

TESTS DE RENDIMIENTO

Grupo 21



Contenido

[1. Descripción del ordenador 1](#_Toc7025405)

[1.1. Descripción de máquina virtual 1](#_Toc7025406)

[1.2. Descripción de máquina física 2](#_Toc7025407)

[2. Casos de uso e informes 3](#_Toc7025408)

[2.1. Application - Hacker (R.10.1) 3](#_Toc7025409)

[2.1.1. Casos de uso 3](#_Toc7025410)

[2.1.2. Pruebas realizadas 3](#_Toc7025411)

[2.1.3. Conclusiones 5](#_Toc7025412)

[2.2. Finder - Hacker (R.17.2) 5](#_Toc7025413)

[2.2.1. Casos de uso 5](#_Toc7025414)

[2.2.2. Pruebas realizadas 5](#_Toc7025415)

[2.2.3. Conclusiones 7](#_Toc7025416)

[2.3. Curricula - Hacker (R.17.2) 8](#_Toc7025417)

[2.3.1. Casos de uso 8](#_Toc7025418)

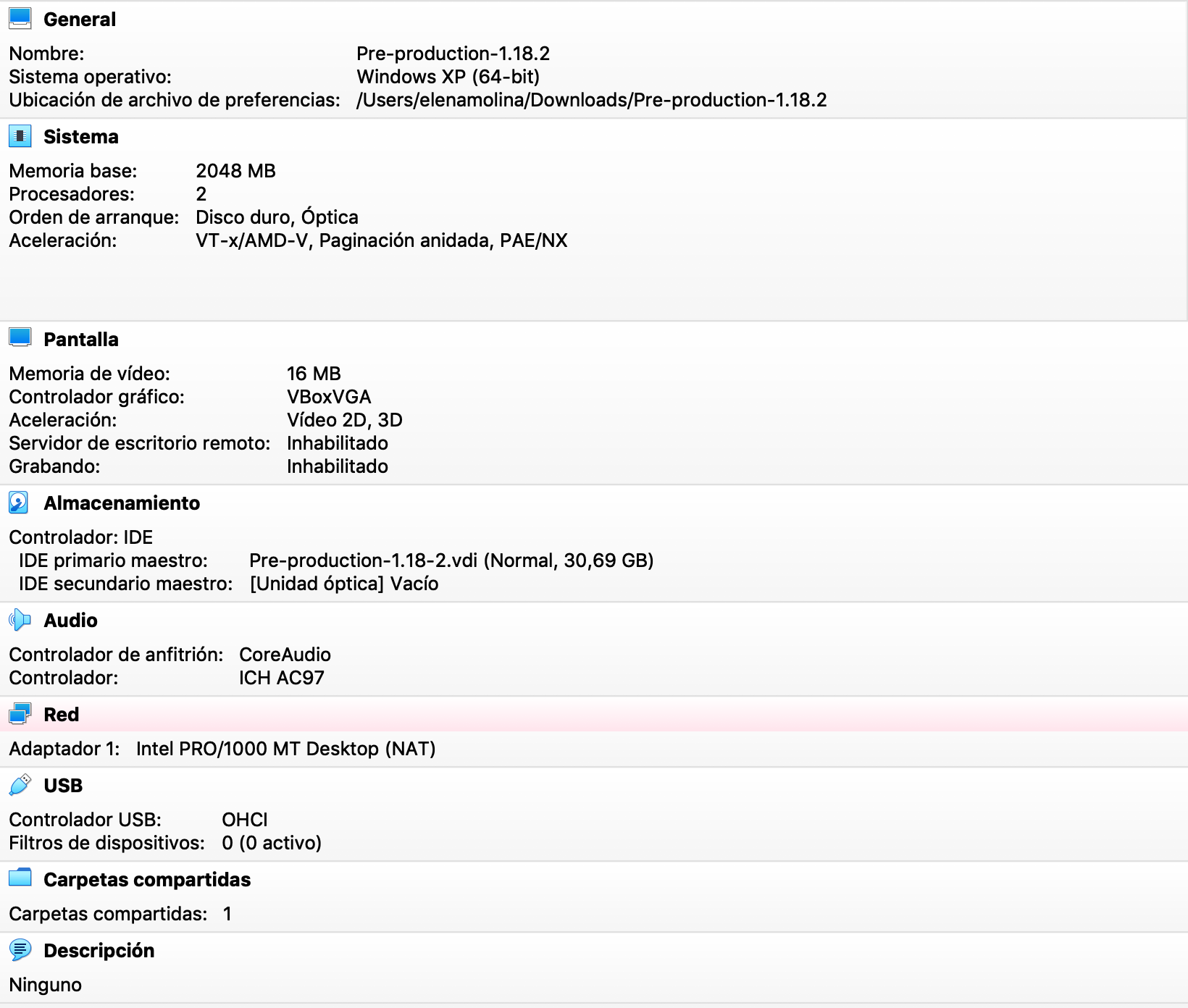
[2.3.2. Pruebas realizadas 8](#_Toc7025419)

