

TESTS DE RENDIMIENTO

Grupo 21



Contenido

[1. Descripción del ordenador 2](#_Toc8928947)

[1.1. Descripción de máquina virtual 2](#_Toc8928948)

[1.2. Descripción de máquina física 3](#_Toc8928949)

[2. Casos de uso e informes 3](#_Toc8928950)

[2.1.1. Casos de uso 3](#_Toc8928951)

[2.1.2. Pruebas realizadas 3](#_Toc8928952)

[2.1.3. Conclusiones 6](#_Toc8928953)

[2.1.4. Casos de uso 6](#_Toc8928954)

[2.1.5. Pruebas realizadas 6](#_Toc8928955)

[2.1.6. Conclusiones 9](#_Toc8928956)

[2.1.7. Casos de uso 9](#_Toc8928957)

[2.1.8. Pruebas realizadas 9](#_Toc8928958)

[2.1.9. Conclusiones 12](#_Toc8928959)

[2.1.10. Casos de uso 12](#_Toc8928960)

[2.1.11. Pruebas realizadas 12](#_Toc8928961)

[2.1.12. Conclusiones 15](#_Toc8928962)

[2.1.13. Casos de uso 15](#_Toc8928963)

[2.1.14. Pruebas realizadas 15](#_Toc8928964)

[2.1.15. Conclusiones 17](#_Toc8928965)

[2.1.16. Casos de uso 18](#_Toc8928966)

[2.1.17. Pruebas realizadas 18](#_Toc8928967)

[2.1.18. Conclusiones 20](#_Toc8928968)

[2.1.19. Casos de uso 20](#_Toc8928969)

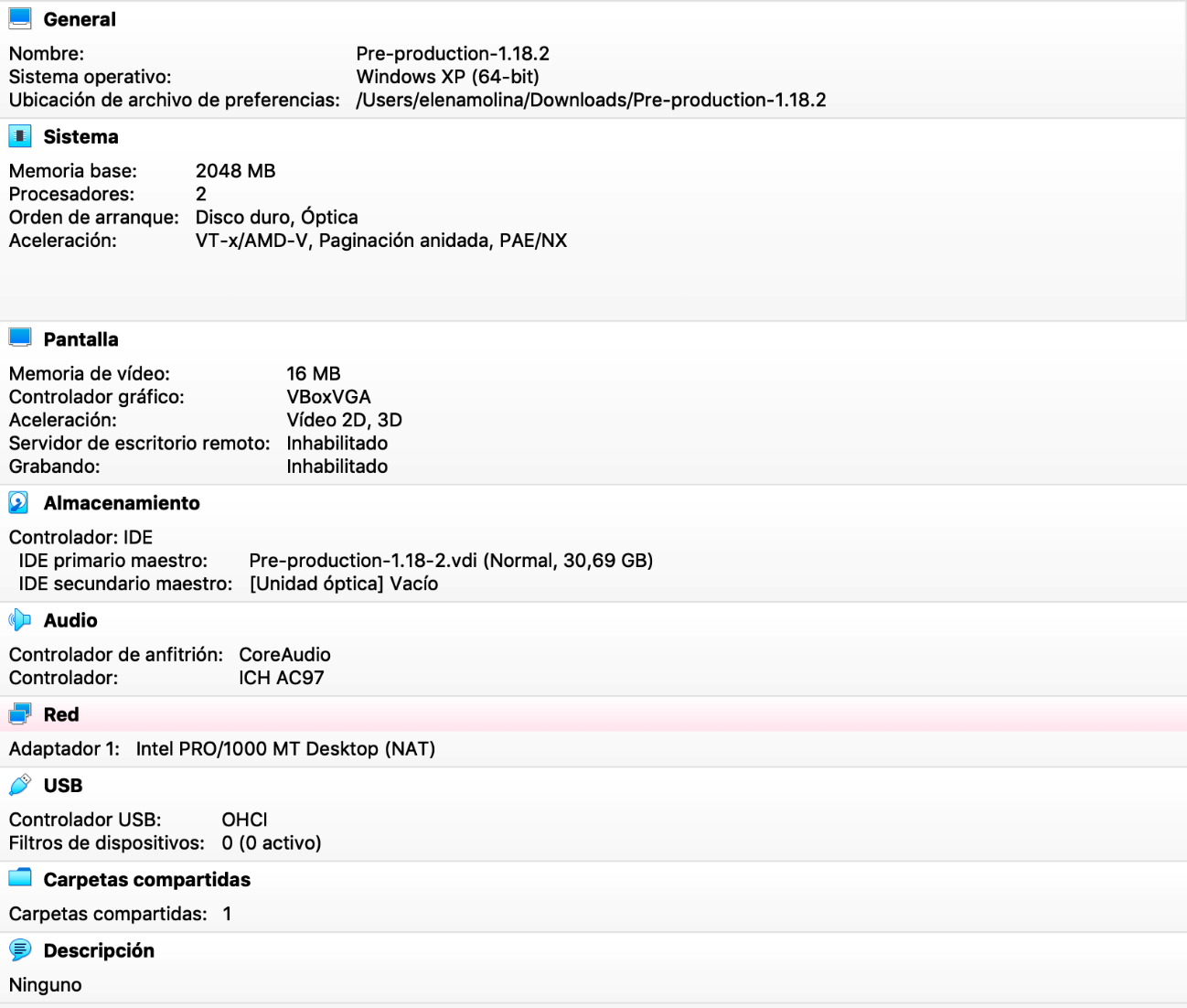
[2.1.20. Pruebas realizadas 21](#_Toc8928970)

[2.1.21. Conclusiones 23](#_Toc8928971)

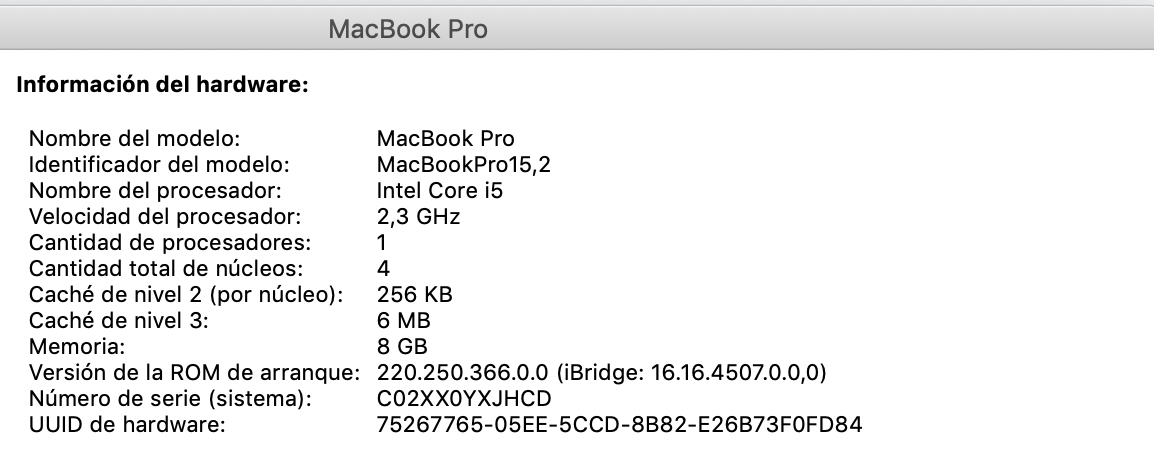
# Descripción del ordenador

Máquina virtual ejecutada con Oracle VM VirtualBox.

## Descripción de máquina virtual



## Descripción de máquina física



Cabe destacar que se realizan todas las peticiones con el protocolo HTTPS.

