

TESTS DE RENDIMIENTO

Grupo 21



Contenido

[1. Descripción del ordenador 2](#_Toc6685503)

[1.1. Descripción de máquina virtual 2](#_Toc6685504)

[1.2. Descripción de máquina física 2](#_Toc6685505)

[2. Casos de uso e informes 3](#_Toc6685506)

[2.1. Configuration (R.14) 3](#_Toc6685507)

[2.1.1. Casos de uso 3](#_Toc6685508)

[2.1.2. Pruebas realizadas 3](#_Toc6685509)

[2.1.3. Conclusiones 5](#_Toc6685510)

[2.2. Register (R.7.1) 5](#_Toc6685511)

[2.2.1. Casos de uso 5](#_Toc6685512)

[2.2.2. Pruebas realizadas 6](#_Toc6685513)

[2.2.3. Conclusiones 7](#_Toc6685514)

[2.3. List positions and display company (R.7.2) 7](#_Toc6685515)

[2.3.1. Casos de uso 7](#_Toc6685516)

[2.3.2. Pruebas realizadas 8](#_Toc6685517)

[2.3.3. Conclusiones 9](#_Toc6685518)

[2.4. List companies and display positions (R.7.3) 9](#_Toc6685519)

[2.4.1. Casos de uso 9](#_Toc6685520)

[2.4.2. Pruebas realizadas 9](#_Toc6685521)

[2.4.3. Conclusiones 10](#_Toc6685522)

[2.5. Edit personal data (R.8.2) 11](#_Toc6685523)

[2.5.1. Casos de uso 11](#_Toc6685524)

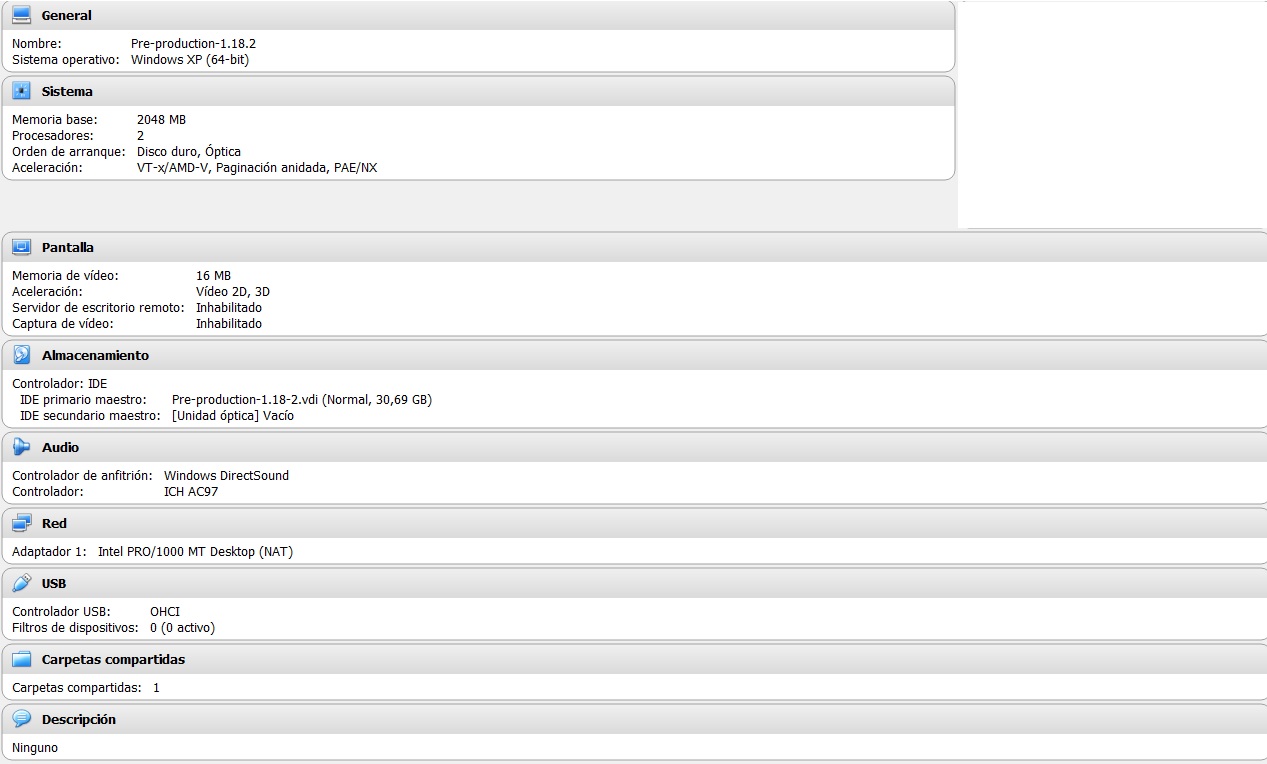
[2.5.2. Pruebas realizadas 11](#_Toc6685525)