[2.3.3. Conclusiones 11](#_Toc7025420)

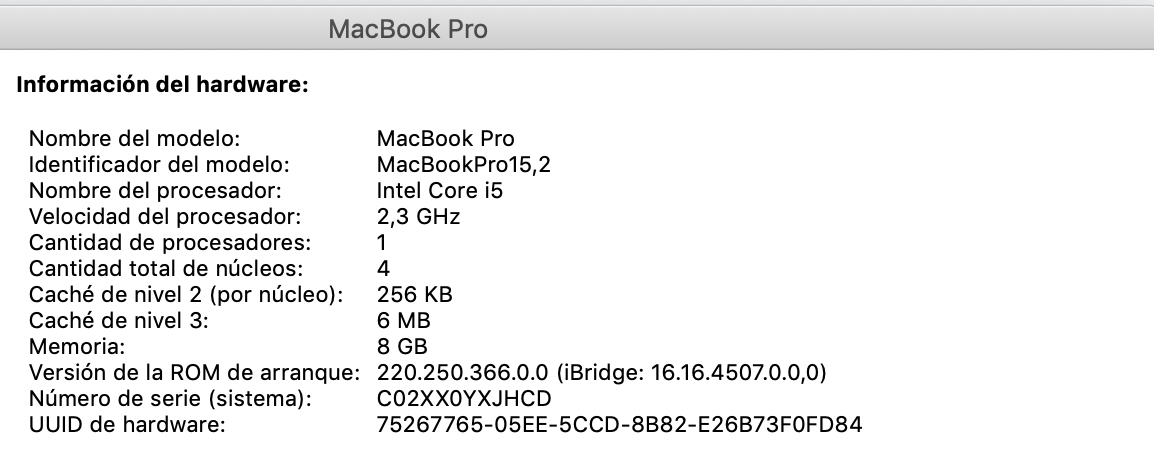
# Descripción del ordenador

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

## Application - Hacker (R.10.1)

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

### Casos de uso

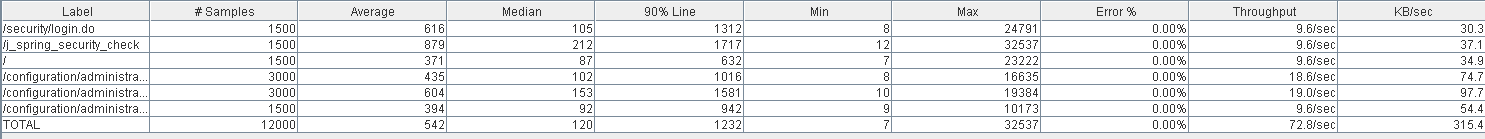
En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse
2. Listar mis solicitudes
3. Crear una solicitud en estado ‘PENDING’
4. Editar esa solicitud con estado ‘SUBMITTED’
5. Ver esa solicitud

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones



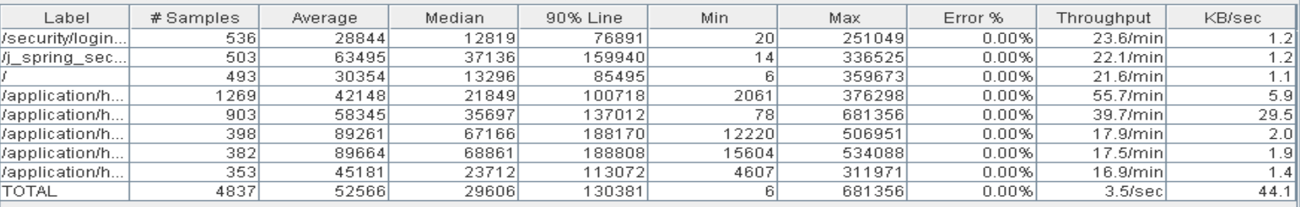
No hay errores HTTP y el tiempo medio de espera total es de 7200ms = 7,2s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1717ms = 1,17s.

