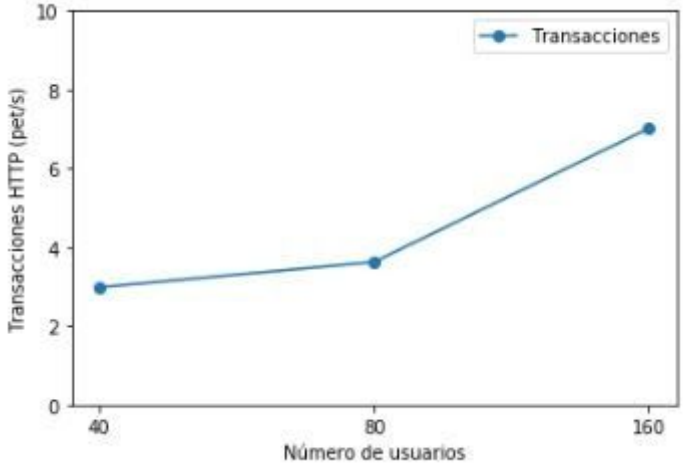

Figura 17 Productividad por nivel de concurrencia

Separando la transacción total en sus dos componentes principales, sesión y transacción [Figura 18] (o bien los datos mostrados en las tablas del ANEXO XIII), podemos comprobar que el tiempo de respuesta usado por el bloque de transacciones es notablemente mayor al del bloque de sesión. Esto es algo razonable, puesto que el bloque de transacciones contiene unas cuantas peticiones HTTP más además de que estas tienen más tiempo de respuesta como se puede ver en el ANEXO XIII.

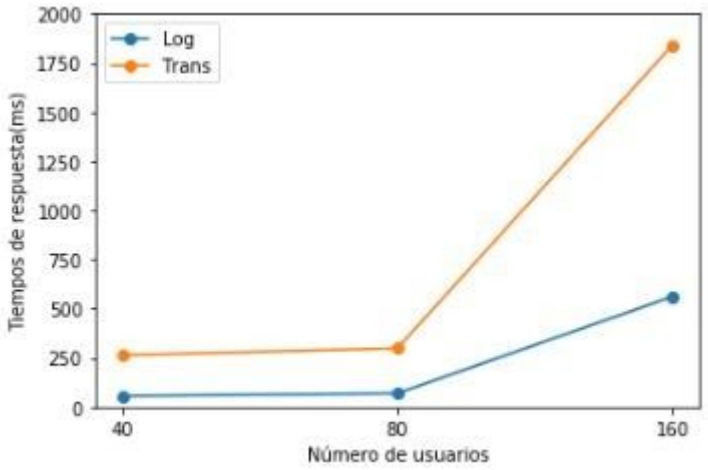

Figura 18 Tiempo de respuesta medio para cada transacción por nivel de usuario concurrentes

Además, a medida que va aumentando el número de usuarios concurrentes, la proporción del tiempo de respuesta debido al bloque de transacciones frente al total también va disminuyendo, pasando de ser el 84% con 40 usuarios a 81% con 80 y finalmente, 78,2% con 160, esto puede deberse al mismo problema citado anteriormente, una saturación de la base de datos, al tener que hacer login, esto sigue siendo una hipótesis para la cual no tenemos hechos que la corroboren.

Al analizar los tiempos de respuesta de las diversas peticiones dentro del apartado Transacción, podemos observar que la petición que mayores tiempos de respuesta requiere es la destinada a mostrar el conjunto de productos disponibles como se puede apreciar en la petición "*/stock/php-action/fetchProduct.php-70*" en la figura 39 del ANEXO XII.

Por último si analizamos las transacciones para cada agrupación por segundo de toda la prueba nos encontramos con que, primero, son prácticamente iguales al tratarse de una secuencia que se repite constantemente y, segundo nos encontramos con un patrón de picos de transacciones que se repiten aproximadamente cada 20..30 segundos, este tiempo coincide aproximadamente con el tiempo que tarda el caso de uso en completarse, se tienen 11 peticiones HTTP con un tiempo de pensar de 2 segundos, a estos 22 segundos habría que añadir los tiempos de respuesta que para el caso de 160 usuarios son alrededor de 2.35 segundos dando un total de 24.35 segundos.

Esto nos indica que la mayoría de hilos han acabado yendo al unísono y puede ser otro de los indicadores de la gran diferencia de tiempo de respuesta respecto a 80 usuarios concurrentes.

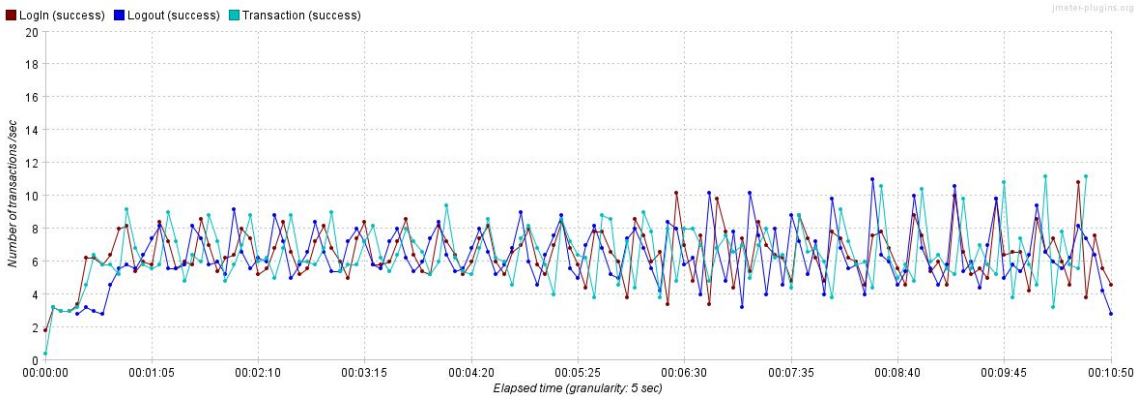

Figura 19 Transacciones por segundo para 160 usuarios

5.4. Conclusiones

Teniendo en cuenta cómo aumenta el uso de la CPU, y cómo aumentan los tiempos de respuesta (podemos deducir del análisis anterior que estos irán aumentando de forma significativa), concluimos que si seguimos aumentando el nivel de concurrencia, el servidor acabará colapsando.

