

TESTS DE RENDIMIENTO

Grupo 21



Contenido

[1. Descripción del ordenador 2](#_Toc8774968)

[1.1. Descripción de máquina virtual 2](#_Toc8774969)

[1.2. Descripción de máquina física 3](#_Toc8774970)

[2. Casos de uso e informes 3](#_Toc8774971)

[2.1.1. Casos de uso 3](#_Toc8774972)

[2.1.2. Pruebas realizadas 3](#_Toc8774973)

[2.1.3. Conclusiones 5](#_Toc8774974)

[2.1.4. Casos de uso 6](#_Toc8774975)

[2.1.5. Pruebas realizadas 6](#_Toc8774976)

[2.1.6. Conclusiones 8](#_Toc8774977)

[2.1.7. Casos de uso 8](#_Toc8774978)

[2.1.8. Pruebas realizadas 8](#_Toc8774979)

[2.1.9. Conclusiones 10](#_Toc8774980)

[2.1.10. Casos de uso 11](#_Toc8774981)

[2.1.11. Pruebas realizadas 11](#_Toc8774982)

[2.1.12. Conclusiones 13](#_Toc8774983)

[2.1.13. Casos de uso 13](#_Toc8774984)

[2.1.14. Pruebas realizadas 13](#_Toc8774985)

[2.1.15. Conclusiones 15](#_Toc8774986)

[2.1.16. Casos de uso 16](#_Toc8774987)

[2.1.17. Pruebas realizadas 16](#_Toc8774988)

[2.1.18. Conclusiones 18](#_Toc8774989)

[2.1.19. Casos de uso 18](#_Toc8774990)

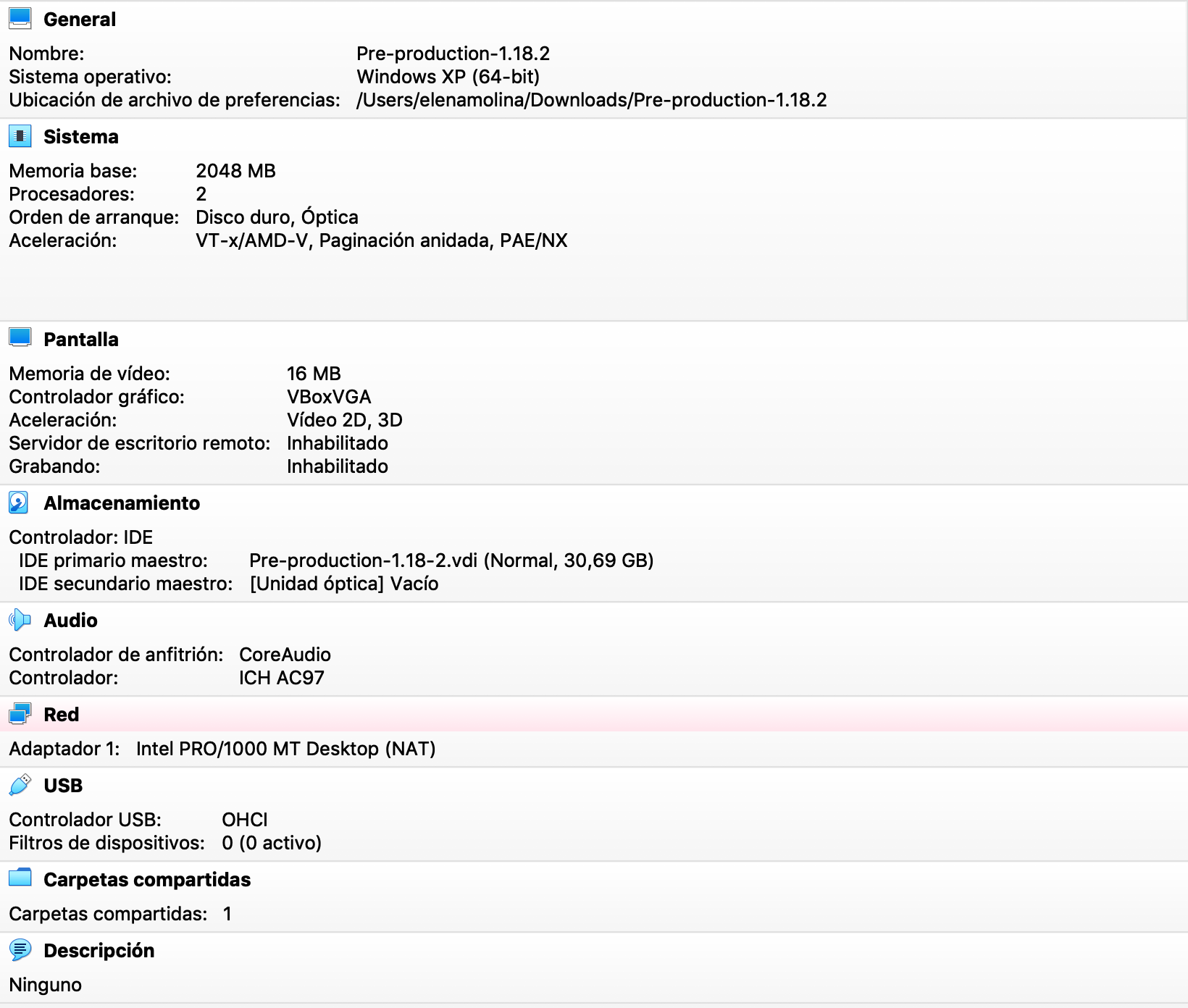
[2.1.20. Pruebas realizadas 18](#_Toc8774991)

[2.1.21. Conclusiones 20](#_Toc8774992)

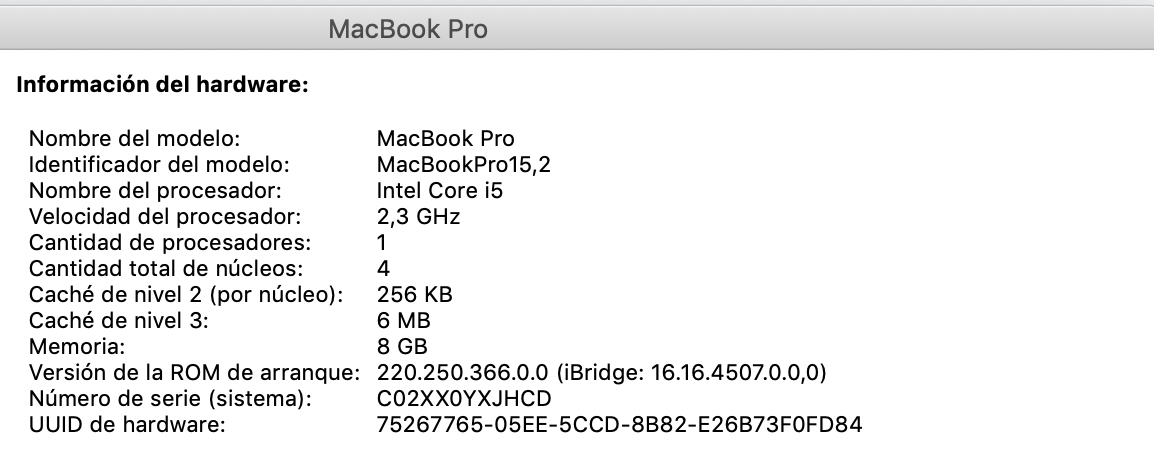
# Descripción del ordenador

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 authenticated as an auditor must be able to self-assign a position to audit it.(RF 3.1)

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

### Casos de uso

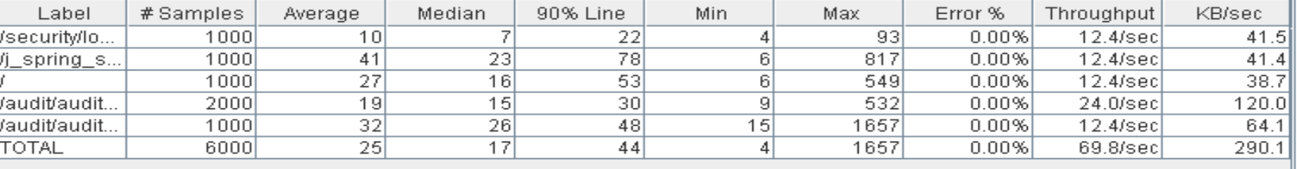
En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse
2. Listar mis Posiciones
3. En el listado de posiciones libres, pulsar en el botón “Asignar”

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones



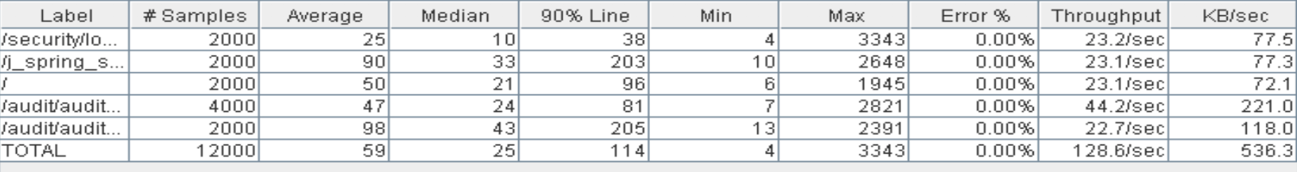
No hay errores HTTP y el tiempo medio de espera total es de 231ms = 0,23s.