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

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

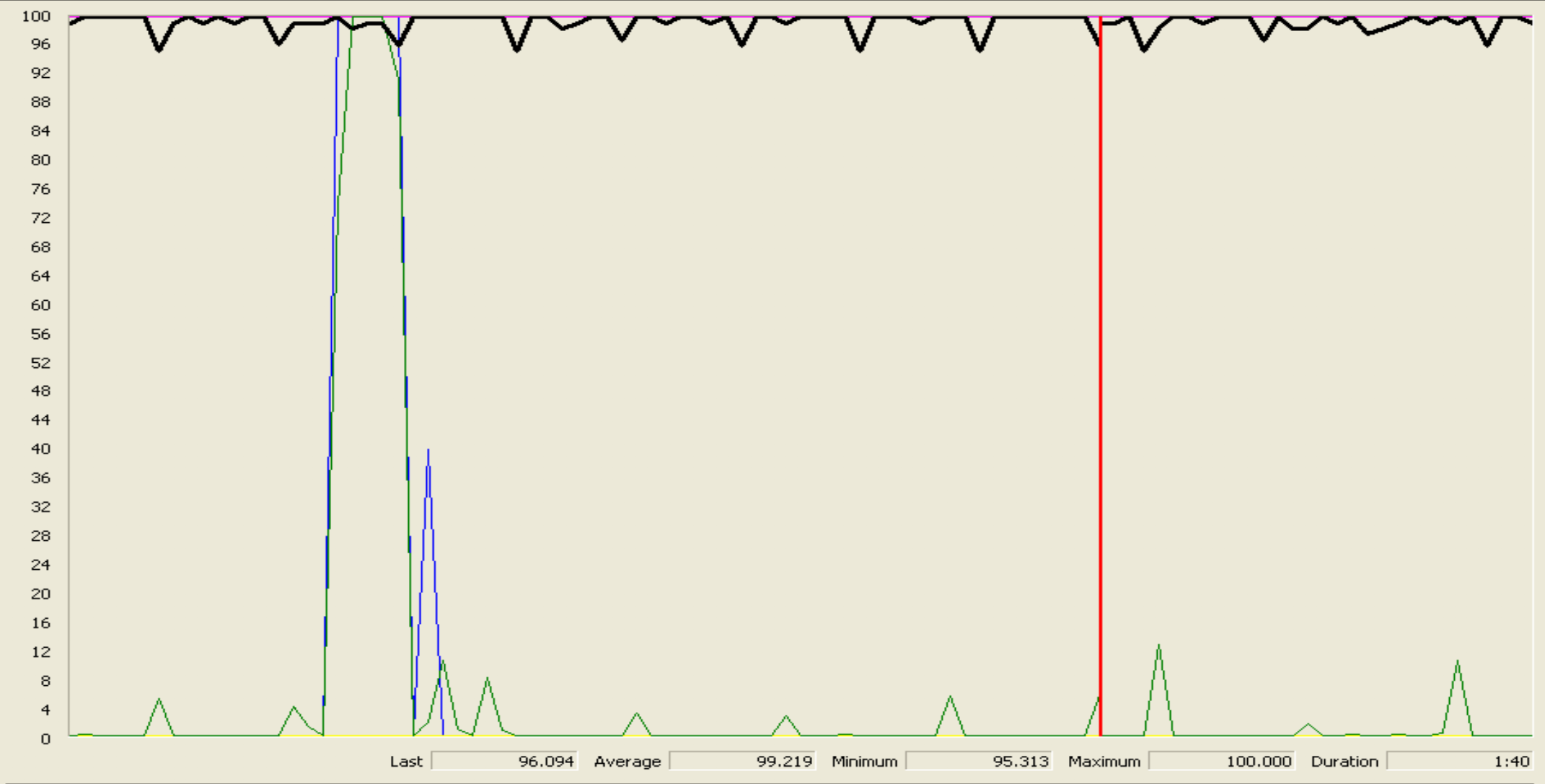


No hay errores HTTP y el tiempo medio de espera total es de 8100ms = 8,1s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 2207ms = 2,207s.