[2.5.3. Conclusiones 12](#_Toc6685526)

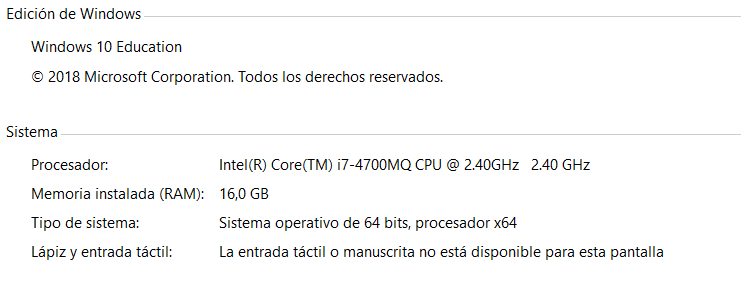
# Descripción del ordenador

Máquina virtual ejecutada con Oracle VM VirtualBox.

## Descripción de máquina virtual



## Descripción de máquina física



Además de ello dispone de una tarjeta de red Intel® Centrino® Wireless-N 2230

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

# Casos de uso e informes

## Configuration (R.14)

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

### Casos de uso

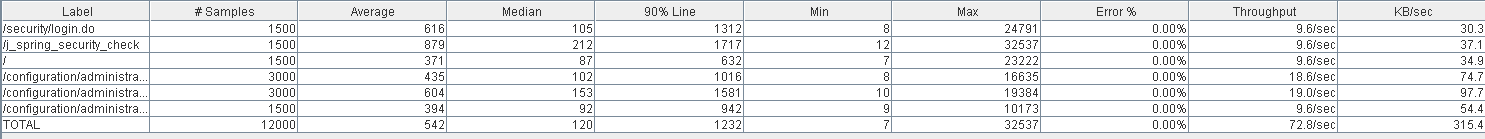
En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse
2. Ir a ventana de configuración
3. Editar la configuración
4. Ver la configuración

### Pruebas realizadas

**Prueba 1:**

* 150 usuarios
* Loop de 10 iteraciones



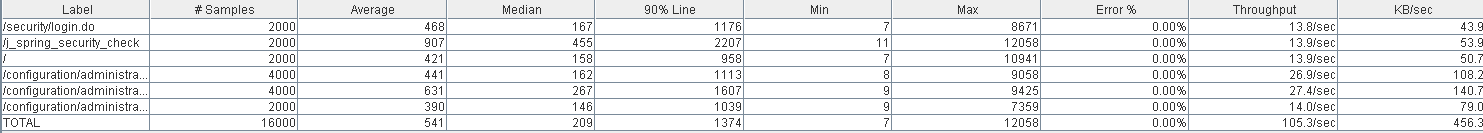
No hay errores HTTP y el tiempo medio de espera total es de 7200ms = 7,2s.