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

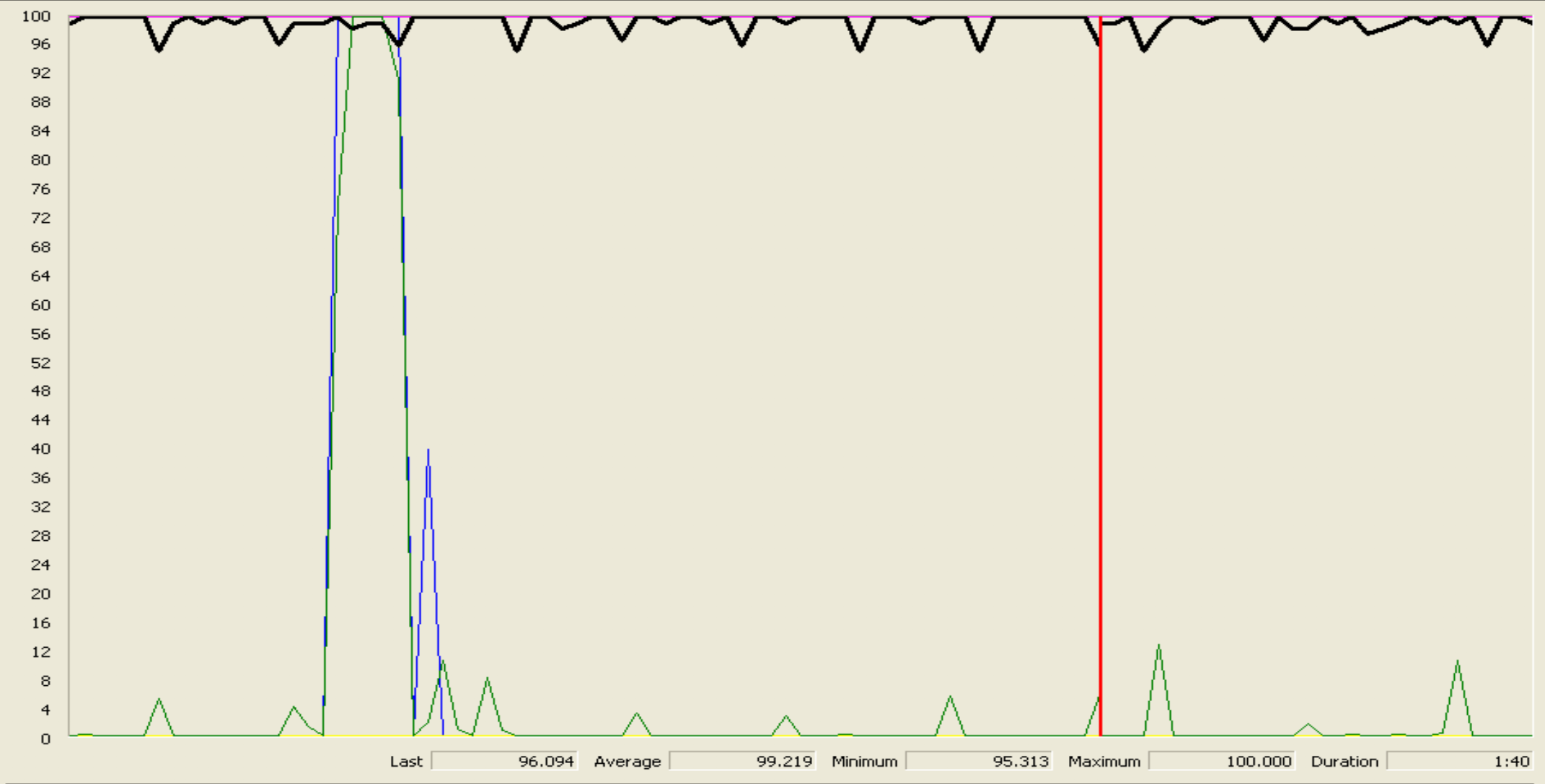


No hay errores HTTP y el tiempo medio de espera total es de 623ms = 0,623s.

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

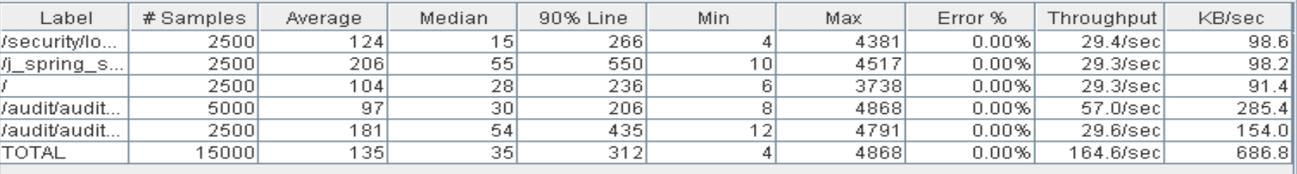
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

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.

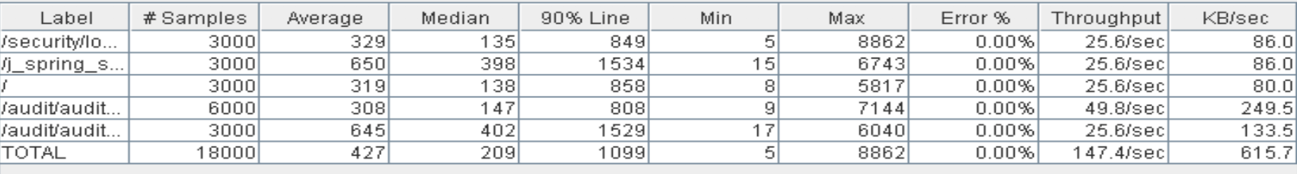
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 550 = 0.55s.

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.

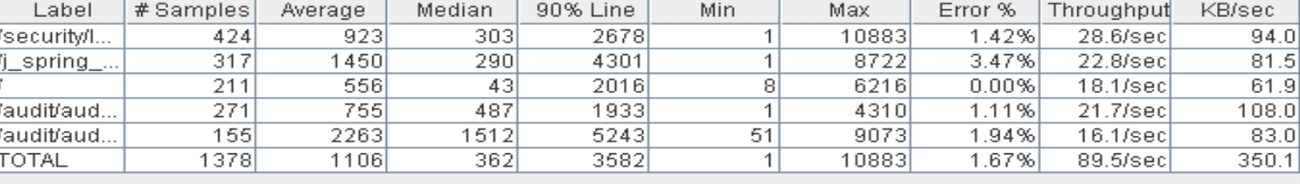
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1534ms = 1,5.

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.