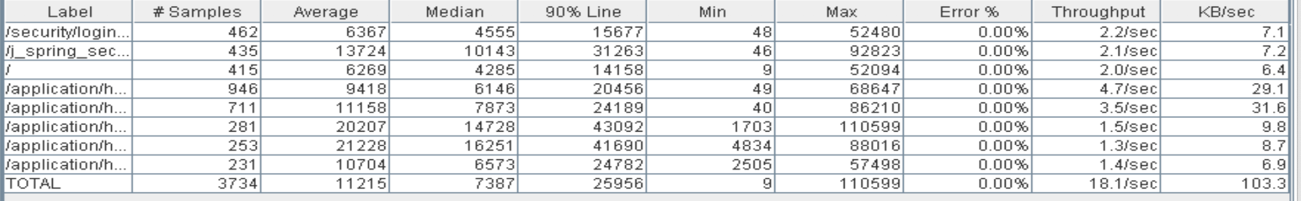
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

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 12388ms = 12,388s.

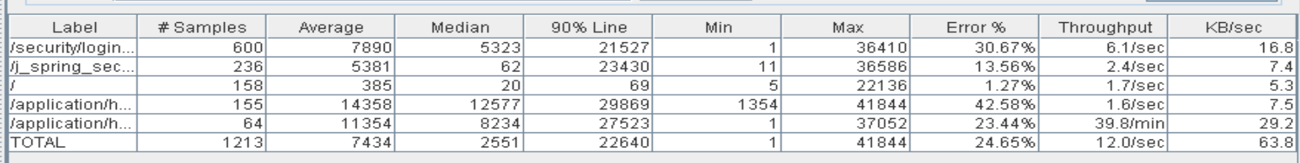
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 3255ms = 3,255s.

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 200 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 250 y 300 usuarios simultáneos.

## Finder - Hacker (R.17.2)

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

### Casos de uso

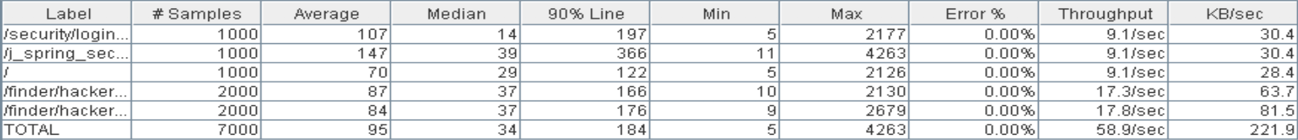
En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse
2. Ir a finder
3. Editar finder
4. Listar resultados

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones



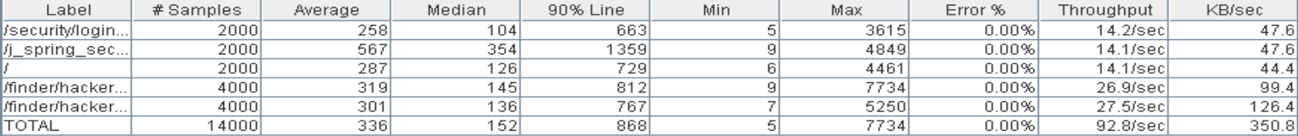
No hay errores HTTP y el tiempo medio de espera total es de 7200ms = 7,2s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1717ms = 1,17s.