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

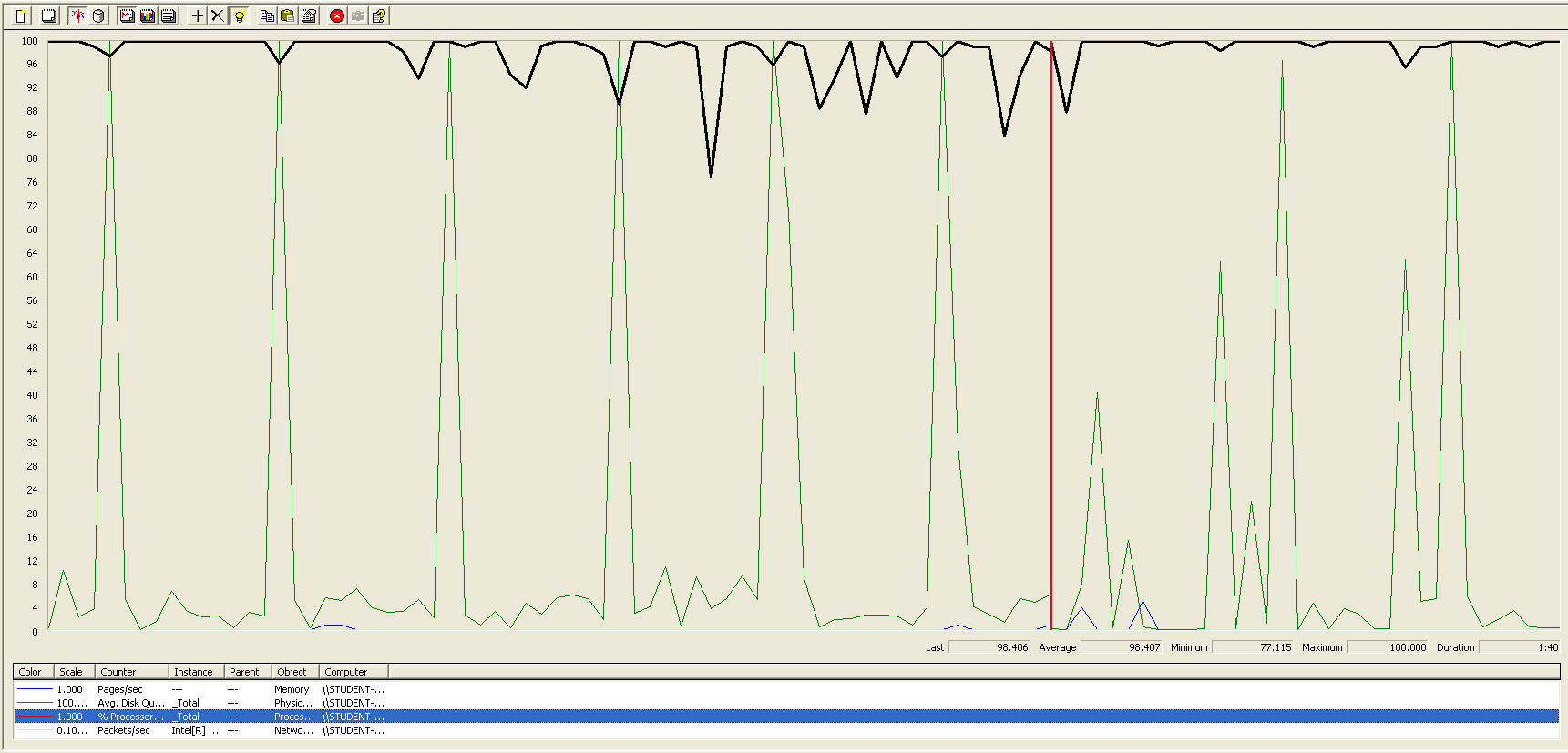


No hay errores HTTP y el tiempo medio de espera total es de 8100ms = 8,1s.