**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.

3.2. An actor who is authenticated as an auditor must be able to manage his or her audits.

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

### Casos de uso

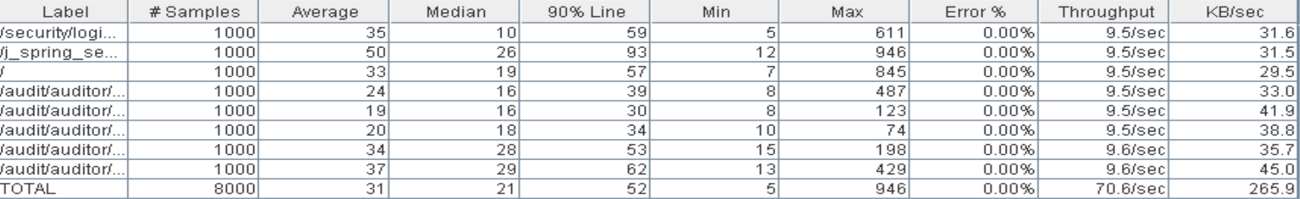
En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse
2. Ir a “Mis Audits”
3. Crear un audit
4. Editar ese audit
5. Mostrar ese audit
6. Eliminar dicho audit

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones



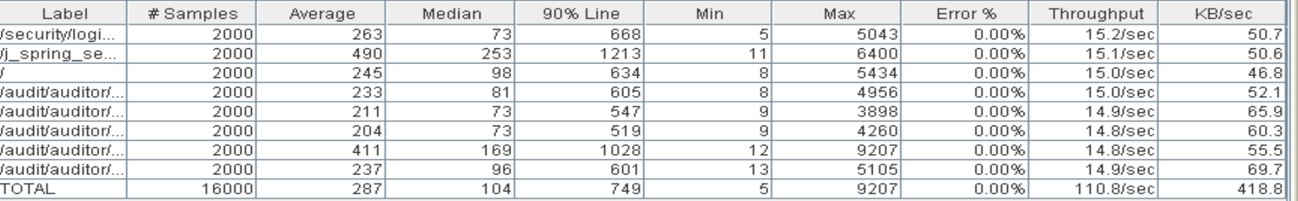
No hay errores HTTP y el tiempo medio de espera total es de 427ms = 0,42s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 93ms = 0,09s.

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

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

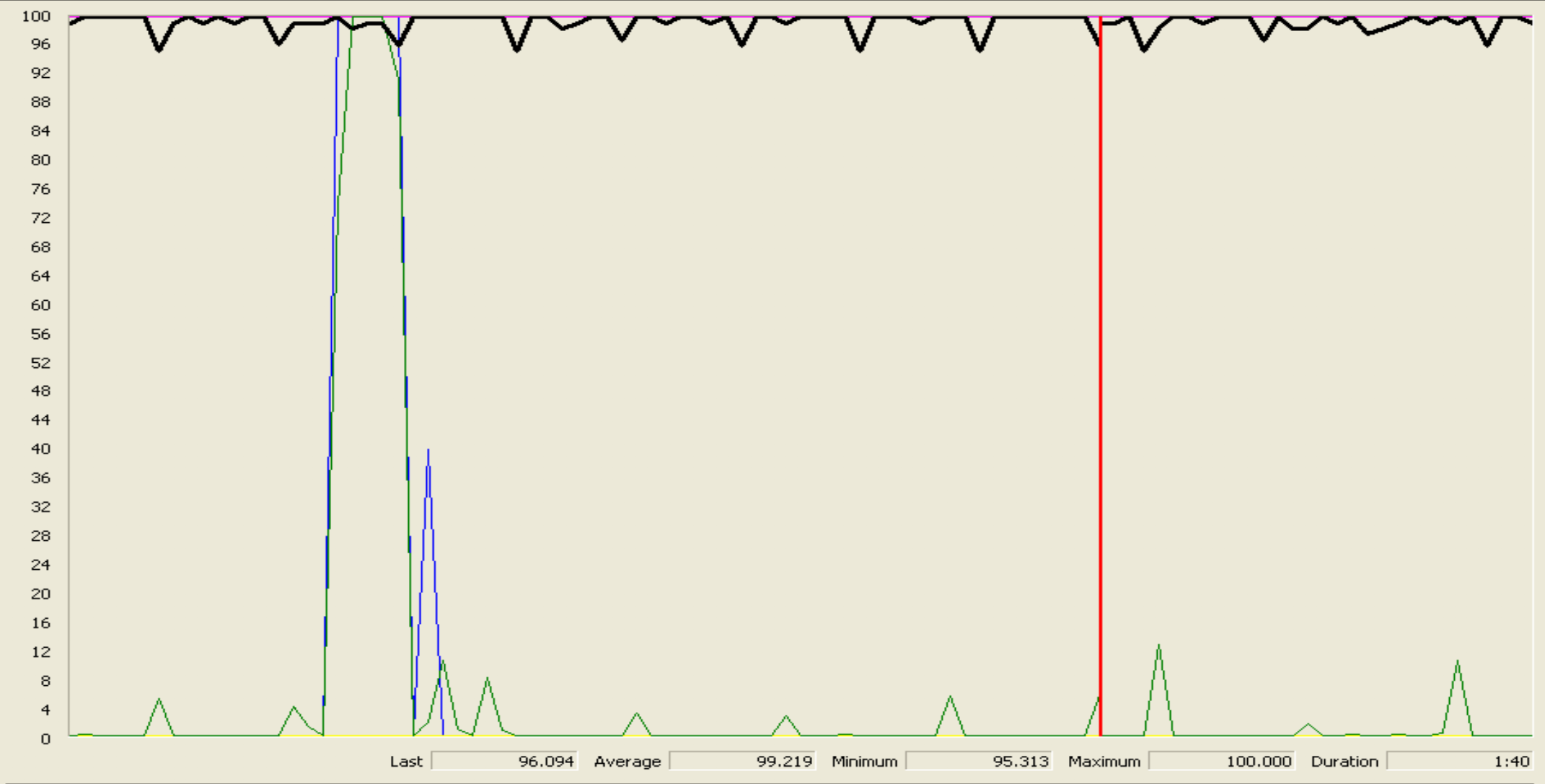


No hay errores HTTP y el tiempo medio de espera total es de 5815 = 5,8s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1213 = 1,213s.

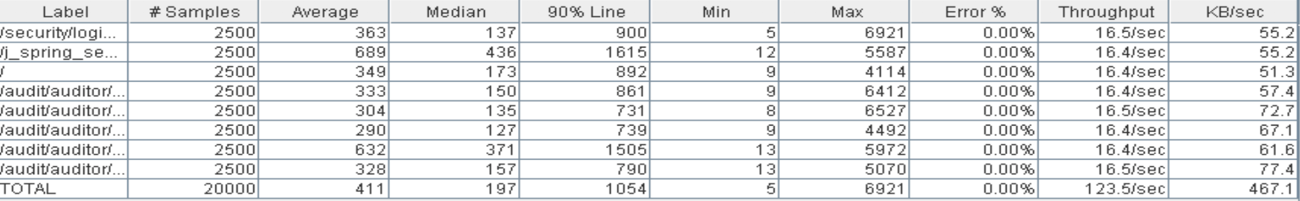
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

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 8033ms = 8,03s.

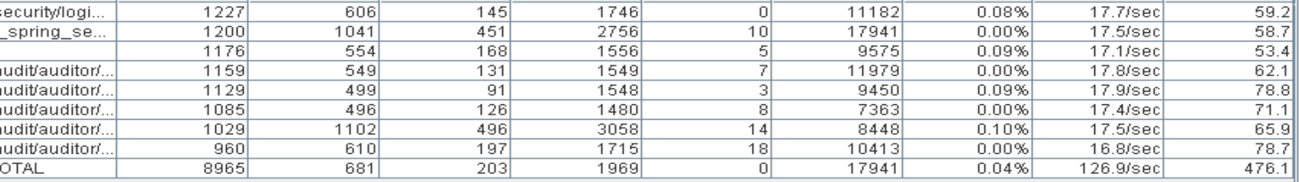
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1615ms = 1,615s.

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 15408ms = 15,4s.

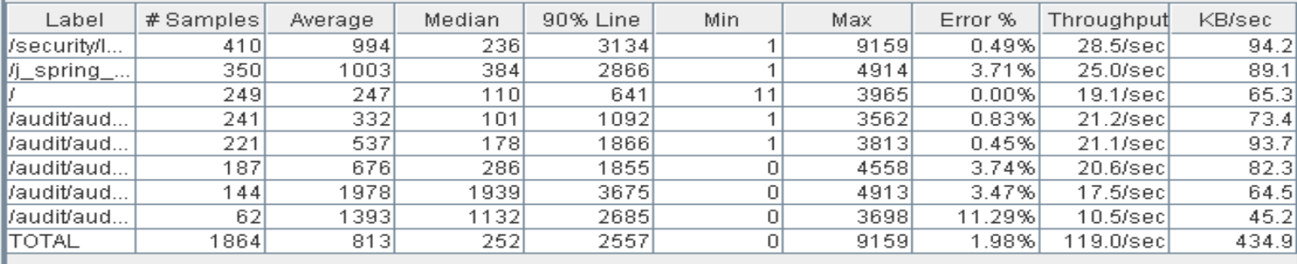
El mayor tiempo de espera se da al realizarse la operación de elominar auditor siendo de 3058ms = 3,05s.