Además a partir de 160 usuarios existe un incremento anómalo en los tiempos de respuesta que con los datos obtenidos no existe una conclusión clara de su porqué, una de nuestra hipótesis es saturación de la base de datos.

6.Conclusiones generales

En las diferentes prácticas se han realizado diferentes tipos de pruebas en función de qué quería saberse acerca de los sistemas en estudio. Hemos entendido que, tanto las pruebas que se ejecutan como la información que se recopila y analiza, varían en función de cuáles son los objetivos en el estudio de los sistemas.

En la primera práctica, que se basa en dos pruebas de carga, nos ha interesado analizar las métricas del sistema de forma que pudiéramos extrapolar si existen diferencias en el rendimiento del sistema en función del recurso solicitado, concluyendo que este no es un factor determinante.

En la segunda práctica, que se basa en una prueba de “estrés” a la máquina, hemos provocado la situación en la que el sistema ha llegado a un punto de fallo y hemos analizado cuál es ese punto y qué elementos del sistema generan que se llegue a esa situación, concluyendo que es la CPU la que limita el comportamiento del sistema una vez los usuarios superan la cifra de los 310 usuarios concurrentes.

En la tercera práctica, que se basa en una carga de trabajo más compleja y cercana a una situación real, hemos analizado el comportamiento del servidor y cuáles de las etapas de la prueba influyen más en este comportamiento.

Como conclusión final, el análisis de los sistemas debe plantearse, realizarse y analizarse teniendo en cuenta el sistema analizado y que se busca conocer en él.

7. Bibliografía

[1] X. Molero, C. Juiz, and M. Rodeño, Evaluación y Modelado del Rendimiento de los Sistemas Informáticos. Pearson Educación S.A., 2004. [Online]. Available: http://www.pearsoneducacion.com/molero/
[2] Apache JMeter - apache JMeterTM. [Online]. Available: http://jmeter.apache.org/
[3] Perfmon server agent. [Online]. Available: https://jmeter-plugins.org/wiki/PerfMonAgent/
[4] Documentation Ultimate Thread Group. [Online]. Available: https://jmeter-plugins.org/wiki/UltimateThreadGroup/
[5] Sistemas de gestión de inventario con PHP - Sistemas web. [Online]. Available: https://obedalvarado.pw/blog/sistema-gestion-inventario-php/
[6] Apache JMeter HTTP(S) Test Script Recorder . [Online]. Available: https://jmeter.apache.org/usermanual/jmeter_proxy_step_by_step.html
[7] "E1 - INTRODUCCIÓN (I)", class notes for Evaluación y Rendimiento Software, Informática, Universidad de Valladolid, 2020. [Online]. Available: https://aulas.inf.uva.es/pluginfile.php/39722/mod_resource/content/5/T1_ERSS_IntroRdto_1920.pdf
[8] pandas - Python Data Analysis Library. [Online]. Available: https://pandas.pydata.org/
[9] Matplotlib: Python plotting — Matplotlib 3.2.1 documentation. [Online]. Available: https://matplotlib.org/
[10] NumPy. [Online]. Available: https://numpy.org/

8. Anexos

8.1. ANEXO I

Comando para obtener el nombre, versión y otros detalles del sistema operativo:

```
$ uname -a

Linux virtual 4.15.0-101-generic #102-Ubuntu SMP Mon May 11 10:07:26 UTC
2020 x86_64 x86_64 x86_64 GNU/Linux
```

Comando para mostrar información sobre los elementos hardware del sistema:

```
$ lscpu

Architecture:        x86_64
CPU op-mode(s):      32-bit, 64-bit
Byte Order:          Little Endian
CPU(s):              2
On-line CPU(s) list: 0,1
Thread(s) per core:  1
Core(s) per socket:  2
Socket(s):           1
NUMA node(s):        1
Vendor ID:           GenuineIntel
CPU family:          15
Model:               6
Model name:          Common KVM processor
Stepping:            1
CPU MHz:             2394.270
BogoMIPS:            4788.54
Hypervisor vendor:   KVM
Virtualization type: full
L1d cache:           32K
L1i cache:           32K
L2 cache:            4096K
L3 cache:            16384K
NUMA node0 CPU(s):   0,1
Flags:                 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx lm
constant_tsc nopl xtopology cpuid tsc_known_freq pni cx16 pcid x2apic
hypervisor lahf_lm cpuid_fault pti
```

Comando para mostrar uso del disco duro, específicamente el espacio en unidades comunes del sistema:

```
$ df -h

Filesystem      Size  Used Avail Use% Mounted on
udev            966M     0  966M   0% /dev
tmpfs            97M  5,8M   92M   6% /run
```

```
/dev/vda2       9,8G  5,1G  4,2G  55% /
tmpfs           485M     0  485M   0% /dev/shm
tmpfs           5,0M     0  5,0M   0% /run/lock
tmpfs           485M     0  485M   0% /sys/fs/cgroup
/dev/loop0       94M   94M     0 100% /snap/core/9066
/dev/loop1       94M   94M     0 100% /snap/core/8935
tmpfs           117M     0  117M   0% /run/user/1000
```

Comando para mostrar la memoria y swap del sistema:

```
$ free -h

              total        used        free      shared  buff/cache   available
Mem:           1,9G        343M        894M        7,3M        754M        1,4G
Swap:          2,0G          0B        2,0G
```

8.2. ANEXO II

Comando para obtener el nombre, versión y otros detalles del sistema operativo:

```
$ uname -a

Linux jair 5.4.28-gentoo-eii #1 SMP Sun Apr 5 16:02:08 CEST 2020 x86_64
Common KVM processor GenuineIntel GNU/Linux
```