# Casos de uso e informes

2.1 An actor who is not authenticated must be able to register to the system as a manager (RF 11.1) (HECHO Y FUNCIONA Y NUMEROS BAJOS)

El archivo correspondiente a este test es *AsignPositionAuditorCasePlan.jmx.*

### Casos de uso

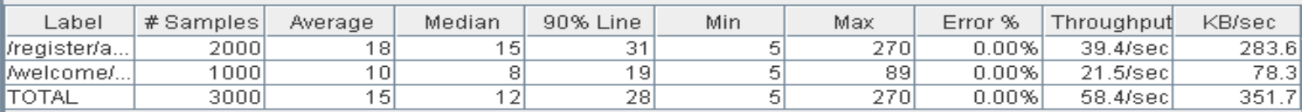
En este caso tenemos un caso de uso con los siguientes pasos:

1. Acceder al menú Register y pulsar en el botón “Register a Manager”

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones



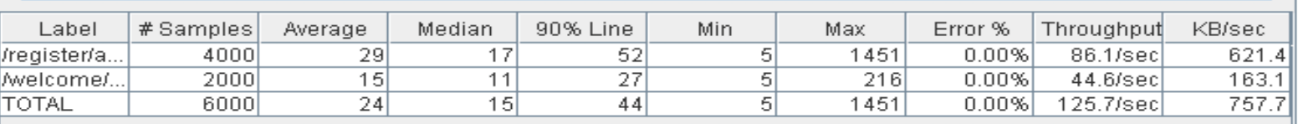
No hay errores HTTP y el tiempo medio de espera total es de 50ms = 0,05s.

El mayor tiempo de espera se da al realizarse la operación de logueo siendo de 31ms = 0,03s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 79ms = 0,07s.

El mayor tiempo de espera se da al realizarse la operación de logueo siendo de 52ms = 0,052s.

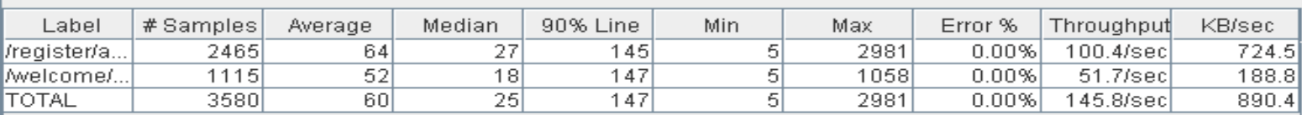
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 292ms = 0,29.

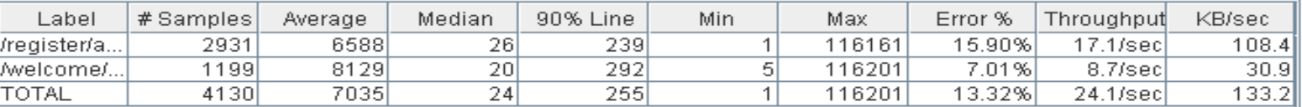
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 147 = 0.14s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 531ms = 0,531s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 292ms = 0,29.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.2. An actor who is not authenticated must be able to register to the system as a client (RF 11.1) (HECHO Y FUNCIONA Y NUMEROS BAJOS)

El archivo correspondiente a este test es *AuditAuditorCasePlan.jmx.*

### Casos de uso

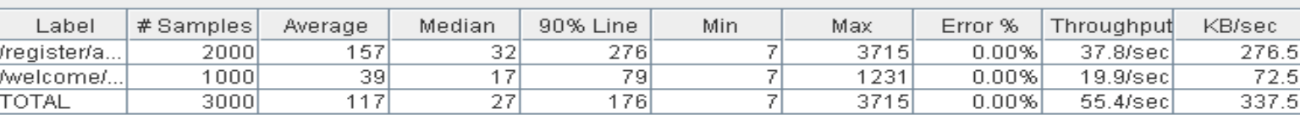
En este caso tenemos un caso de uso con los siguientes pasos:

1. Acceder al menú Register y pulsar el botón “Register a Client”

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones



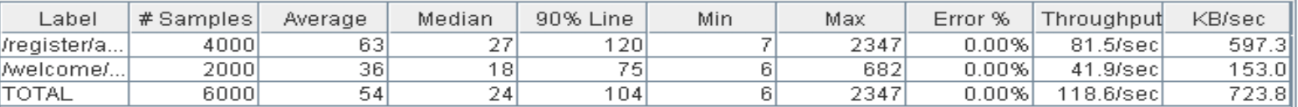
No hay errores HTTP y el tiempo medio de espera total es de 355ms = 0,355s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 276ms = 0,027s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 195 = 0,19s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 120 = 0,12s.

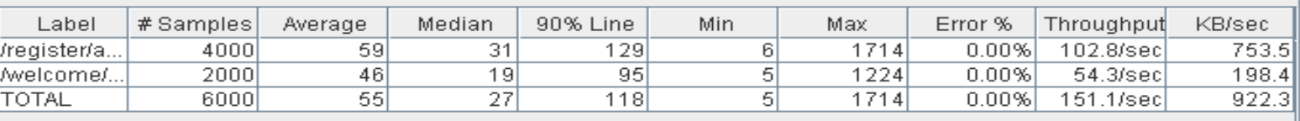
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 224 = 0,22.

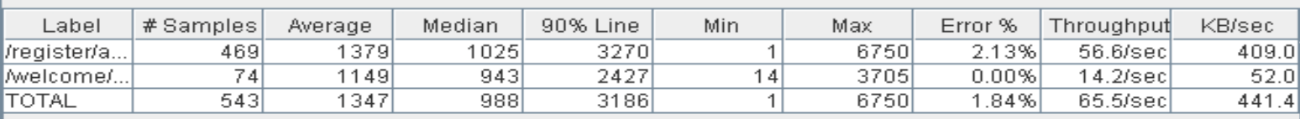
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 129ms = 0,12s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones



En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“*Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated*”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el “*Constant Delay Offset*” de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

2.1 A actor who is not authenticated must be able to list all the accepted clubs that are in the system, see their information, and navigate to his manager or to the list of their events (RF 11.2) (NO HECHO)

El archivo correspondiente a este test es *AsignPositionAuditorCasePlan.jmx.*

### Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Acceder al menú club/managers/events y dar click a List clubs

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 231ms = 0,23s.

El mayor tiempo de espera se da al realizarse la operación de logueo siendo de 0,78ms = 0,07s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 623ms = 0,623s.

El mayor tiempo de espera se da al realizarse la operación asignación siendo de 205ms = 0,2s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 1693ms = 1,693s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 550 = 0.55s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 5578ms = 5,5s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1534ms = 1,5.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

2.1 An actor who is not authenticated must be able to list all managers that are in the system, see their information, and navigate to their clubs and their social profiles (RF 11.3) .(NO HECHO)

El archivo correspondiente a este test es *AsignPositionAuditorCasePlan.jmx.*

### Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Acceder al menú club/managers/events y dar click a List manager

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 231ms = 0,23s.

El mayor tiempo de espera se da al realizarse la operación de logueo siendo de 0,78ms = 0,07s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 623ms = 0,623s.

El mayor tiempo de espera se da al realizarse la operación asignación siendo de 205ms = 0,2s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 1693ms = 1,693s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 550 = 0.55s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 5578ms = 5,5s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1534ms = 1,5.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

2.1 An actor who is not authenticated must be able to list all events that are in the system saved in final mode and navigate to theis respective club. In addition to showing the opinions of each event (RF 11.4). (NO HECHO)

El archivo correspondiente a este test es *AsignPositionAuditorCasePlan.jmx.*

### Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Acceder al menú club/managers/events y dar click a List events

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 231ms = 0,23s.

El mayor tiempo de espera se da al realizarse la operación de logueo siendo de 0,78ms = 0,07s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 623ms = 0,623s.