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:**

* 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.

4.1. An actor who is authenticated as an administrator must be able to run a procedure.

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

### Casos de uso

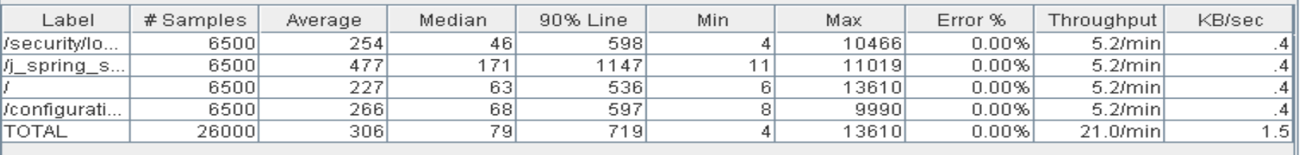
En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse
2. Acceder al menú de “Administrador” y pulsar “Calculate Audit Score”

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones



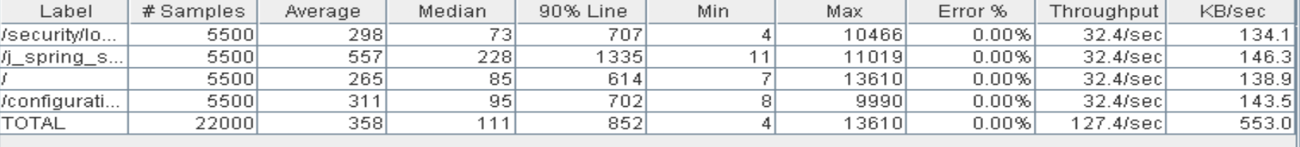
No hay errores HTTP y el tiempo medio de espera total es de 2878ms = 2,878s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1147 ms = 1,1 segs

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

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

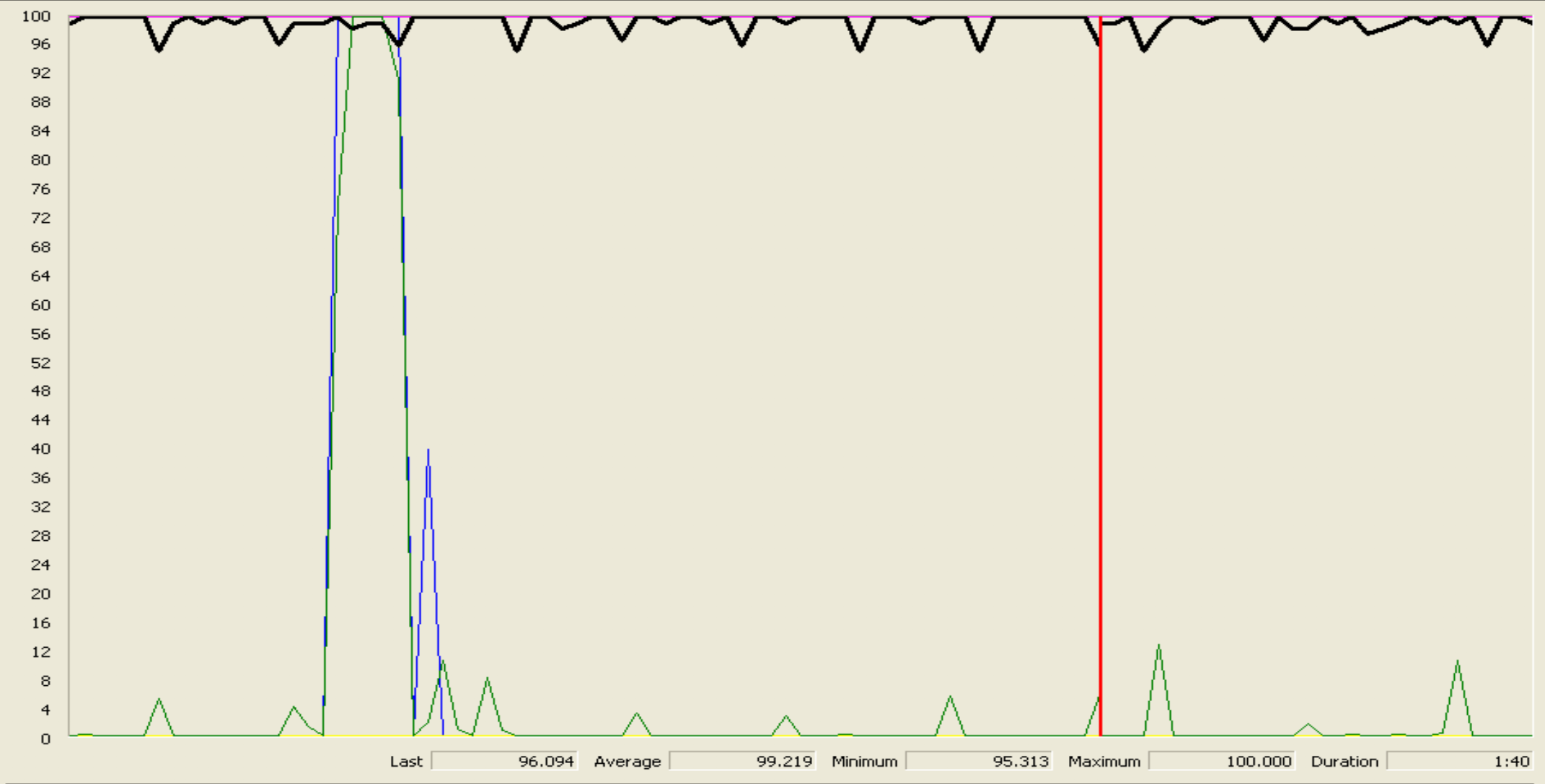


No hay errores HTTP y el tiempo medio de espera total es de 3358ms = 3,358.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1335 ms = 1,33 segs.

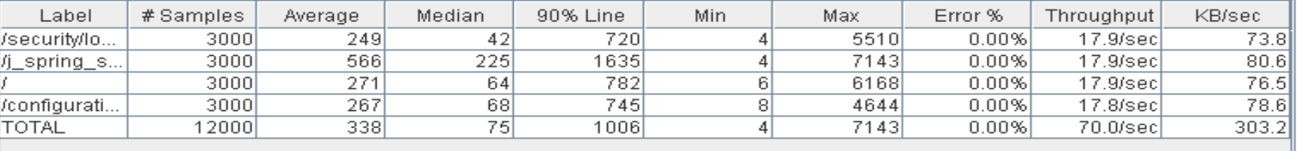
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

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 3882ms = 3,88s.

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

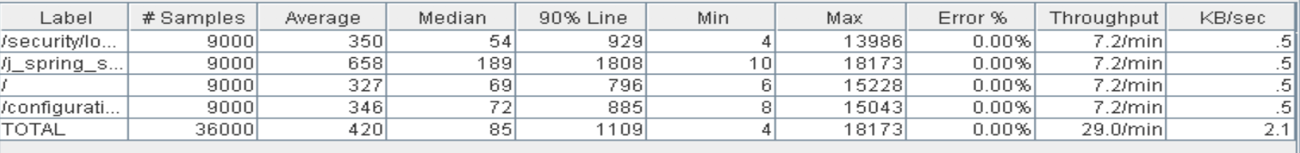
1635 ms = 1,6 segs

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 4418ms = 4,41s.

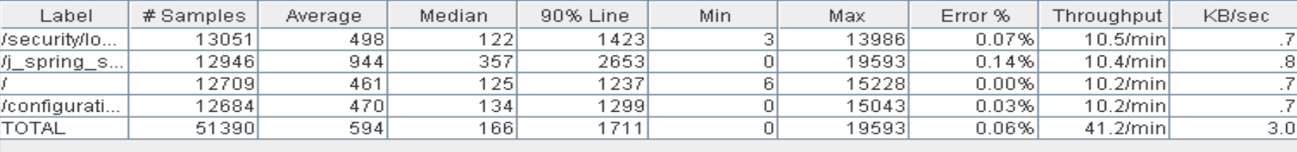
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1808ms = 1,8s.

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. 4.2. An actor who is authenticated as an administrator must be able to create user accounts for new auditors.