Comando para mostrar información sobre los elementos hardware del sistema:

```
$ lscpu

Arquitectura:                        x86_64
modo(s) de operación de las CPUs:    32-bit, 64-bit
Orden de los bytes:                  Little Endian
Tamaños de las direcciones:          40 bits physical, 48 bits virtual
CPU(s):                              2
Lista de la(s) CPU(s) en línea:      0,1
Hilo(s) de procesamiento por núcleo: 1
Núcleo(s) por «socket»:              2
«Socket(s)»                          1
Modo(s) NUMA:                        1
ID de fabricante:                    GenuineIntel
Familia de CPU:                      15
Modelo:                              6
Nombre del modelo:                   Common KVM processor
Revisión:                            1
CPU MHz:                             2194.898
BogoMIPS:                            4389.79
Fabricante del hipervisor:           KVM
Tipo de virtualización:              lleno
Caché L1d:                           64 KiB
Caché L1i:                           64 KiB
Caché L2:                            8 MiB
Caché L3:                            16 MiB
CPU(s) del nodo NUMA 0:              0,1
Indicadores:                          fpu vme de pse tsc msr pae mce cx8
apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht
syscall nx lm constant_tsc nopl xtopology cpuid pni cx16 pcid x2apic
hypervisor lahf_lm cpuid_fault pti
```

Comando para mostrar uso del disco duro, específicamente el espacio en unidades comunes del sistema:

```
$ df -h

S.ficheros            Tipo     Tamaño Usados  Disp Uso% Montado en
/dev/root             ext4       9,5G   4,2G  4,8G  47% /
devtmpfs              devtmpfs    10M      0   10M   0% /dev
tmpfs                 tmpfs      394M   2,0M  392M   1% /run
cgroup_root           tmpfs       10M      0   10M   0% /sys/fs/cgroup
```

```
shm                     tmpfs       2,0G      0  2,0G   0% /dev/shm
```

Comando para mostrar la memoria y swap del sistema:

```
$ free -h

               total        used        free      shared  buff/cache   available
Mem:           3,8Gi       144Mi       3,5Gi       1,0Mi       249Mi       3,6Gi
Swap:          255Mi          0B       255Mi
```

8.3. ANEXO III

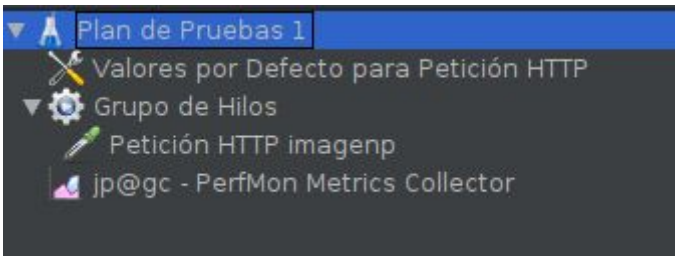

Figura 20 Visión general del plan de pruebas del primer escenario

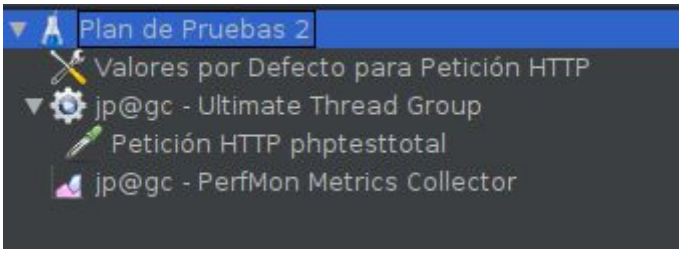

Figura 21 Visión general del plan de pruebas del segundo escenario

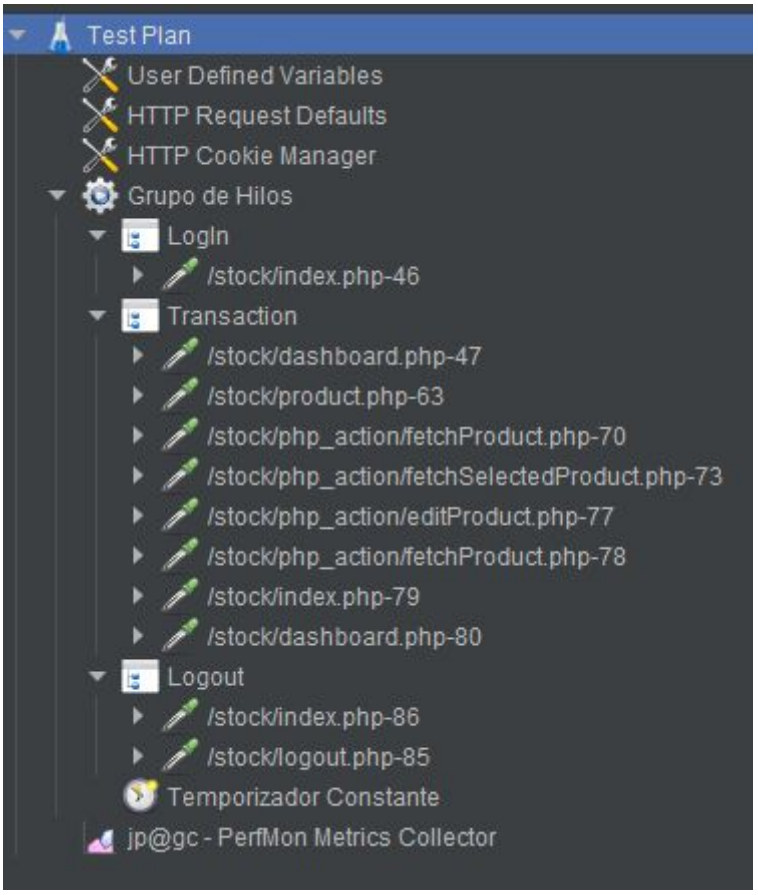

Figura 22 Visión general del plan de pruebas del tercer escenario

8.4. ANEXO IV

```
def ic_media(med_glob, var_meds, m, t):
    tmp = t*(var_meds/m)**(1/2)
    min = med_glob - tmp
    max = med_glob + tmp
    return [min,max]
```