El mayor tiempo de espera se da al realizarse la operación asignación siendo de 205ms = 0,2s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 1693ms = 1,693s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 550 = 0.55s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 5578ms = 5,5s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1534ms = 1,5.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

4.1. An actor who is authenticated must be able to edit his or her personal data. (RF 12.2)

(HECHO Y FUNCIONA Y NUMEROS BAJOS)

El archivo correspondiente a este test es *AdministratorProcedureCasePlan.jmx.*

### Casos de uso

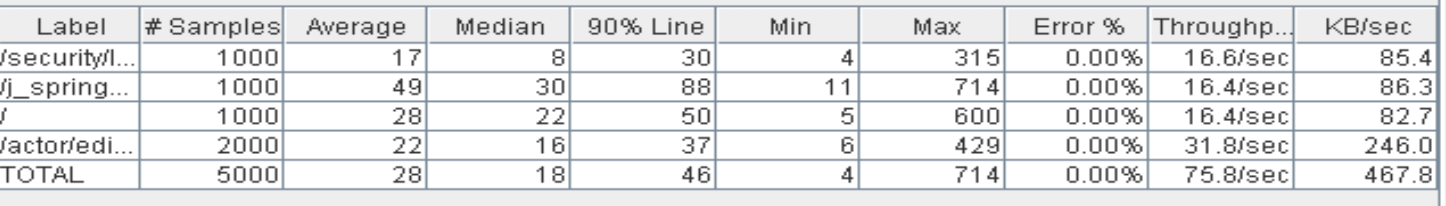
En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse
2. Acceder al menú de el perfil y pulsar “Edit Personal Data”

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones



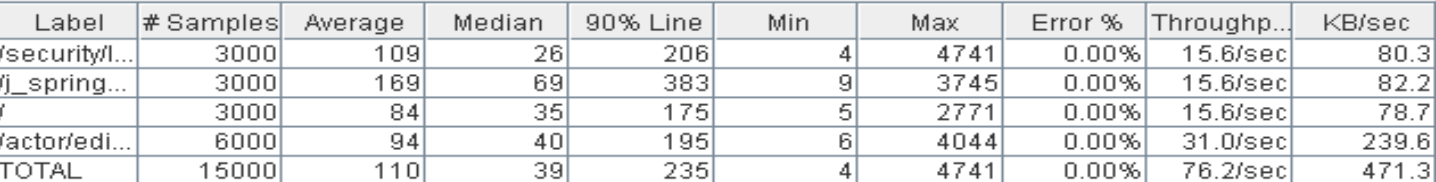
No hay errores HTTP y el tiempo medio de espera total es de 205ms = 0,20.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 88 ms = 0,08 segs

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 959ms = 0,95 s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 383 ms = 0,38 segs.

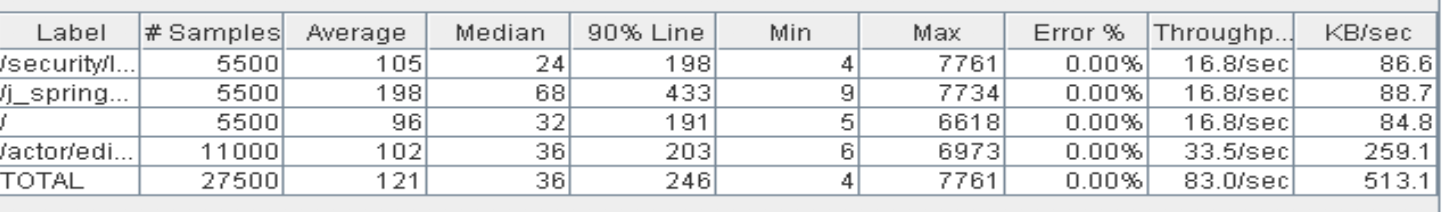
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 1025ms = 1,025.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de

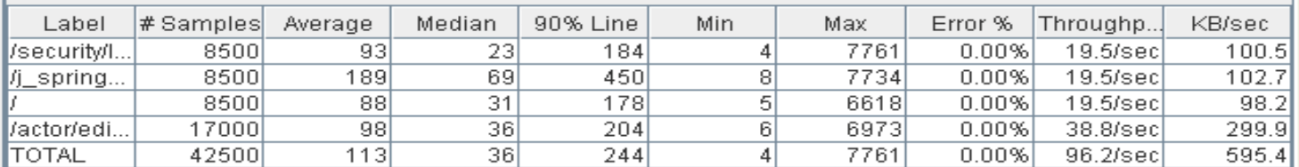
433 ms = 0,43 segs

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 1016ms = 1,016s.

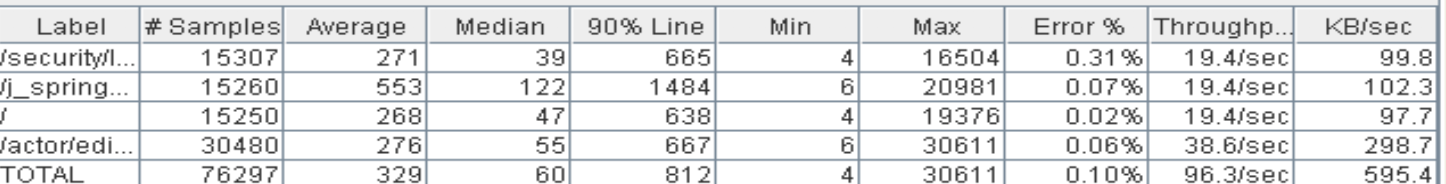
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 450ms = 0,4.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 5:**

* 400 usuarios
* Loop de 10 iteraciones



En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“*Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated*”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el “*Constant Delay Offset*” de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated must be able to manage his or her social profiles, who include creating ,updating and deleting them (RF 12.3) (HECHO Y FUNCIONA Y NUMEROS BAJOS)

El archivo correspondiente a este test es *RegisterAuditorCasePlan.jmx.*

### Casos de uso

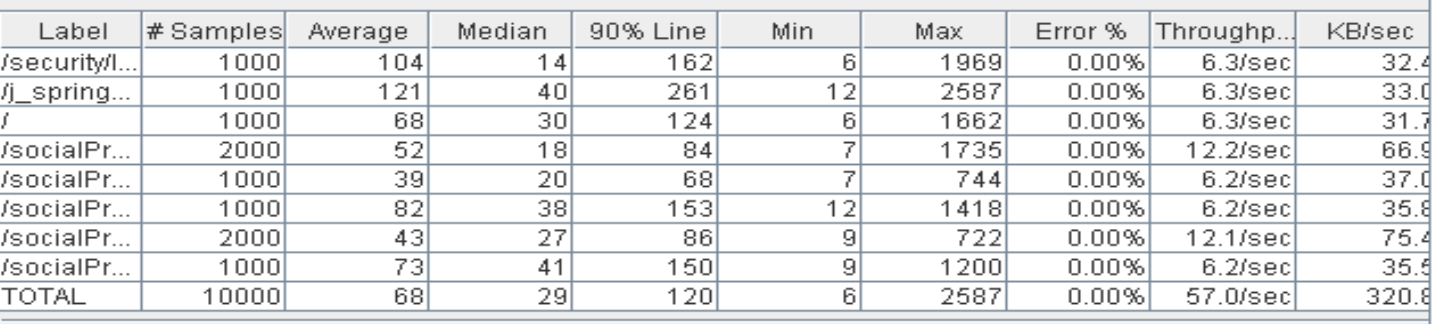
En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse
2. Acceder al menú del perfil y pulsar “List Social Profiles”
3. Pulsar el botón de créate, rellnar los campos y pulsar sabe
4. Pulsar el botón edit y save al escribir un cambio
5. Pulsar el botón display del social profile creado
6. Pulsar el botón de eliminar en la vista edit