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

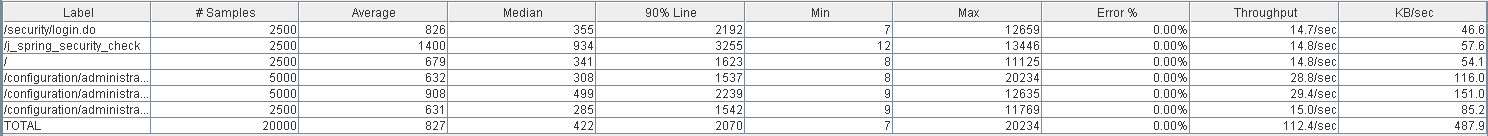
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

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.

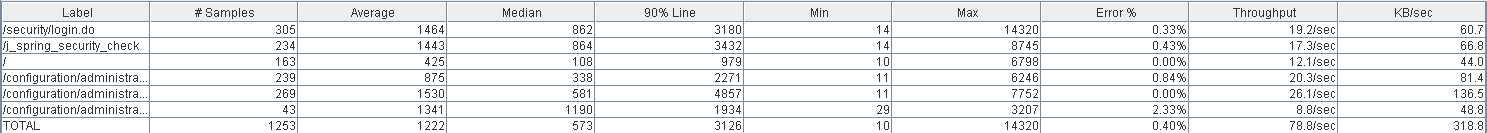
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 3255ms = 3,255s.

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.

## Register (R.7.1)

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

### Casos de uso

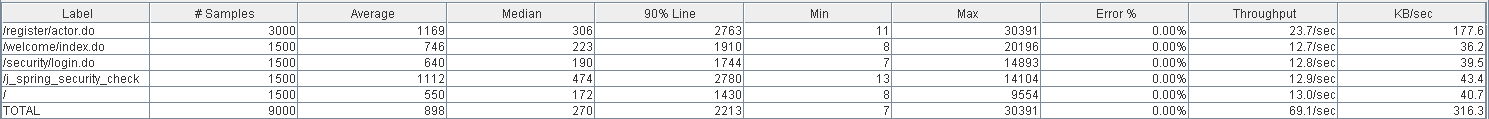
En este caso tenemos un caso de uso con los siguientes pasos:

1. Registrarse
2. Loguearse

### Pruebas realizadas

**Prueba 1:**

* 150 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 10627ms = 10,627s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 2780ms = 2,78s.

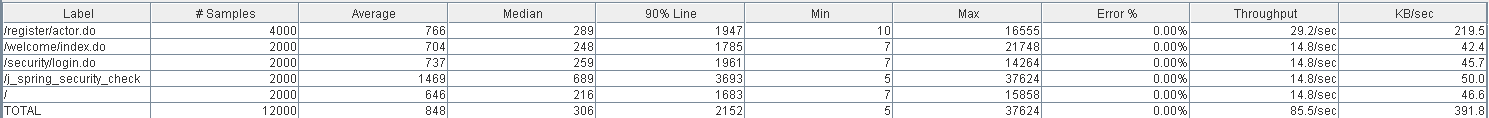
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable pero no óptimo.

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



**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 11069ms = 11,069s.

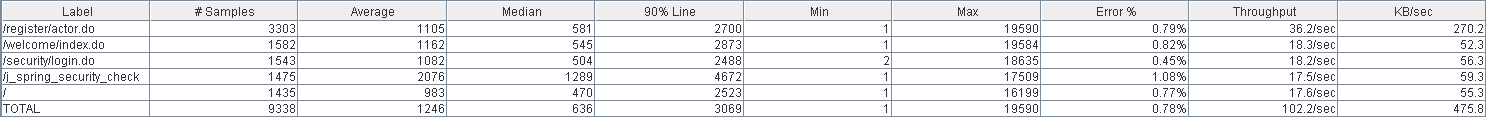
El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 3693ms = 3,693s.

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 3:**

* 250 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 100 y 150 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 200 y 250 usuarios simultáneos.