Figura 23. Código para la obtención del intervalo de confianza de la media.

```
def est_var_medias(meds, valor_t):
    m = len(meds)
    # Media global
    med_glob = 0
    for media in meds:
        med_glob += media
    med_glob /= m

    # Varianza de la media global
    var = 0
    for media  in meds:
        var += (media - med_glob)**2
    var /= m - 1

    # Calculo del intervalo de confianza para la media
    ic = ic_media(med_glob, var, m, valor_t)

    # Resultado
    return med_glob, var, ic
```

Figura 24. Código para la obtención de las medias, varianzas y el intervalo de confianza

```
labels = ['40', '80', '160']
ycpu = [
    archivo40["virtual.lab.inf.uva.es CPU system"].mean() + archivo40["virtual.lab.inf.uva.es CPU user"].mean(),
    archivo80["virtual.lab.inf.uva.es CPU system"].mean() + archivo80["virtual.lab.inf.uva.es CPU user"].mean(),
    archivo160["virtual.lab.inf.uva.es CPU system"].mean() + archivo160["virtual.lab.inf.uva.es CPU user"].mean()
]

ymem = [
    archivo40["virtual.lab.inf.uva.es Memory"].mean(),
    archivo80["virtual.lab.inf.uva.es Memory"].mean(),
    archivo160["virtual.lab.inf.uva.es Memory"].mean()
]

x = np.arange(len(labels))  # the label locations
width = 0.35  # the width of the bars

fig, ax = plt.subplots()
rects1 = ax.bar(x - width/2, ycpu, width, label='CPU')
rects2 = ax.bar(x + width/2, ymem, width, label='Memory')

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Utilizacion(%)')
ax.set_xlabel('Número de usuarios')
ax.set_xticks(x)
ax.set_xticklabels(labels)
ax.legend(loc = "lower right")


fig.tight_layout()

plt.show()
```

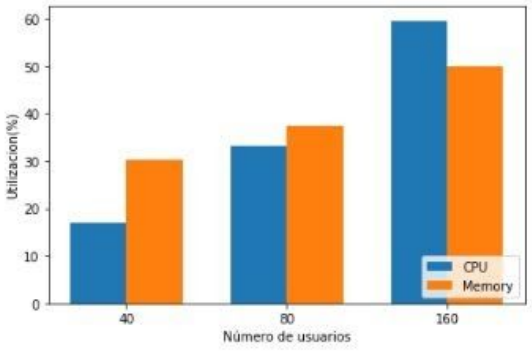

Figura 25. Visualización de un gráfico de barras con “Matplotlib”

```
a1 = pd.read_csv("C:/Users/IGNACIO/Desktop/2019-2020-ERSS/Practica Grupo/Parte2/respTime15.csv", sep=";")
a2 = pd.read_csv("C:/Users/IGNACIO/Desktop/2019-2020-ERSS/Practica Grupo/Parte2/respTime30.csv", sep=";")
a3 = pd.read_csv("C:/Users/IGNACIO/Desktop/2019-2020-ERSS/Practica Grupo/Parte2/respTime60.csv", sep=";")
a4 = pd.read_csv("C:/Users/IGNACIO/Desktop/2019-2020-ERSS/Practica Grupo/Parte2/respTime120.csv", sep=";")
a5 = pd.read_csv("C:/Users/IGNACIO/Desktop/2019-2020-ERSS/Practica Grupo/Parte2/respTime240.csv", sep=";")
a6 = pd.read_csv("C:/Users/IGNACIO/Desktop/2019-2020-ERSS/Practica Grupo/Parte2/respTime480.csv", sep=";")
```

```
y = [
    a1["Petición HTTP phptesttotal"].mean(),
    a2["Petición HTTP phptesttotal"].mean(),
    a3["Petición HTTP phptesttotal"].mean(),
    a4["Petición HTTP phptesttotal"].mean(),
    a5["Petición HTTP phptesttotal"].mean(),
    a6["Petición HTTP phptesttotal"].mean()
]

ydev = [
    a1["Petición HTTP phptesttotal"].std(),
    a2["Petición HTTP phptesttotal"].std(),
    a3["Petición HTTP phptesttotal"].std(),
    a4["Petición HTTP phptesttotal"].std(),
    a5["Petición HTTP phptesttotal"].std(),
    a6["Petición HTTP phptesttotal"].std()
]

x = ["15","30","60","120","240","480"]

plt.xlabel("Numero de usuarios")
plt.ylabel("Tiempos de respuesta (ms)")

plt.errorbar(x, y, label="Tiempos de respuesta", yerr=ydev)

plt.legend(loc = "upper left")

plt.show()
```

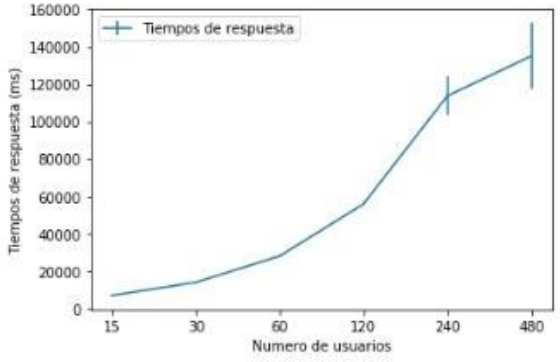

Figura 26. Visualización de una gráfica básica con la herramienta “Matplotlib”

8.5. ANEXO V

	75 usuarios	150 usuario	300 usuarios
Media (ms)	11.043659	23.042350	50.421562
IC^{95} Media (ms)	(7.47303; 14.61428)	(20.26827; 25.81642)	(40.30640; 60.53672)
Varianza (ms)	2.065985	1.247023	16.580001

Tabla 1 Tiempo de respuesta del test sobre la imagen pequeña

	75 usuarios	150 usuario	300 usuarios
Media (ms)	12.269352	22.676476	46.975176
IC^{95} Media (ms)	(7.49952; 17.03918)	(16.50086;28.85209)	(35.88236;58.06798)
Varianza(ms)	3.686753	6.180158	19.939864