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones



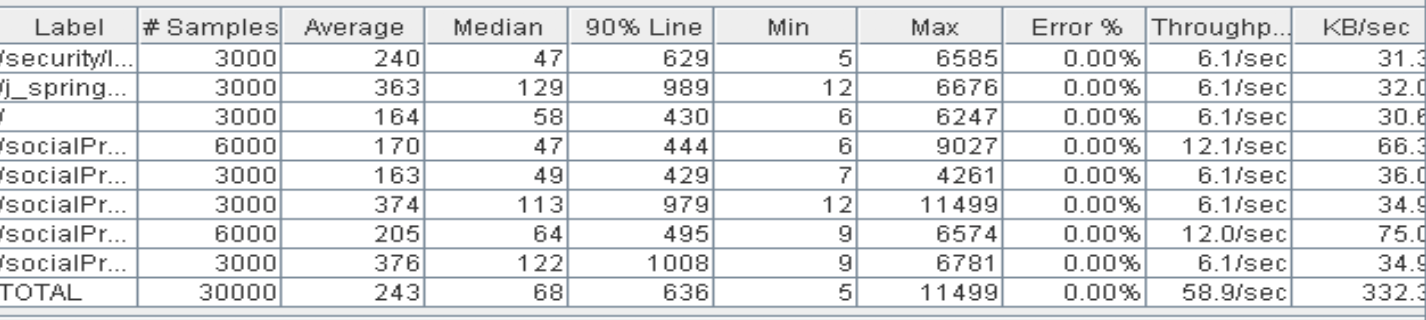
No hay errores HTTP y el tiempo medio de espera total es de 1088ms = 1,08s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 261ms = 0,261 s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 5403ms = 5,4s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 989ms = 0,9s.

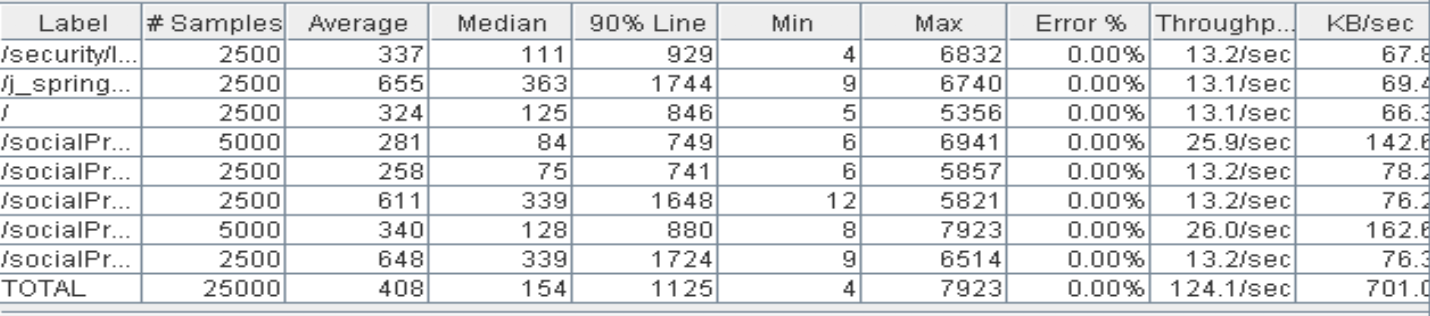
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 9261ms = 9,2s.

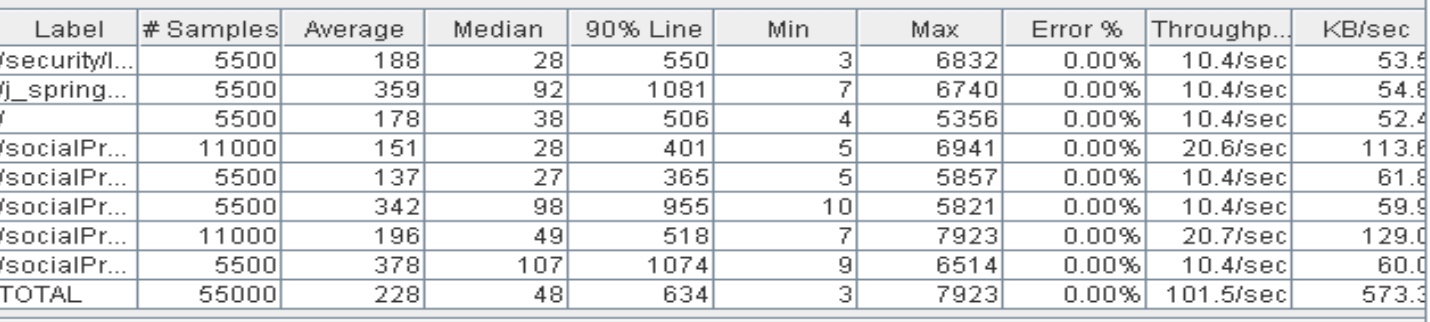
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1744ms = 1,7s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 10450ms = 10,4s.

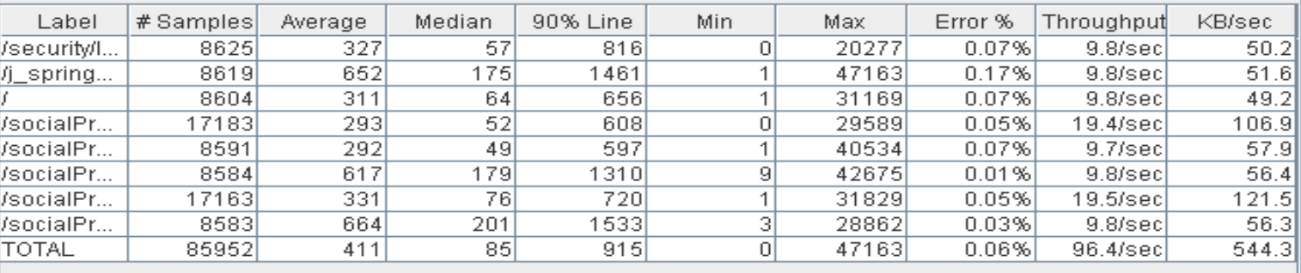
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1081ms = 1,081s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 5:**

* 400 usuarios
* Loop de 10 iteraciones



En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“*Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated*”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el “*Constant Delay Offset*” de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated must be able to exchange messages with other actors and manage them. Manage his or her messages boxes, except for the system boxes. . An actor who is authenticated as an administrator must be able to broadcast a message to all of the actors of the system (RF 12.4, RF 12.5 and RF 15.4) (HECHO Y FUNCIONA Y NUMEROS BAJOS)

El archivo correspondiente a este test es *AdministratorProcessCasePlan.jmx.*

### Casos de uso

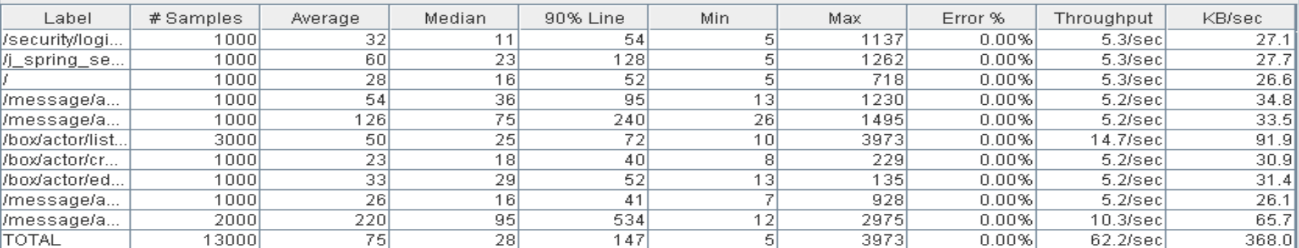
En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse
2. Acceder al menú de “My boxes” y pulsar “Exchange message”
3. Rellenar formulario y pulsar botón save
4. Pulsar “Create box”
5. Rellenar el formulario y pulsar el botón save
6. Acceder al menú de “My Boxes” y pulsar “Exchange broadcast message”
7. Rellenar el formulario y pulsar el botón save.