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

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

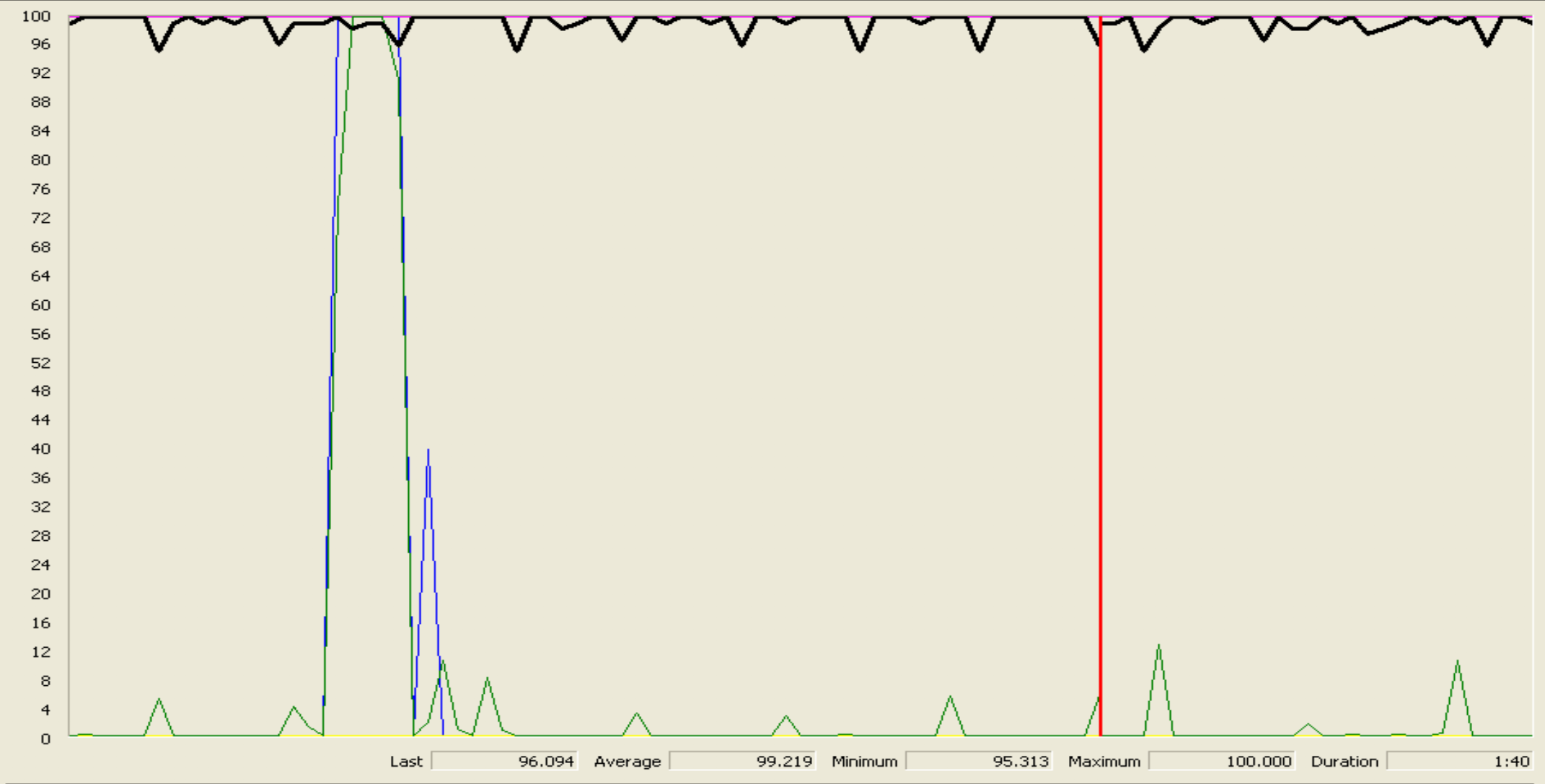


No hay errores HTTP y el tiempo medio de espera total es de 8100ms = 8,1s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 2207ms = 2,207s.

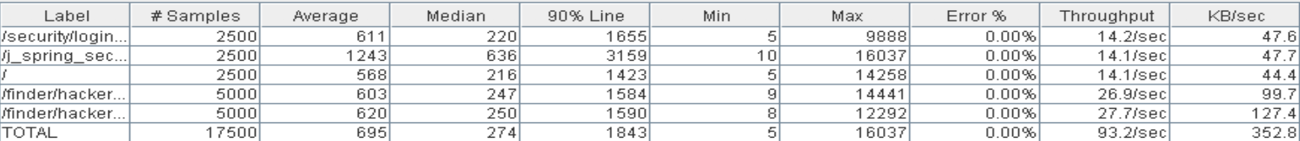
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

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 12388ms = 12,388s.

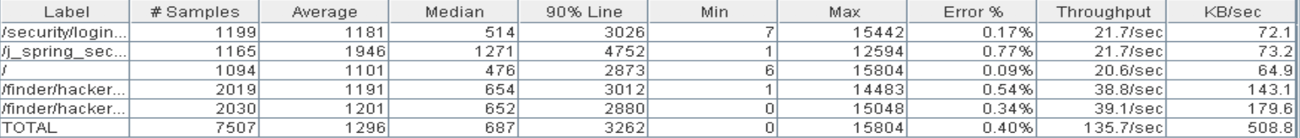
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 3255ms = 3,255s.