Tabla 2 Tiempo de respuesta del test sobre la imagen grande

	75 usuarios	150 usuario	300 usuarios
Media (peticiones/s)	7617.911	7348.295	7130.746
IC^{95} Media (peticiones/s)	(5161.003; 10074.819)	(4917.992; 9778.598)	(4659.299; 9602.193)
Varianza (peticiones/s)	978175.611	957105.881	989786.784

Tabla 3 Productividad del test sobre la imagen pequeña

	75 usuarios	150 usuario	300 usuarios
Media (peticiones/s)	7547.762	6808.333	6689.545
IC^{95} Media (peticiones/s)	(5079.135; 10016.387)	(4304.136; 9312.529)	(4537.630; 8841.459)
Varianza (peticiones/s)	987528.777	1016192.189	750393.769

Tabla 4 Productividad del test sobre la imagen grande

8.6. ANEXO VI

	Intervalo de Confianza 95 para tiempo de respuesta (ms)
75 usuarios	(-7.183; 5.958)
150 usuarios	(-21.166 ;18.798)
300 usuarios	(-11.566; 18.458)

Tabla 5 Intervalos de confianza para la diferencia entre tiempos de respuesta

	Intervalo de Confianza 95 para la productividad (peticiones/s)
75 usuarios	(-3412.739; 3553.037)
150 usuarios	(-2949.647; 4029.571)
300 usuarios	(-2835.808; 3718.210)

Tabla 6 Intervalos de confianza para la diferencia entre productividades

8.7. ANEXO VII

	15	30	60	120	240	480
Media (ms)	6818.04554	13615.68725	26894.75297	53885.02083	103748.7044	135076.417729
IC^{95} Media (ms)	(6785.9128; 6850.1782)	(13569.4270; 13661.9474)	(26839.5100; 26949.9959)	(53803.2451; 53966.7965)	(102325.9164; 105171.4921)	(132291.156; 137861.679)
Varianza (ms)	213.49066	305.53600	362.68083	523.74280	10382.88894	17724.983904

Tabla 7 Tiempos de respuesta del segundo escenario

8.8. ANEXO VIII

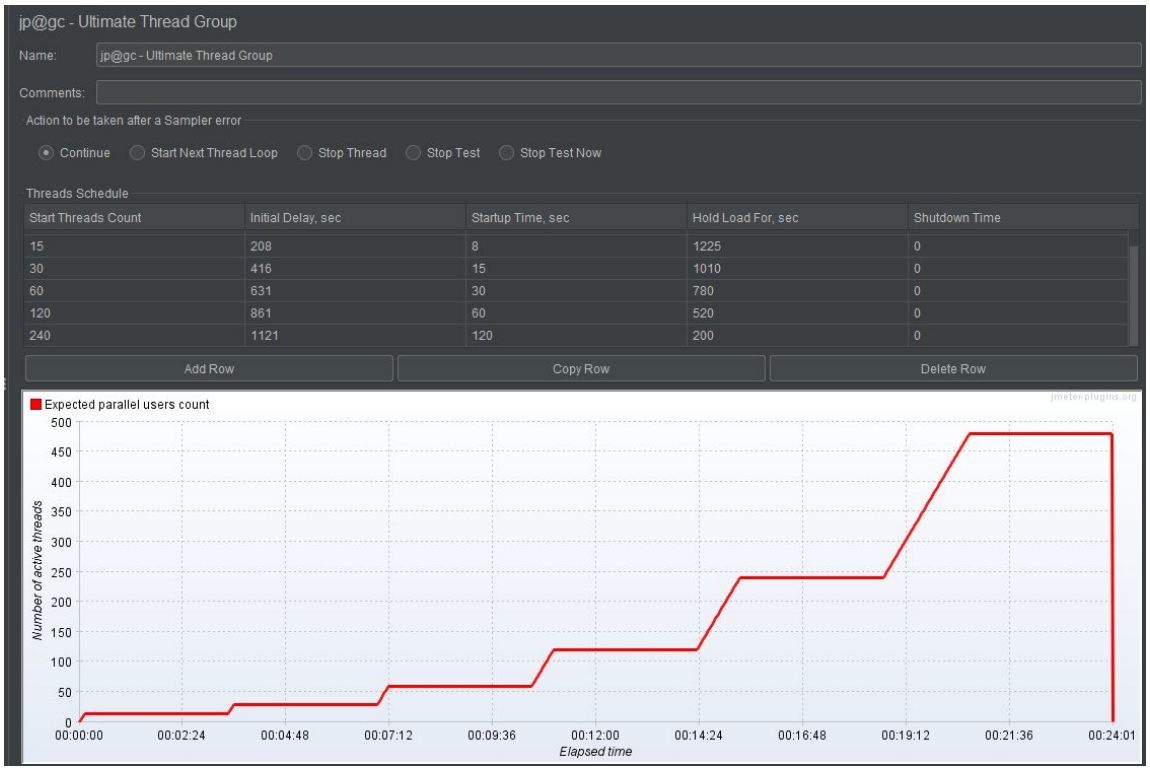

Figura 27 Configuración utilizada en la prueba dos para la gestión de Ultimate thread group