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones



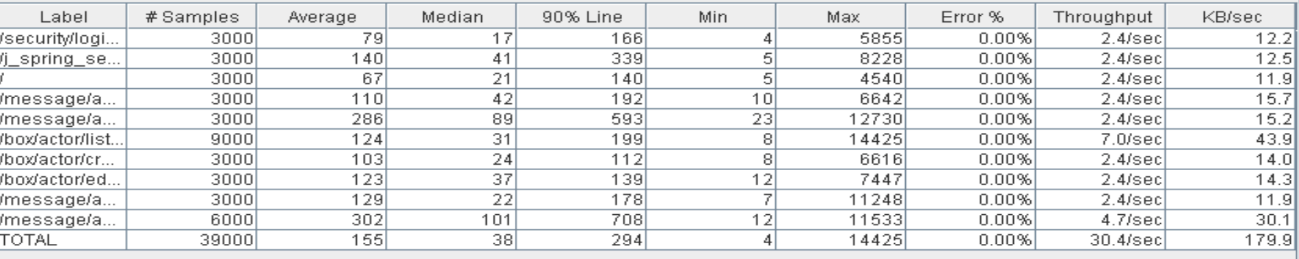
No hay errores HTTP y el tiempo medio de espera total es de 1308ms = 1,3s.

El mayor tiempo de espera se da al realizarse la operación de mensaje broadcast siendo de 534ms = 0,5s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 2766ms = 2,766s.

El mayor tiempo de espera se da al realizarse la operación de mensaje bvroadcast siendo de 708ms = 0,7.

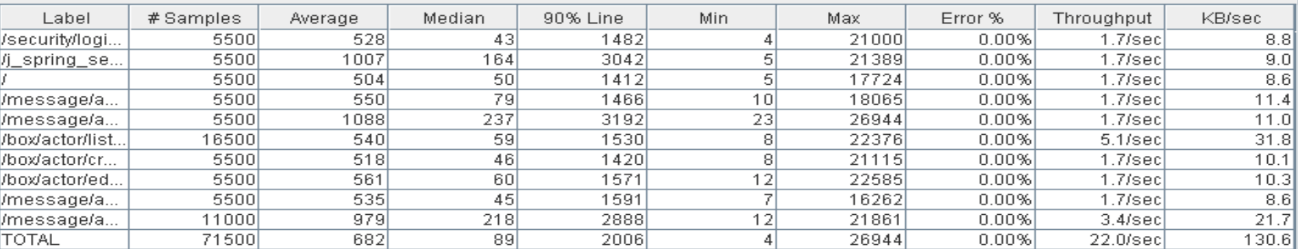
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 33412ms = 33,4.

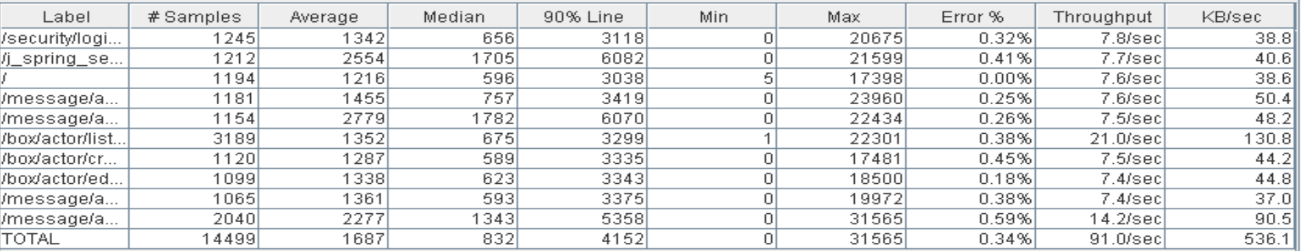
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 3042ms = 3,042s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones



En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“*Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated*”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el “*Constant Delay Offset*” de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

### **MANAGER**

3.1. An actor who is authenticated as manager must be able to manage their clubs, which includes listing, showing, creating, updating, and deleting them.

A club in final mode cannot be update or deleted. When a club is displayed, the manager can see the score of this club(RF 13.1) (HECHO,FUNCIONA, NUMEROS ALTOS)

El archivo correspondiente a este test es *NoRegisterProviderCasePlan.jmx.*

### Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse como manager
2. Pulsar en el menú “My clubs” , “create club”
3. Rellenar formulario y pulsar botón save
4. Pulsar el botón Edit del club creado
5. Rellenar formulario y pulsar botón save
6. Pulsar el botón Edit display y volver
7. Pulsar el botón Edit del club
8. Pulsar botón Delete

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 46ms = 0,046s.

El mayor tiempo de espera se da al realizarse la operación de registro siendo de 34ms = 0,34s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 97ms = 0,09s.

El mayor tiempo de espera se da al realizarse la operación de registro siendo de 72ms = 2,207s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 295ms = 0,295s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 179ms = 0,179segs.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“*Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated*”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el “*Constant Delay Offset*” de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as manager must be able to manage the events of theis clubs, which includes listing, creating,updating and deleting them. A manager may update or delete an event only if it’s saved in draft mode. Whean an event is saved in final mode, a notification must be sent at all the cleints who follows the club where the event is organized. (RF 13.2) (HECHO,FUNCIONA, NUMEROS ALTOS)

El archivo correspondiente a este test es *itemProviderCasePlan.jmx.*

### Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse como manager
2. Pulsar en el menú “My events” , “create event”
3. Rellenar formulario y pulsar botón save
4. Pulsar el botón Edit del event creado
5. Rellenar formulario y pulsar botón save
6. Pulsar el botón Edit display y volver
7. Pulsar el botón Edit del club
8. Pulsar botón Delete

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 236ms = 0,236s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 43ms = 0,043s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 881ms = 0,81s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 209ms = 0,209s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 1051ms = 1,051s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 239ms = 0,239s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 7026ms = 7,026s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1263ms = 1,263s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as manager must be able to cancel an event. When an event is cancelled a notification must be send at all the clients who participle in this events. Only events saved in final mode can be cancelled. (RF 13.3) (NO HECHO)

El archivo correspondiente a este test es *itemProviderCasePlan.jmx.*

### Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse como manager
2. Listar mis clubs
3. Pulsar botón “cancel”

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 236ms = 0,236s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 43ms = 0,043s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 881ms = 0,81s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 209ms = 0,209s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 1051ms = 1,051s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 239ms = 0,239s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 7026ms = 7,026s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1263ms = 1,263s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. .An actor who is authenticated as manager must be able to list the opinions of his events and the average of his scores for each event and navigate to the client who participle in the event and show his personal data and social profiles. (RF 13.4) (NO HECHO)

El archivo correspondiente a este test es *itemProviderCasePlan.jmx.*

### Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse como manager
2. Listar my opinions
3. Pulsar Show Client

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 236ms = 0,236s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 43ms = 0,043s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 881ms = 0,81s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 209ms = 0,209s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 1051ms = 1,051s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 239ms = 0,239s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 7026ms = 7,026s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1263ms = 1,263s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as manager must be able to list the participations of his events and the total number of participations for each event and navigate to the client who participle in the event and show his personal data and social profiles (RF 13.5) (NO HECHO)

El archivo correspondiente a este test es *itemProviderCasePlan.jmx.*

### Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse como manager
2. Listar my participations
3. Pulsar el boton Show event