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 200 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 250 y 300 usuarios simultáneos.

## Curricula - Hacker (R.17.2)

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

### Casos de uso

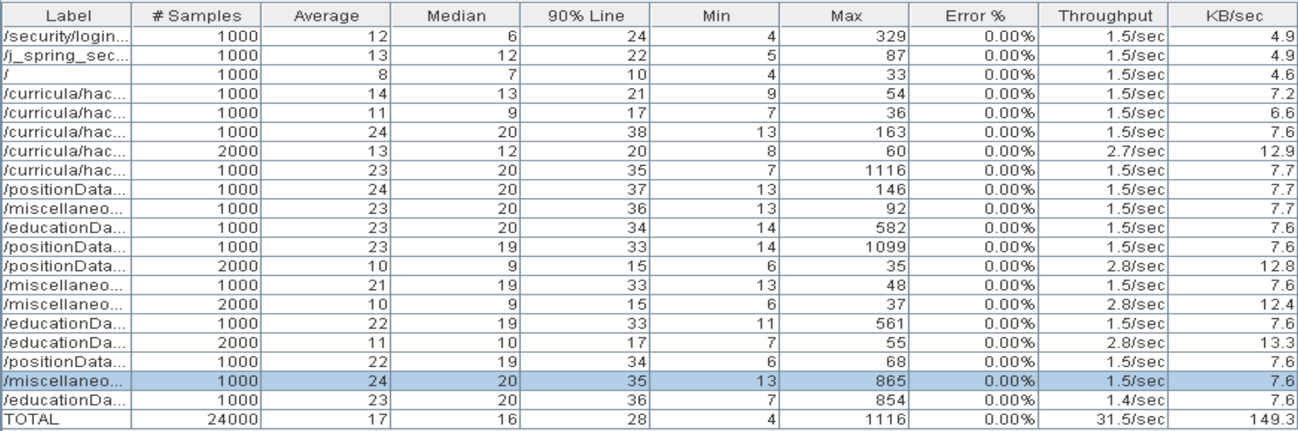
En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse
2. Listar curriculas
3. Crear una curricula
4. Editar la curricula
5. Mostrar la curricula
6. Crear datos de posición
7. Crear datos varios
8. Crear datos de educación
9. Editar datos de posición
10. Editar datos varios
11. Editar datos de educación
12. Mostrar datos de posición
13. Mostrar datos varios
14. Mostrar datos de educación
15. Eliminar datos de posición
16. Eliminar datos varios
17. Eliminar datos de educación
18. Eliminar curricula

### Pruebas realizadas

**Prueba 1:**

* 100 usuarios
* Loop de 10 iteraciones



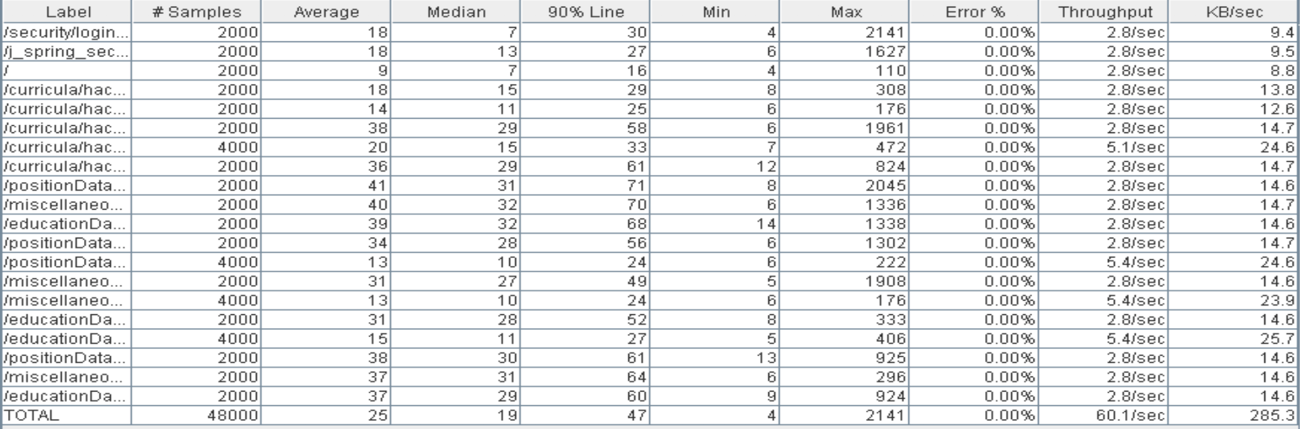
No hay errores HTTP y el tiempo medio de espera total es de 7200ms = 7,2s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1717ms = 1,17s.