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

### Casos de uso

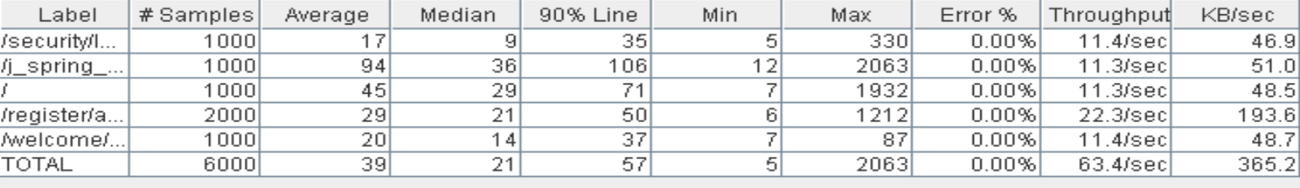
En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse
2. Acceder al menú y pulsar en Register a provider

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones



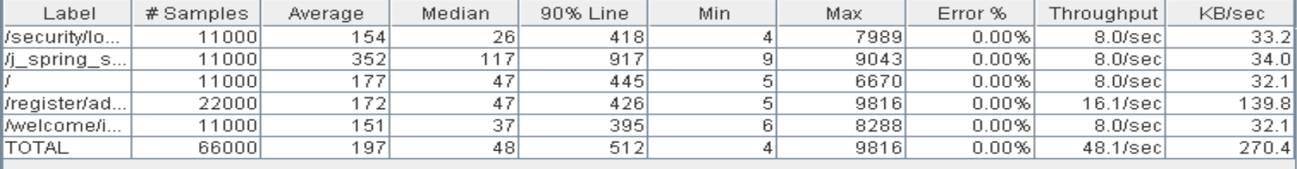
No hay errores HTTP y el tiempo medio de espera total es de 299ms = 0,29s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 106ms = 0,1.

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

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

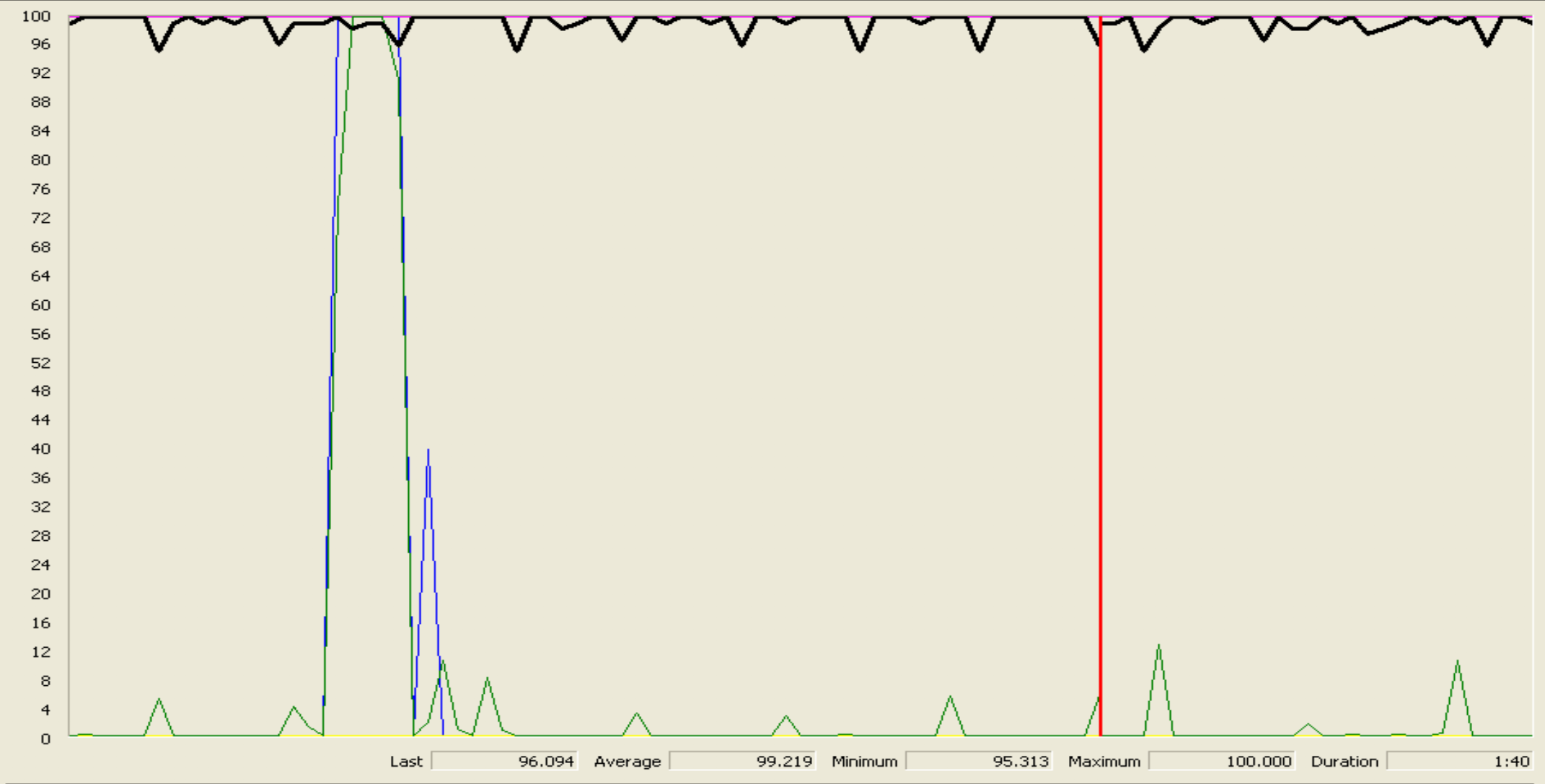


No hay errores HTTP y el tiempo medio de espera total es de 2601ms = 2,6s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 917ms = 0,9s.

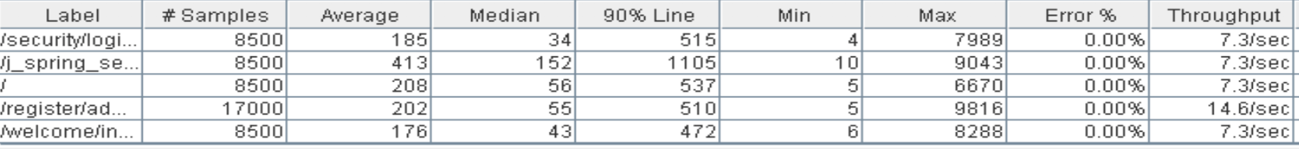
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

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 2667ms = 2,6s.

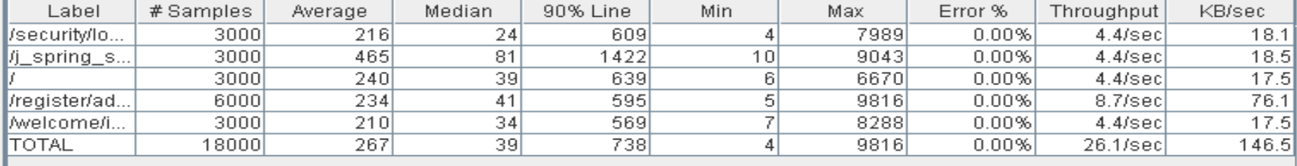
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1105ms = 1,1s.

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 3834ms = 3,834s.

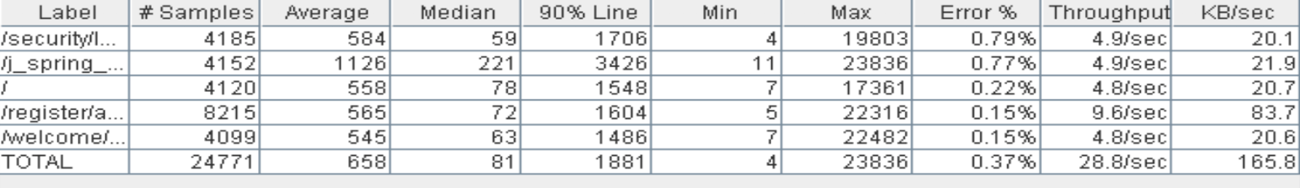
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1422ms = 1,4s.

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. 4.3. An actor who is authenticated as an administrator must be able to launch a process.