8.9. ANEXO IX

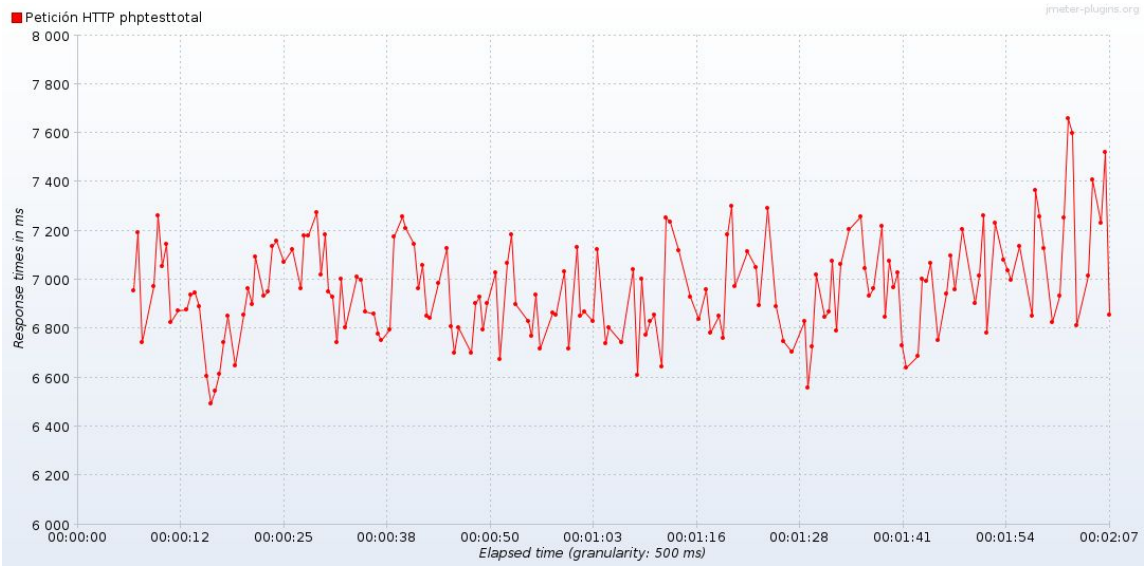

Figura 28 Tiempos de respuesta con 15 usuarios concurrentes

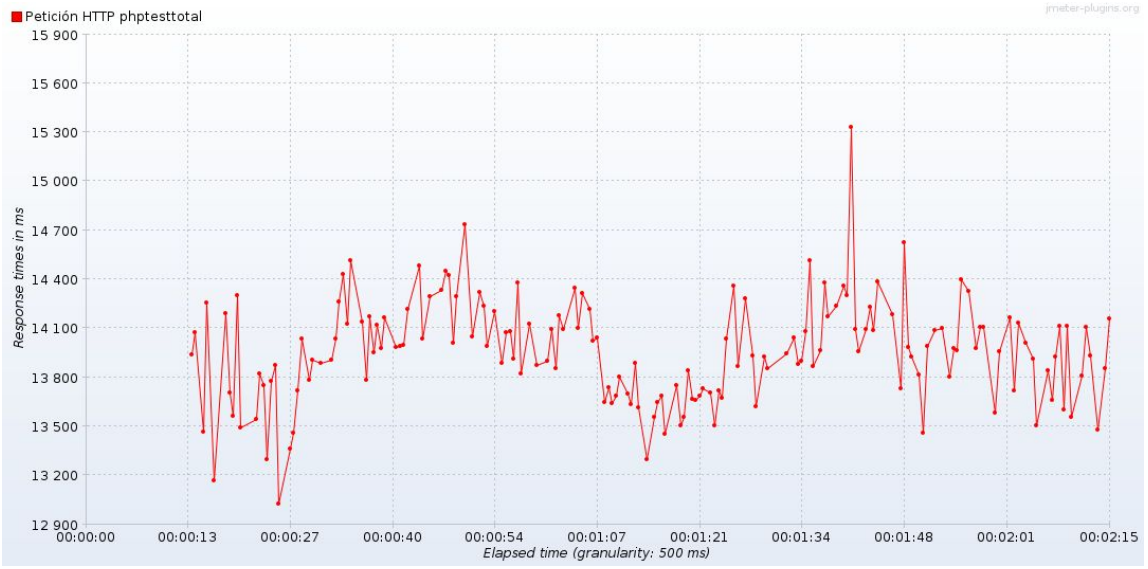

Figura 29 Tiempos de respuesta con 30 usuarios concurrentes

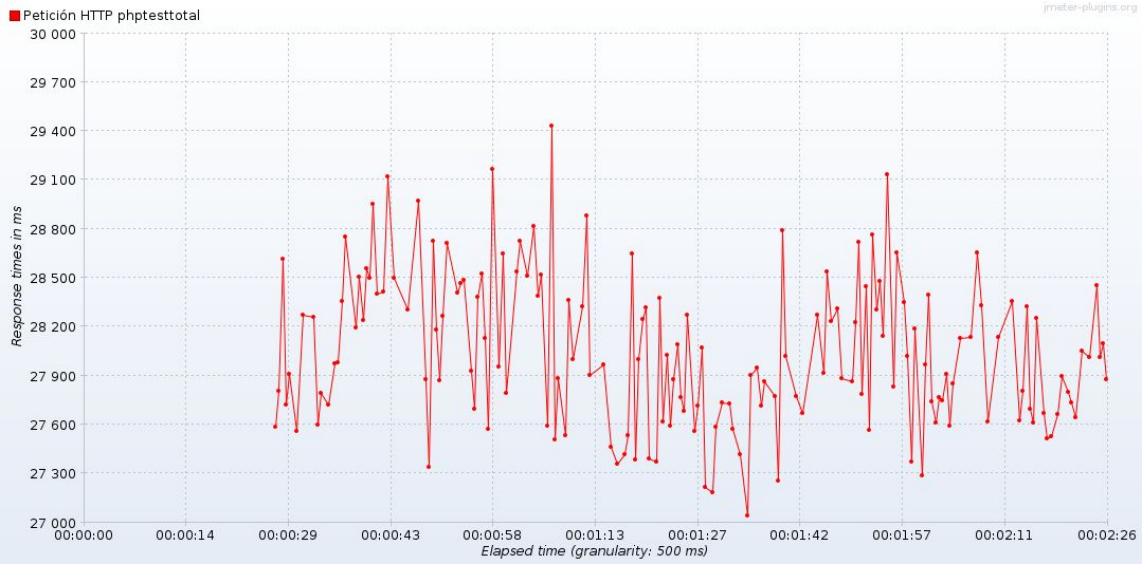

Figura 30 Tiempos de respuesta con 60 usuarios concurrentes

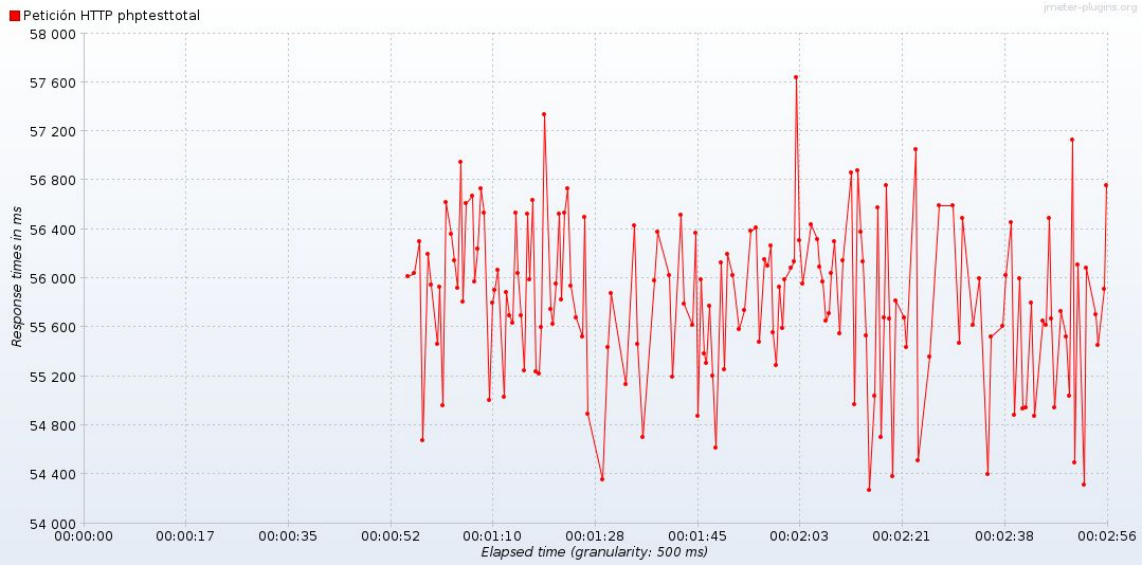

Figura 31 Tiempos de respuesta con 120 usuarios concurrentes

Figura 32 Tiempos de respuesta con 240 usuarios concurrentes

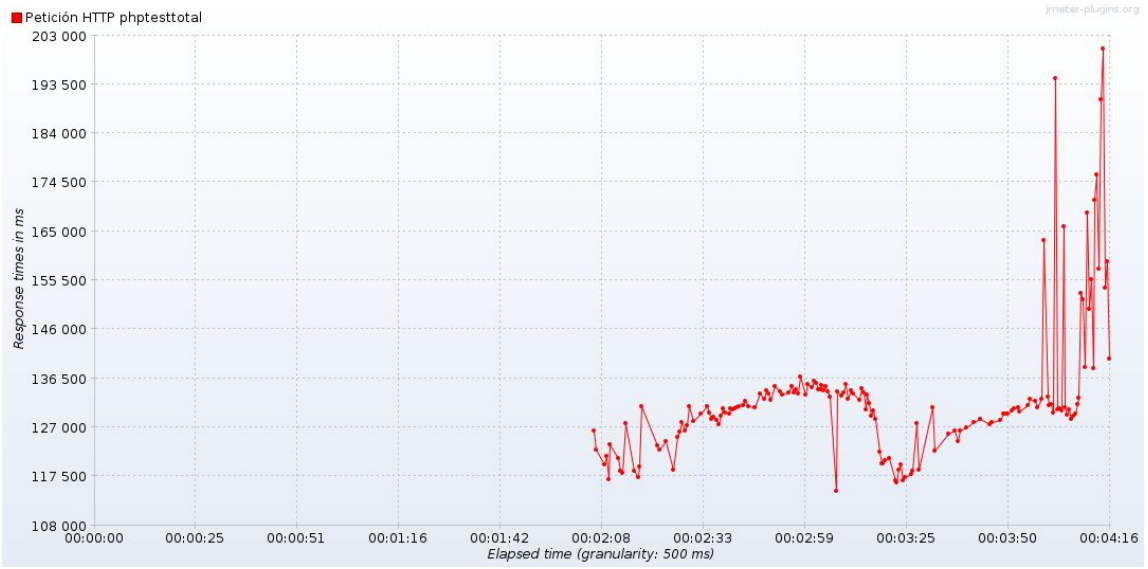

Figura 33 Tiempos de respuesta con 480 usuarios concurrentes