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

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

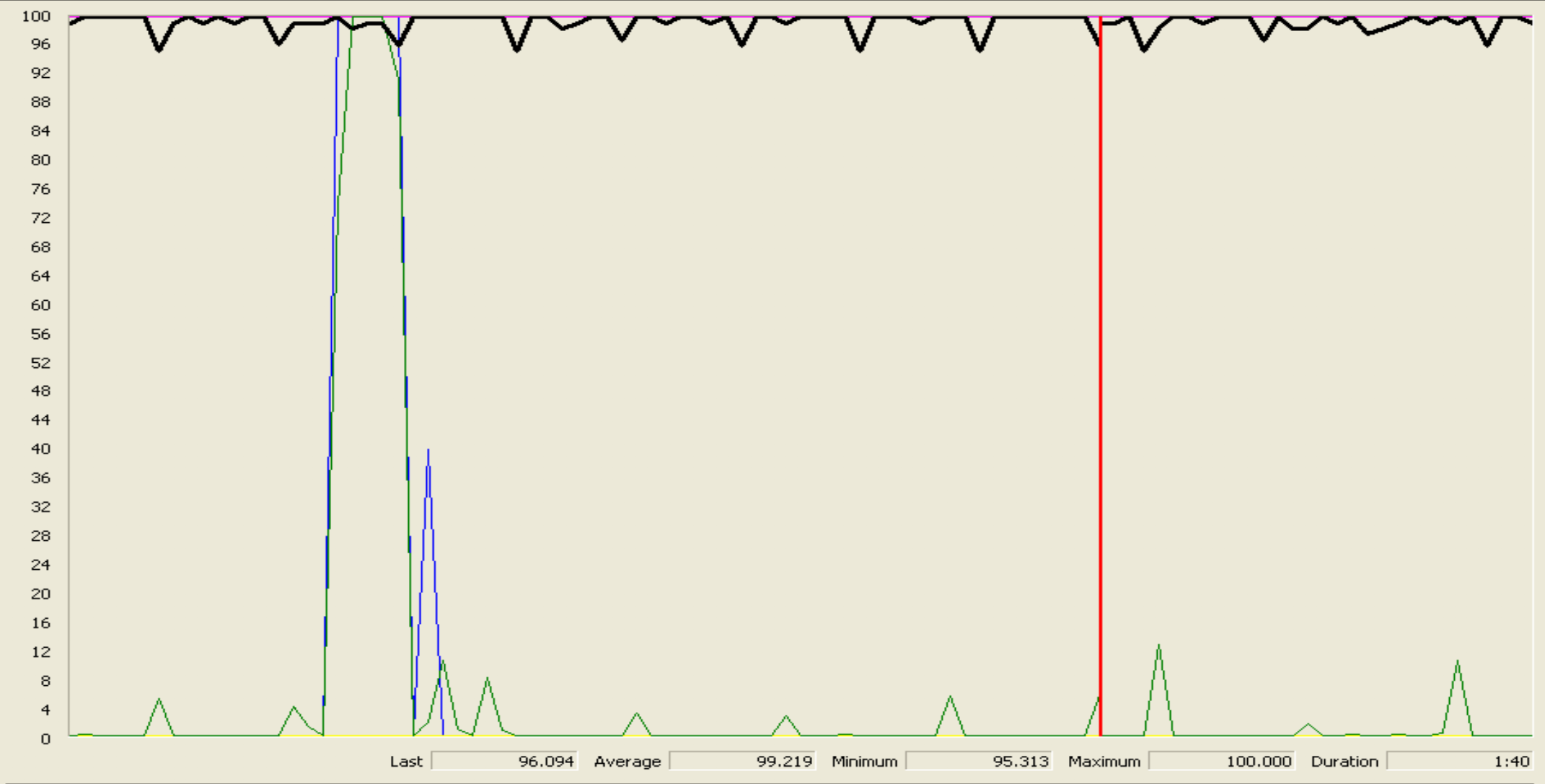


No hay errores HTTP y el tiempo medio de espera total es de 8100ms = 8,1s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 2207ms = 2,207s.