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 236ms = 0,236s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 43ms = 0,043s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 881ms = 0,81s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 209ms = 0,209s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 1051ms = 1,051s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 239ms = 0,239s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 7026ms = 7,026s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1263ms = 1,263s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 5:**

* 400 usuarios
* Loop de 10 iteraciones

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“*Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated*”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el “*Constant Delay Offset*” de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

### **CLIENT**

3.1. An actor who is authenticated as a client must be able to follow or unfollow a club (RF 14.1) (NO HECHO)

El archivo correspondiente a este test es *itemProviderCasePlan.jmx.*

### Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse como client
2. Listar my clubs
3. Pulsar botón follow/unfollow

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 236ms = 0,236s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 43ms = 0,043s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 881ms = 0,81s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 209ms = 0,209s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 1051ms = 1,051s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 239ms = 0,239s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 7026ms = 7,026s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1263ms = 1,263s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as a client must be able to list the clubs he follows and navigate to the events that are organized in that clubs ordered by date and divided in three lists (AVALAIBLE, CANCELLED and FINISHED)(RF 14.2) (NO HECHO)

El archivo correspondiente a este test es *itemProviderCasePlan.jmx.*

### Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse como client
2. Acceder al menú “My follow clubs”

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 236ms = 0,236s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 43ms = 0,043s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 881ms = 0,81s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 209ms = 0,209s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 1051ms = 1,051s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 239ms = 0,239s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 7026ms = 7,026s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1263ms = 1,263s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as a client must be able to filter the catalogue of events using the following filters: a single key word that must appear somewhere in its club, ticker, description, a category to which the event must belong, a range of prices, or a range of dates (RF 14.3). (NO HECHO)

El archivo correspondiente a este test es *itemProviderCasePlan.jmx.*

### Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse como client
2. Listar mis events
3. Rellenar filtro y pulsar botón

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 236ms = 0,236s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 43ms = 0,043s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 881ms = 0,81s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 209ms = 0,209s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.

**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 1051ms = 1,051s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 239ms = 0,239s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 7026ms = 7,026s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1263ms = 1,263s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 5:**

* 400 usuarios
* Loop de 10 iteraciones

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“*Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated*”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el “*Constant Delay Offset*” de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as a client must be able to participate in an event of one of the clubs he follows. The client must have a valid credit card linked. Each client can only have one participation by event. Clients can not participate in events which have already finished (RF 14.4) .(NO HECHO)

El archivo correspondiente a este test es *itemProviderCasePlan.jmx.*

### Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse como client
2. Listar mis clubs

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 236ms = 0,236s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 43ms = 0,043s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 881ms = 0,81s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 209ms = 0,209s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 1051ms = 1,051s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 239ms = 0,239s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 7026ms = 7,026s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1263ms = 1,263s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 5:**

* 400 usuarios
* Loop de 10 iteraciones

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“*Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated*”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el “*Constant Delay Offset*” de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as a client must be able to list the events he participate( RF 14.5) (NO HECHO)

El archivo correspondiente a este test es *itemProviderCasePlan.jmx.*

### Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse como client
2. Acceder a My Participations en el menu

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 236ms = 0,236s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 43ms = 0,043s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 881ms = 0,81s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 209ms = 0,209s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 1051ms = 1,051s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 239ms = 0,239s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 7026ms = 7,026s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1263ms = 1,263s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 5:**

* 400 usuarios
* Loop de 10 iteraciones

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“*Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated*”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el “*Constant Delay Offset*” de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as a client must be able to publish an opinion about an event he participated who is already finished (RF 14.6) (NO HECHO)

El archivo correspondiente a este test es *itemProviderCasePlan.jmx.*

### Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse como client
2. Listar my opinions
3. Pulsar botón de “Create opinión”
4. Rellenar formulario y pulsar save

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 236ms = 0,236s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 43ms = 0,043s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 881ms = 0,81s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 209ms = 0,209s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 1051ms = 1,051s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 239ms = 0,239s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 7026ms = 7,026s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1263ms = 1,263s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 5:**

* 400 usuarios
* Loop de 10 iteraciones

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“*Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated*”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el “*Constant Delay Offset*” de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as a client must be able to manage his credit card which includes creating, editing and deleting it. (RF 14.7) (NO HECHO)

El archivo correspondiente a este test es *itemProviderCasePlan.jmx.*

### Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse como client
2. En el menú del perfil, pulsar “Edit personal data”
3. Cambiar tarjeta de crédito y pulsar save

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 236ms = 0,236s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 43ms = 0,043s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 881ms = 0,81s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 209ms = 0,209s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 1051ms = 1,051s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 239ms = 0,239s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 7026ms = 7,026s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1263ms = 1,263s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 5:**

* 400 usuarios
* Loop de 10 iteraciones

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“*Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated*”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el “*Constant Delay Offset*” de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

### **ADMINISTRATOR**

3.1. An actor who is authenticated as an administrator must be able to create new user accounts for new administrators(RF 15.1) (HECHO Y FUNCIONA)

El archivo correspondiente a este test es *RegisterAdminTestPlan.jmx.*

### Casos de uso

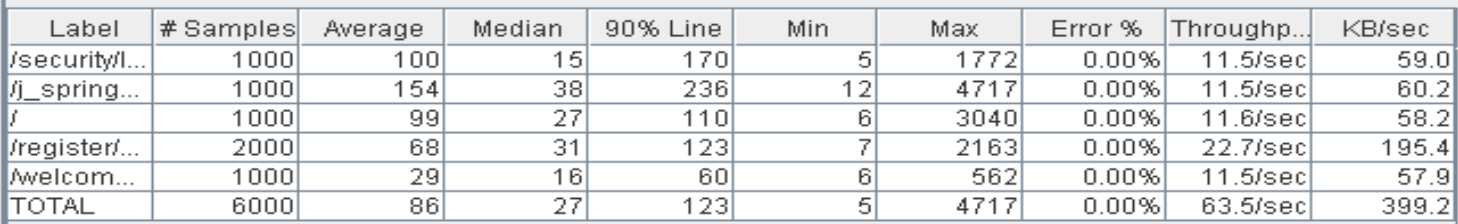
En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse como admin
2. Acceder al menú Register
3. Pulsar botón “Register a administrator”
4. Rellenar formulario y pulsar save

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones



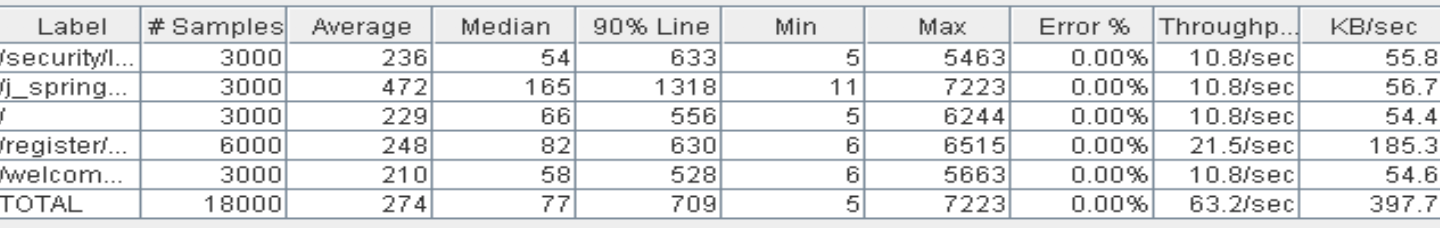
No hay errores HTTP y el tiempo medio de espera total es de 699ms = 0,699s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 236ms = 0,236s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 3665ms = 3,6s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1318ms = 1,3s.