8.10. ANEXO X

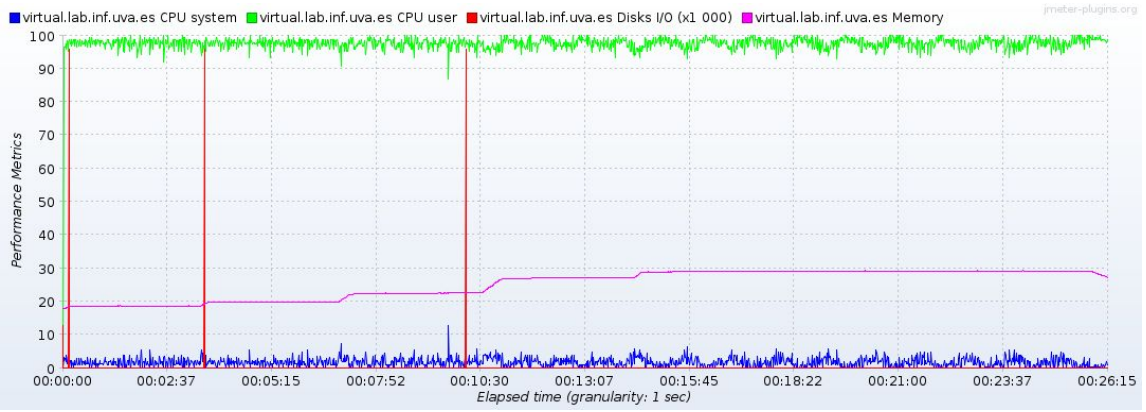

Figura 34 Utilización a lo largo del tiempo de los recursos del sistema para la práctica dos

8.11. ANEXO XI

login / UseCase Description

ITEM	VALUE
UseCase	login
Summary	Conectarse al sistema
Actor	Usuario
Precondition	El usuario no está conectado en el sistema. El usuario está registrado en el sistema
Postcondition	El usuario ha iniciado sesión en la aplicación.
Base Sequence	· El usuario introduce nombre y contraseña y da a ingresar. · El sistema comprueba que sea correcto y muestra el panel de control principal.