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

### Casos de uso

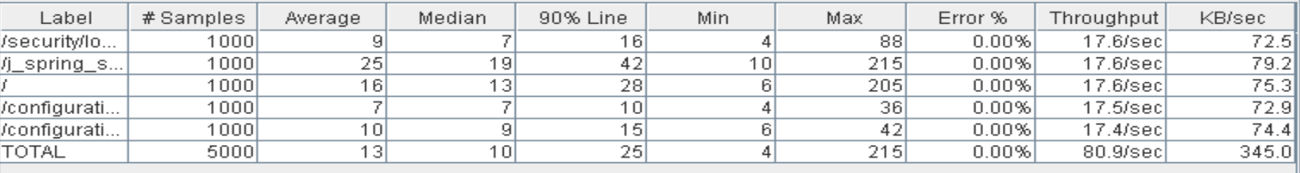
En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse
2. Acceder al menú de “Administrador” y pulsar “Change Brand”
3. Pulsar Launch Process

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones



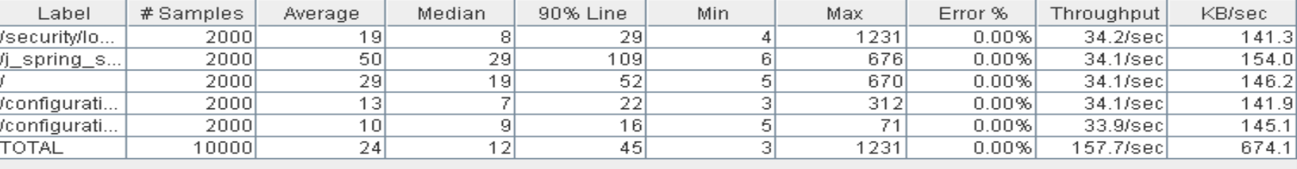
No hay errores HTTP y el tiempo medio de espera total es de 111ms = 0,11s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 42ms = 0,04s.

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

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

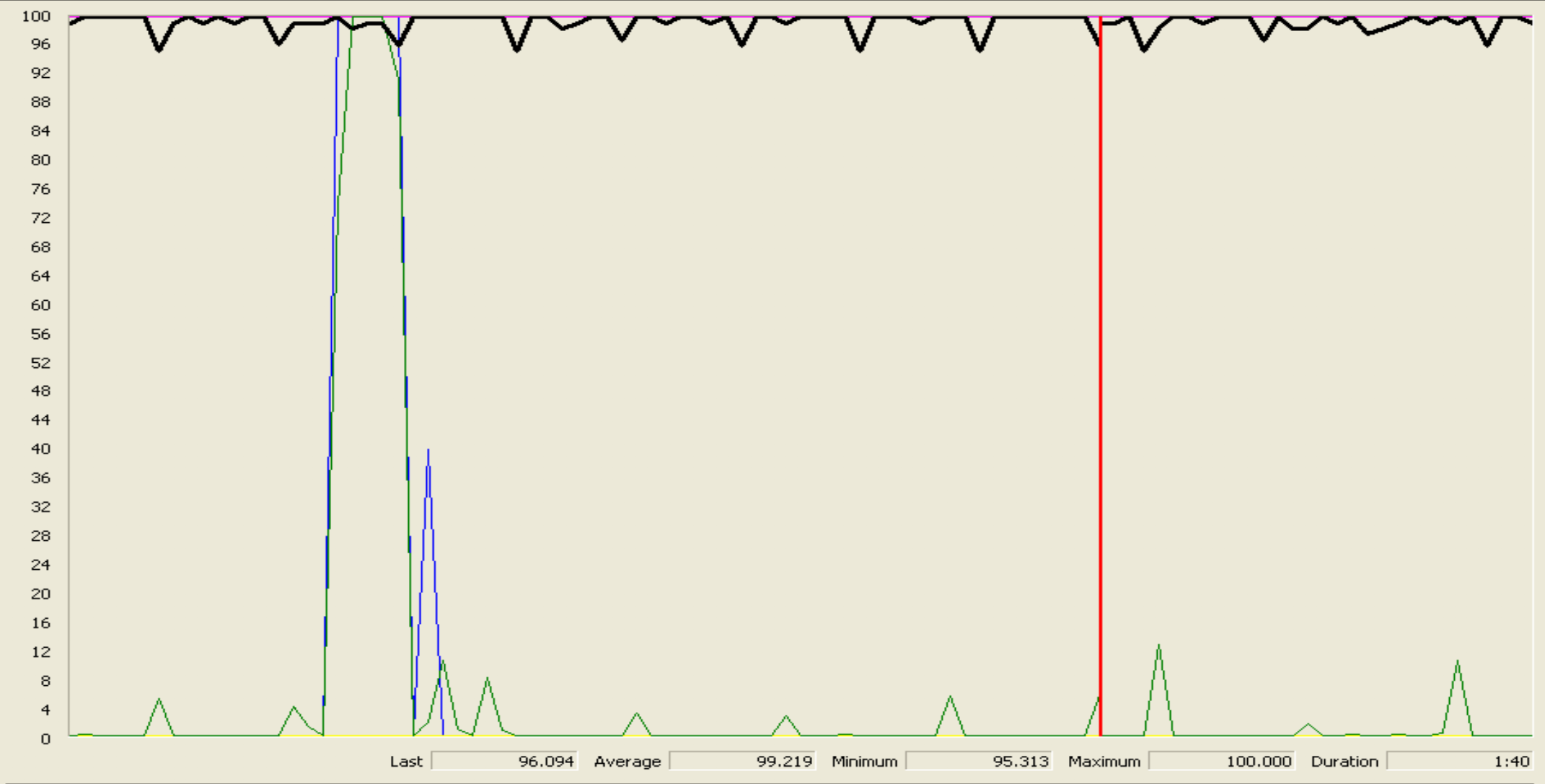


No hay errores HTTP y el tiempo medio de espera total es de 228ms = 0,228s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 109ms = 0,1.

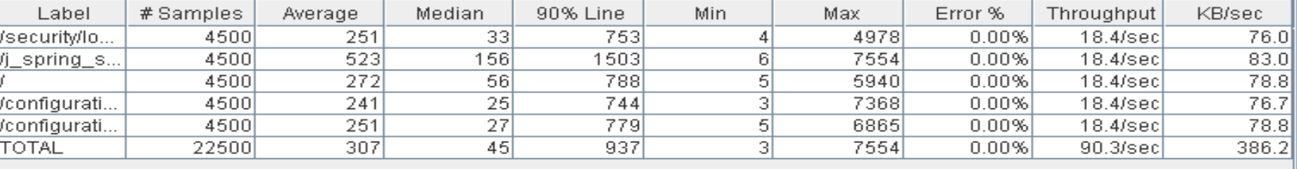
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

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 4567ms = 4,5.

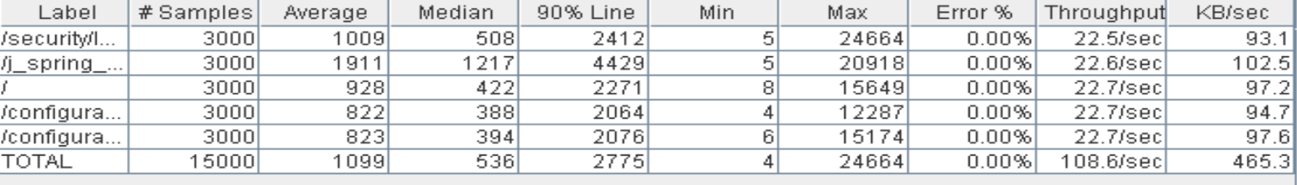
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1503ms = 1,5s.

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 13252ms = 13,252.

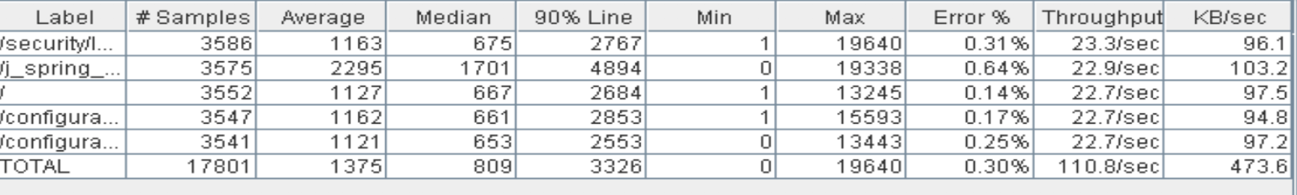
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 4429ms = 4,4s.