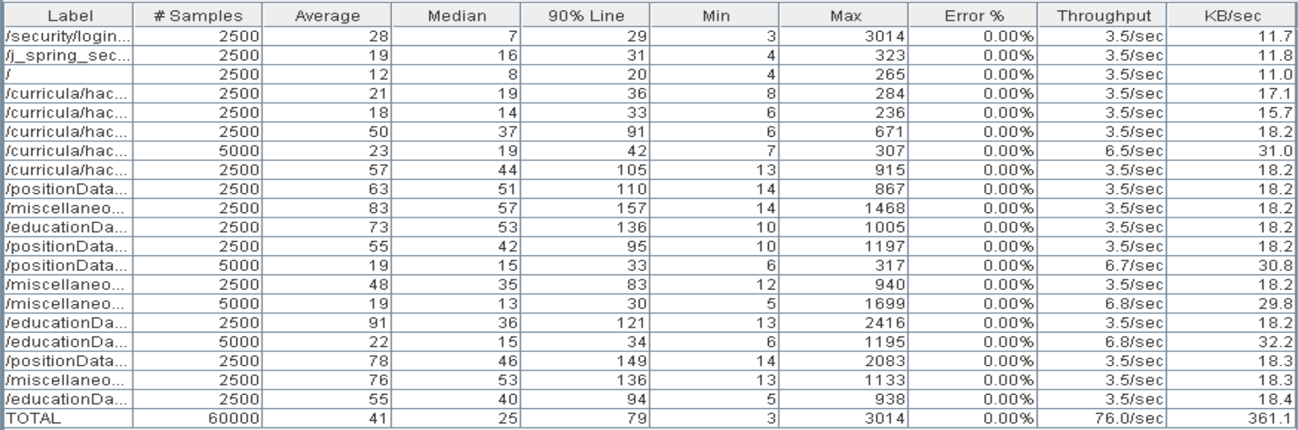
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

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 12388ms = 12,388s.

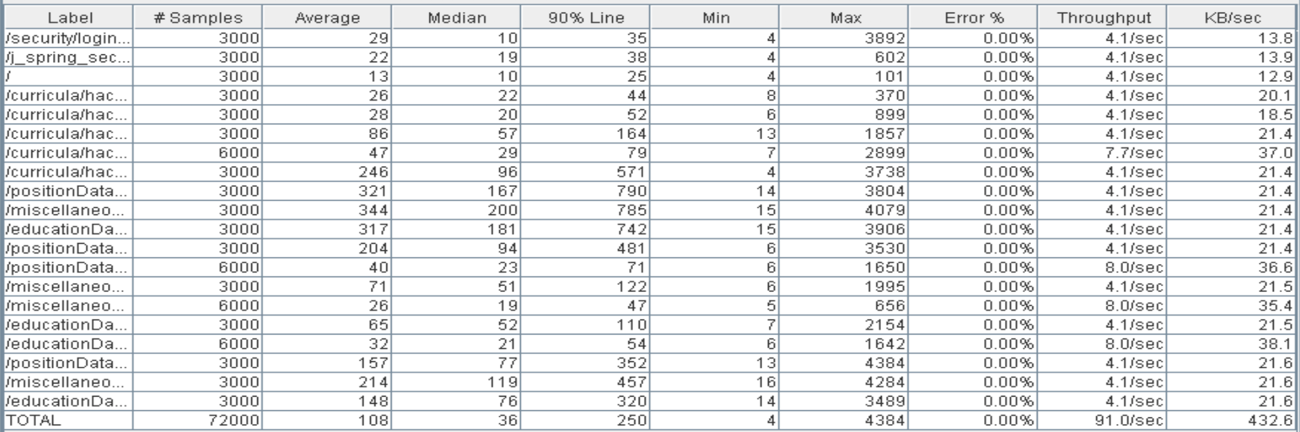
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 3255ms = 3,255s.

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 200 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 250 y 300 usuarios simultáneos.