Figura 35 Descripción del caso de uso Login.

Seleccionar Productos / UseCase Description

ITEM	VALUE
UseCase	Seleccionar Productos
Summary	
Actor	Usuario
Precondition	El usuario ha iniciado sesión en el sistema
Postcondition	
Base Sequence	El usuario selecciónа productos. El sistema muestra la lista de productos disponible.

Figura 36 Descripción del caso de uso Seleccionar Productos.

Editar Producto / UseCase Description	
ITEM	VALUE
UseCase	Editar Producto
Summary	El usuario quiere editar el peso y el stock de un producto.
Actor	
Precondition	El usuario se encuentra loggeado en el sistema.
Postcondition	Se han modificado los datos de precio y stock de un producto
Base Sequence	· El usuario selecciona acción. · El sistema muestra las opciones de editar y eliminar. · El usuario selecciona editar. · El sistema muestra la ficha del producto. · El usuario selecciona Información del producto · El sistema muestra los datos del producto. · El usuario edita el precio y el stock del producto y confirma el cambio. · El sistema guarda los cambios y confirma que el cambio ha sido realizado. · El usuario cierra la ventana de editar producto. · El sistema cierra la ventana y el caso de uso finaliza.

Figura 37 Descripción del caso de uso Editar Producto.

logout / UseCase Description	
ITEM	VALUE
UseCase	logout
Summary	El usuario se desconecta de la aplicación.
Actor	Usuario
Precondition	El usuario está conectado a la aplicación.
Postcondition	El usuario no está conectado a la aplicación.
Base Sequence	· El usuario selecciona el desplegable del navbar. · El sistema despliega las opciones de configuración y salir. · El usuario selecciona salir. · El sistema cierra sesion y se vuelve a la ventana de login y el caso de uso finaliza.

Figura 38 Descripción del caso de uso Logout.

8.12.ANEXO XII

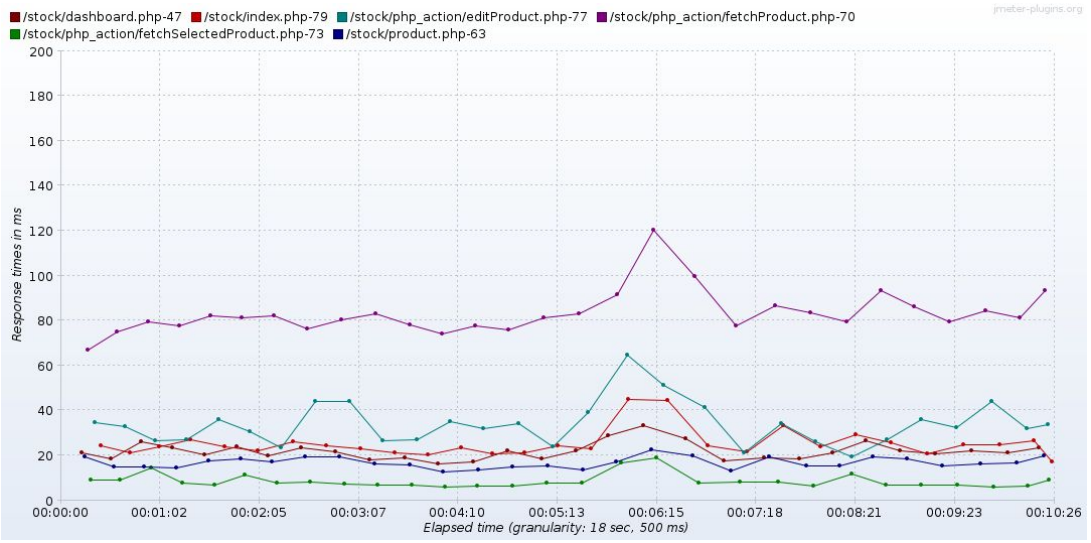

Figura 39 Tiempos de respuesta de la transacción, para 80 usuarios concurrentes.

8.13. ANEXO XIII

	40 usuarios		
	Sesión	Transacción	Total
Media (ms)	55.63469	262.70386	311.02948
IC^{95} Media (ms)	(33.5600; 77.70932)	(176.3143; 349.0933)	(204.0386; 418.0203)
Varianza(ms)	78.96347	1209.37474	1854.95039

Tabla 8 Tiempo de respuesta para las transacciones de la práctica 3 con 40 usuarios concurrentes

	80 usuarios		
	Sesión	Transacción	Total
Media (ms)	68.83641	298.21877	363.71498
IC^{95} Media (ms)	(54.1566; 83.5161)	(251.9868; 344.4506)	(309.8570; 417.5729)
Varianza(ms)	34.92016	346.35633	470.044049

Tabla 9 Tiempo de respuesta para las transacciones de la práctica 3 con 80 usuarios concurrentes

	160 usuarios		
	Sesión	Transacción	Total
Media (ms)	561.20215	1840.42816	2353.35128
IC^{95} Media (ms)	(500.8863; 621.5179)	(1491.3830; 2189.4733)	(2038.6057; 2668.0967)
Varianza(ms)	589.52422	19742.5100	16053.07363

Tabla 10 Tiempo de respuesta para las transacciones de la práctica 3 con 160 usuarios concurrentes

	40 usuarios	80 usuario	160 usuarios
Media (peticiones/s)	2.98062	3.62379	7.01758
IC^{95} Media (peticiones/s)	(2.6311; 3.3301)	(3.5160; 3.7315)	(6.5199; 7.5151)
Varianza (peticiones/s)	0.01979	0.00188	0.04012

Tabla 11 Productividad para las transacciones de la práctica 3