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. 9.3. An actor who is not authenticated must be able to register to the system as a provider.

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

### Casos de uso

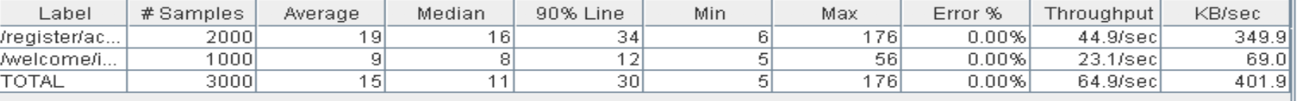
En este caso tenemos un caso de uso con los siguientes pasos:

1. No loguearse
2. Acceder al menú y pulsar en Register a provider

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones



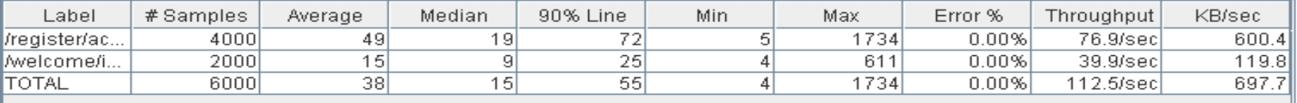
No hay errores HTTP y el tiempo medio de espera total es de 46ms = 0,046s.

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

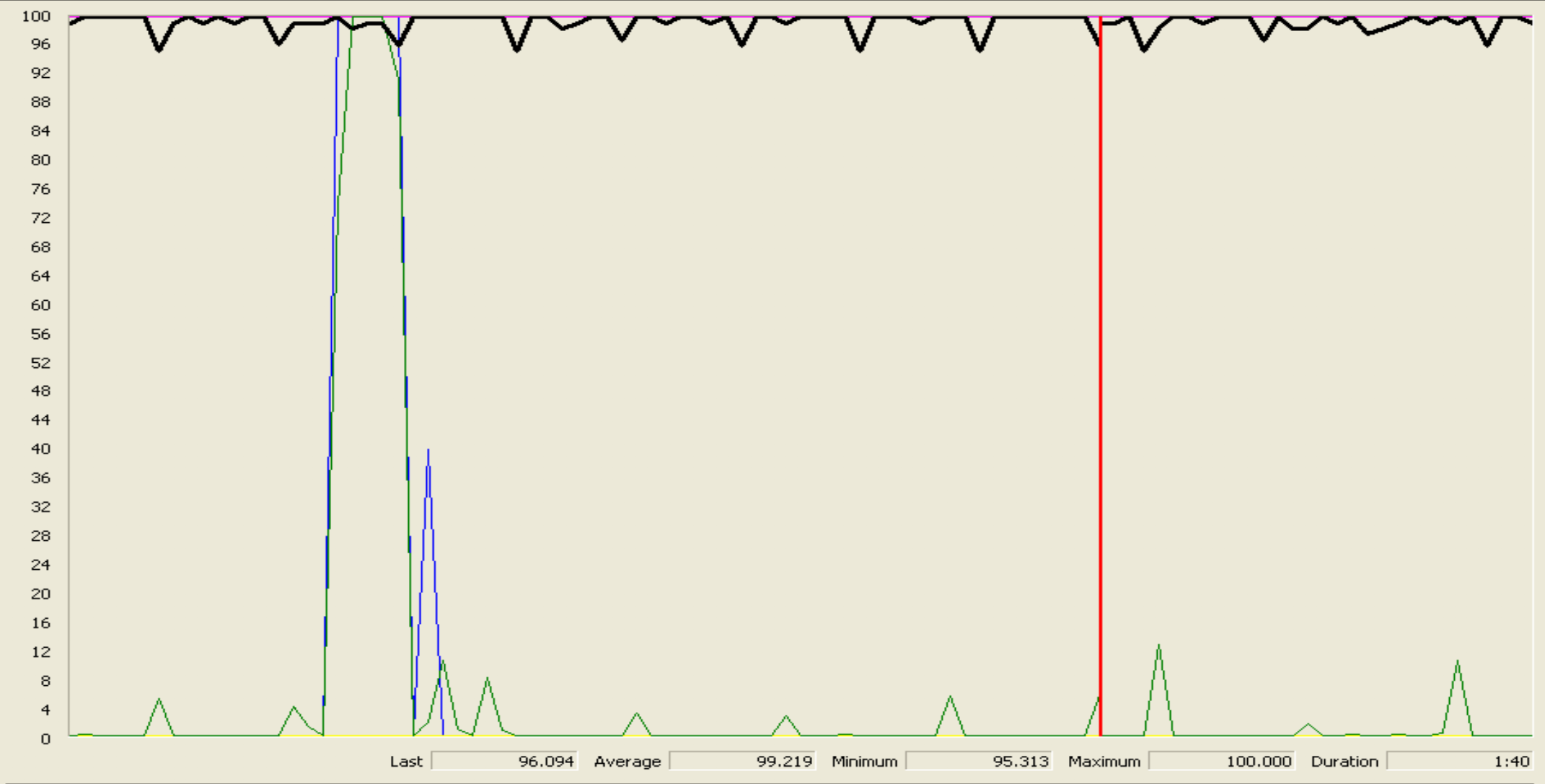


No hay errores HTTP y el tiempo medio de espera total es de 97ms = 0,09s.

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

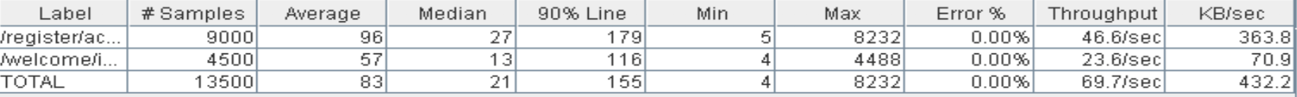
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

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.

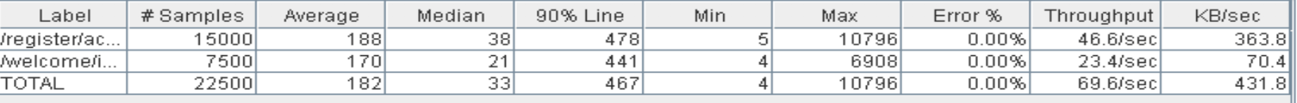
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 179ms = 0,179segs.

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



No hay errores HTTP y el tiempo medio de espera total es de 919ms = 0,919s.

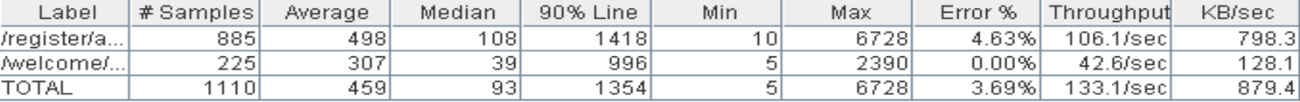
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 478ms = 0,478segs.

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. 10.1. An actor who is authenticated as a provider must be able to manage his or her catalogue of items.

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

### Casos de uso

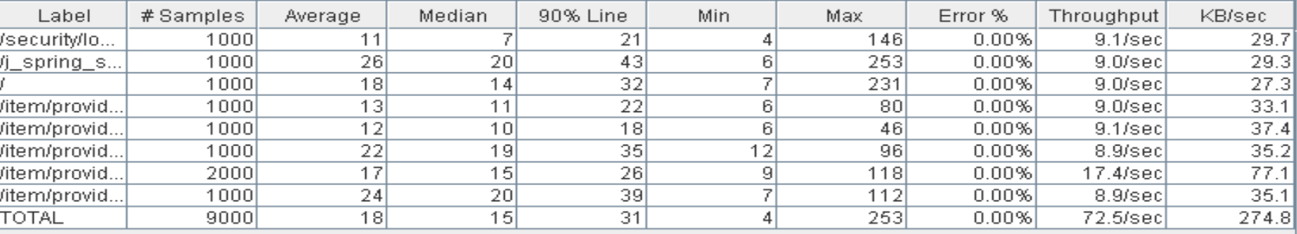
En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse
2. Listar mis items
3. Crear una item
4. Editar ese item
5. Ver ese ítem
6. Eliminar ese item

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones



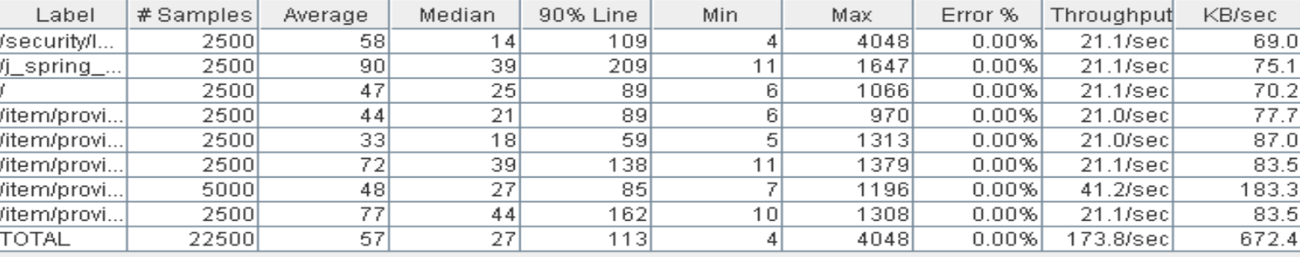
No hay errores HTTP y el tiempo medio de espera total es de 236ms = 0,236s.