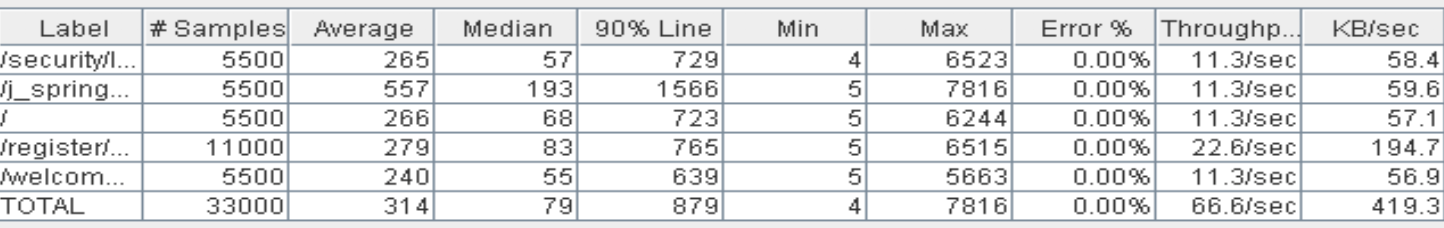
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 3884ms = 3,88s.

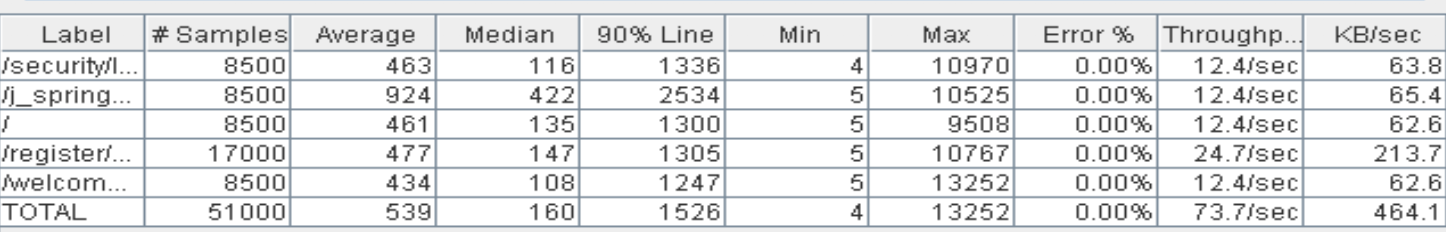
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1566ms = 1,5s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 7722ms = 7,7s.

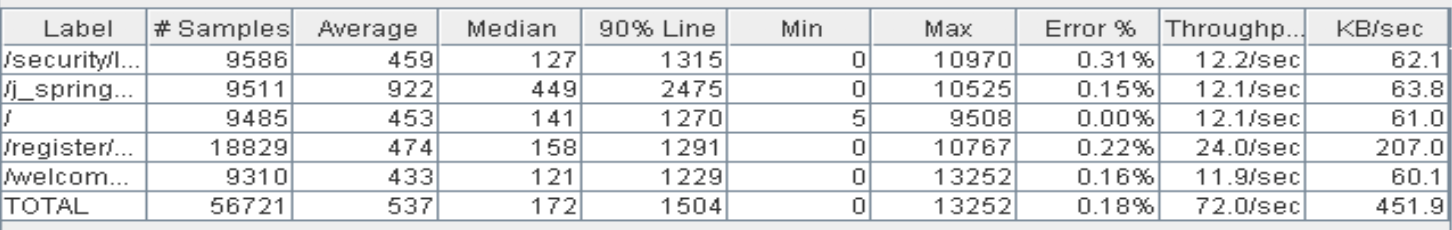
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 2534ms = 2,5s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 5:**

* 350 usuarios
* Loop de 10 iteraciones



En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“*Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated*”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el “*Constant Delay Offset*” de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as an administrator must be able to manage the catalogue of categories, which includes listing, showing, creating, updating and deleting them. Note that categories evolve independently from events, which means that they are referenced from an event or not. (RF 15.2) (HECHO, NO FUNCIONA)

El archivo correspondiente a este test es *itemProviderCasePlan.jmx.*

### Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse como admin
2. Acceder a categories a través del menú
3. Pulsar el botón “Create category”
4. Rellenar el formulario y pulsar save
5. Pulsar el botón edit de la categoría creada
6. Rellenar el formulario y pulsar el botón save
7. Pulsar el botón de display y volver
8. Pulsar el botón de edit y de delete

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 236ms = 0,236s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 43ms = 0,043s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 881ms = 0,81s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 209ms = 0,209s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 1051ms = 1,051s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 239ms = 0,239s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 7026ms = 7,026s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1263ms = 1,263s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 5:**

* 400 usuarios
* Loop de 10 iteraciones

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“*Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated*”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el “*Constant Delay Offset*” de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as an administrator must be able to list the clubs that are not accepted yet and accept or refuse them. If the club is refused, a reason must be given by the administrator (RF 15.3) (HECHO, NUMEROS ALTOS)

El archivo correspondiente a este test es *itemProviderCasePlan.jmx.*

### Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse como admin
2. Listar mis clubs en el menu

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 236ms = 0,236s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 43ms = 0,043s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 881ms = 0,81s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 209ms = 0,209s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 1051ms = 1,051s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 239ms = 0,239s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones

No hay errores HTTP y el tiempo medio de espera total es de 7026ms = 7,026s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1263ms = 1,263s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 5:**

* 400 usuarios
* Loop de 10 iteraciones

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“*Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated*”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el “*Constant Delay Offset*” de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as an administrator must be able to display a dashboard (RF 15.5) (HECHO Y FUNCIONA)

El archivo correspondiente a este test es *itemProviderCasePlan.jmx.*

### Casos de uso

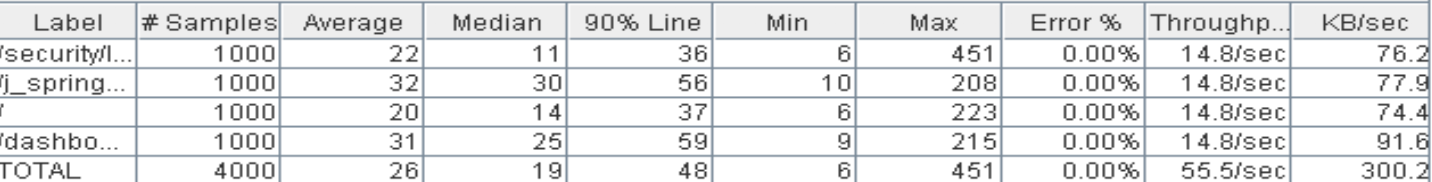
En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse como admin
2. Acceder a la dashboar a través del menu

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones



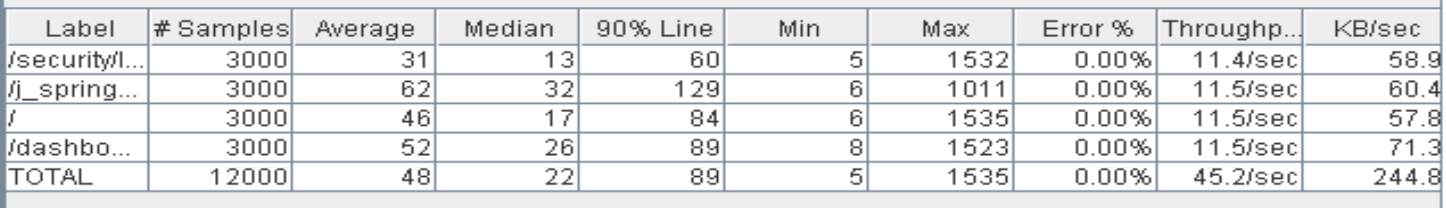
No hay errores HTTP y el tiempo medio de espera total es de 158ms = 0,158s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 30ms = 0,03s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 336ms = 0,36s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 109ms = 0,109s.