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

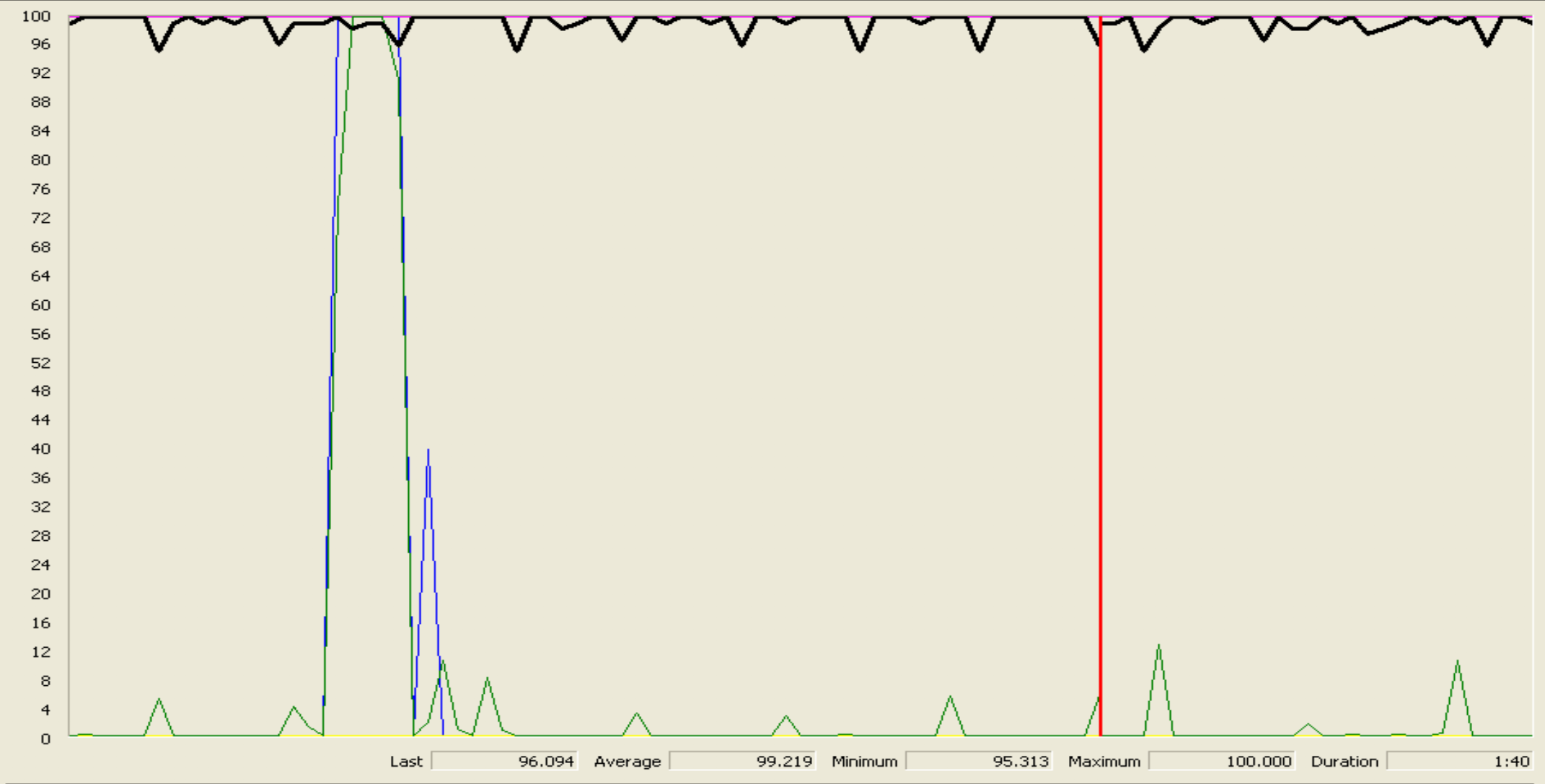


No hay errores HTTP y el tiempo medio de espera total es de 881ms = 0,81s.

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

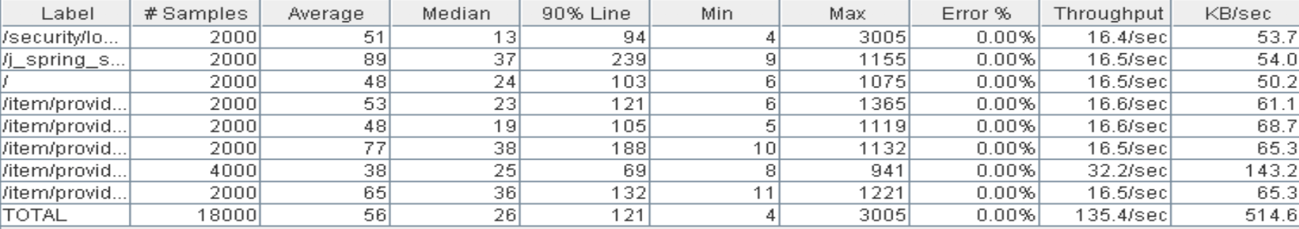
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

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.

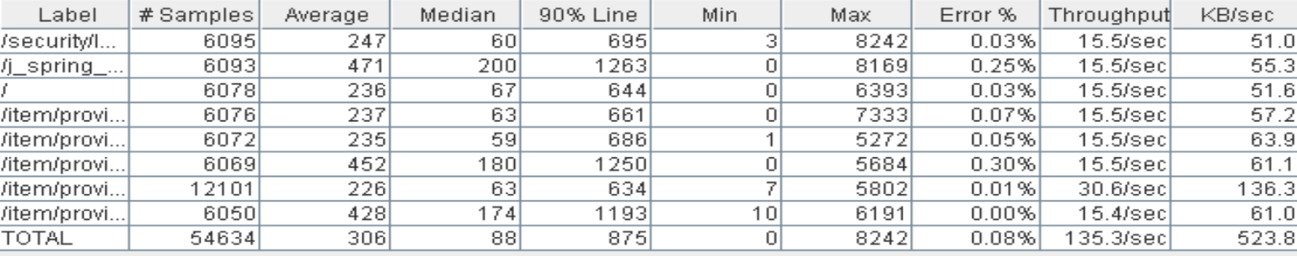
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 239ms = 0,239s.

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.

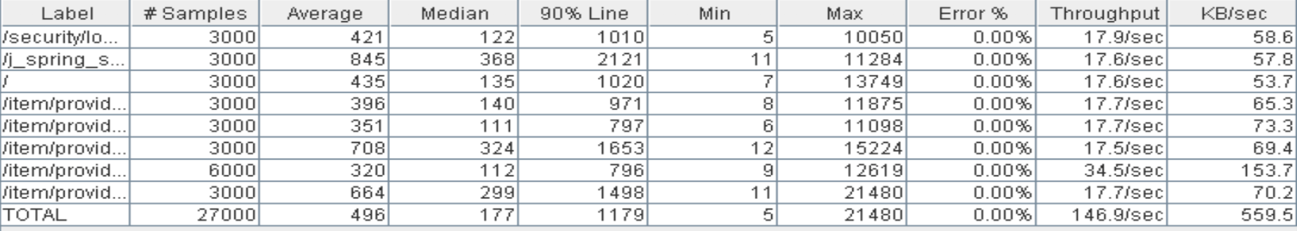
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1263ms = 1,263s.

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.