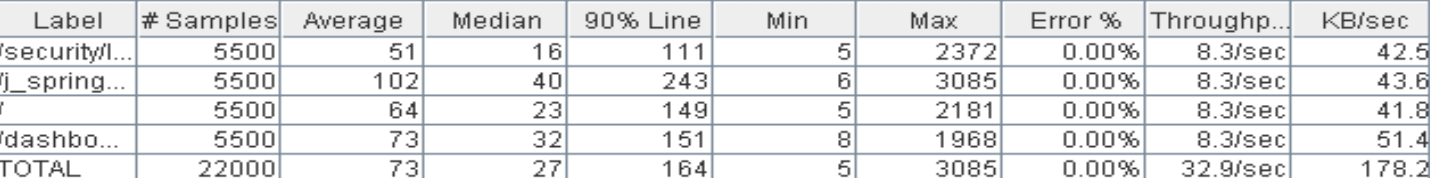
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 752ms = 0,76.

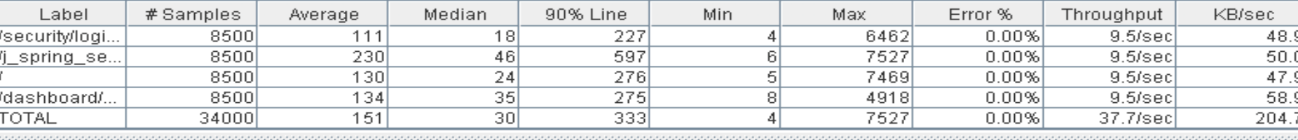
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 243ms = 0,243s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 982ms = 9,8.

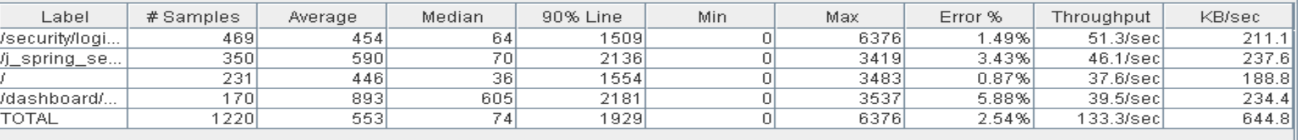
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 592ms = 0,59s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 5:**

* 400 usuarios
* Loop de 10 iteraciones



En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“*Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated*”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el “*Constant Delay Offset*” de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. The system must be easy to customize at run time (RNF 17) (HECHO Y FUNCIONA)

El archivo correspondiente a este test es *itemProviderCasePlan.jmx.*

### Casos de uso

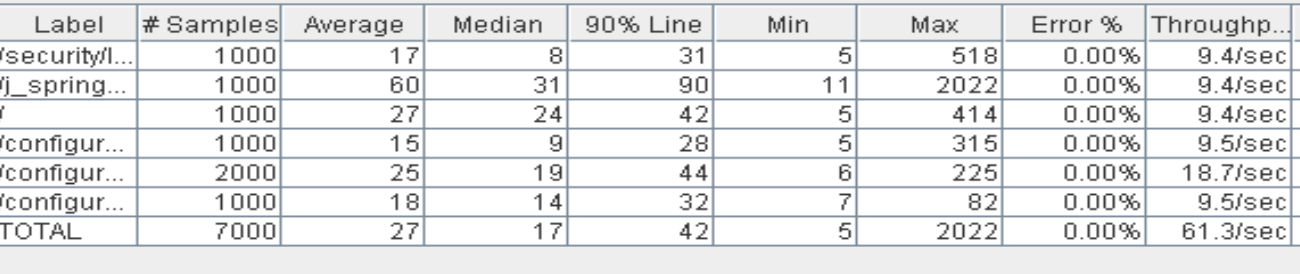
En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse como admin
2. Acceder al menú del perfil y pulsar Configuration
3. Pulsar edit
4. Rellenar el formulario y pulsar save

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones



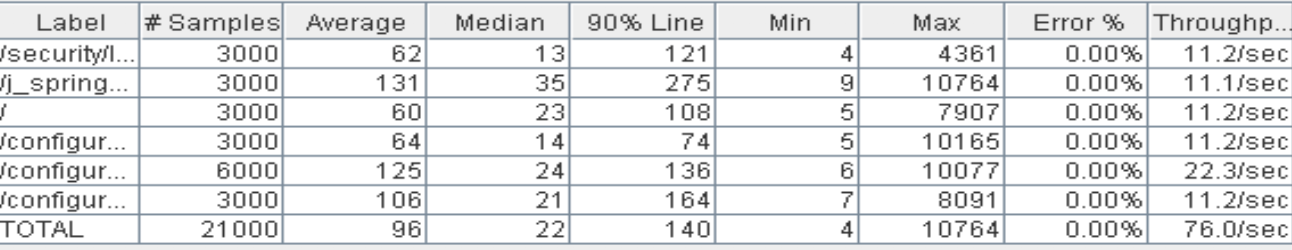
No hay errores HTTP y el tiempo medio de espera total es de 56ms = 0,56s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 90ms = 0,09s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 881ms = 0,81s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 275ms = 0,275s.

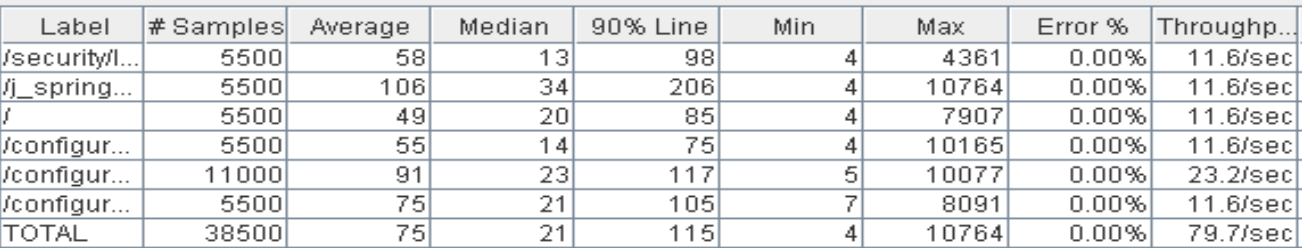
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



**Prueba 3:**

* 250 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 902ms = 0,092s.

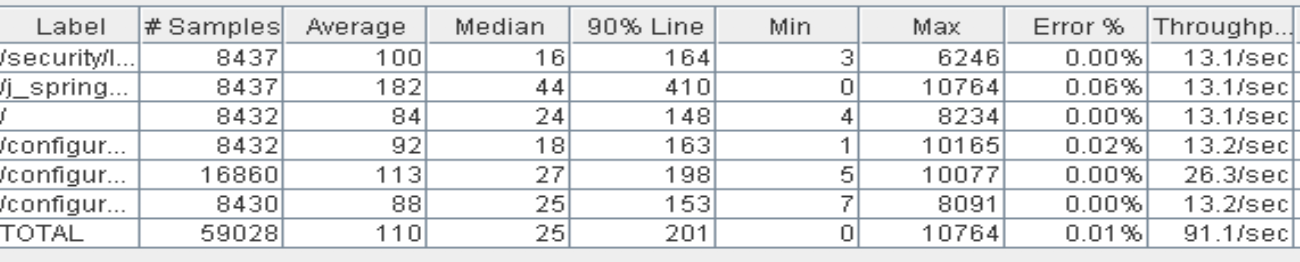
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 206ms = 0,206s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

**Prueba 4:**

* 300 usuarios
* Loop de 10 iteraciones



En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“*Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated*”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el “*Constant Delay Offset*” de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

### Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.