## List positions and display company (R.7.2)

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

### Casos de uso

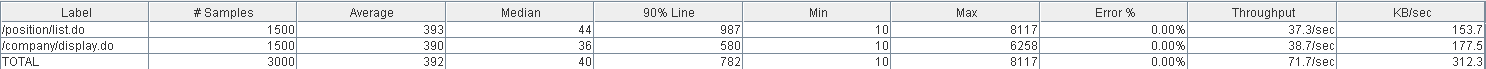
En este caso tenemos un caso de uso con los siguientes pasos:

1. Listar las Positions
2. Ver la Company

### Pruebas realizadas

**Prueba 1:**

* 150 usuarios
* Loop de 10 iteraciones



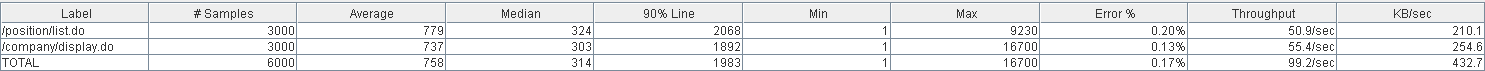
No hay errores HTTP y el tiempo medio de espera total es de 1567ms = 1,567s.

El mayor tiempo de espera se da al realizarse la operación de listar los puestos siendo de 987ms = 0,987s.

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

**Prueba 2:**

* 300 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 3960ms = 3,96s.

El mayor tiempo de espera se da al realizarse la operación de listar los puestos siendo de 2068ms = 2,068s.

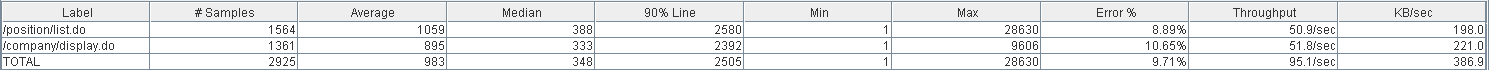
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

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



**Prueba 3:**

* 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 300 y 350 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 300 y 350 usuarios simultáneos.

## List companies and display positions (R.7.3)

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

### Casos de uso

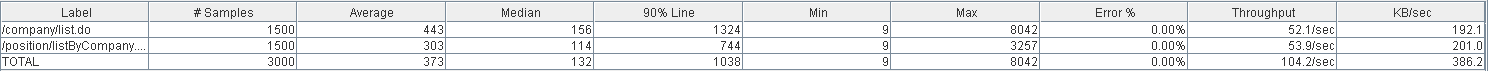
En este caso tenemos un caso de uso con los siguientes pasos:

1. Listar las Companies
2. Ver sus Positions

### Pruebas realizadas

**Prueba 1:**

* 150 usuarios
* Loop de 10 iteraciones



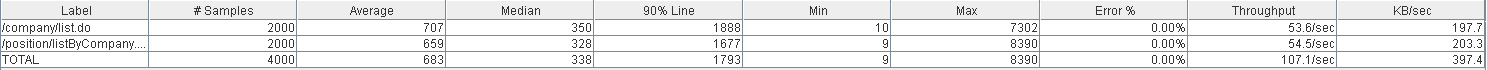
No hay errores HTTP y el tiempo medio de espera total es de 2068ms = 2,068s.

El mayor tiempo de espera se da al realizarse la operación de listar las empresas siendo de 1324ms = 1,324s.

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

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones



No hay errores HTTP y el tiempo medio de espera total es de 3565ms = 3,565s.

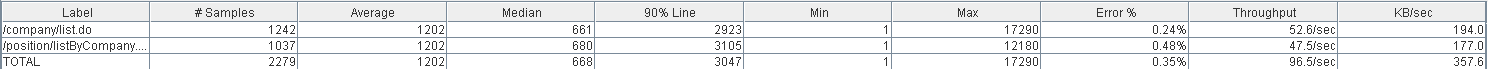
El mayor tiempo de espera se da al realizarse la operación de listar los puestos siendo de 1888ms = 1,888s.

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

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

**Prueba 3:**

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

## Edit personal data (R.8.2)

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

### Casos de uso

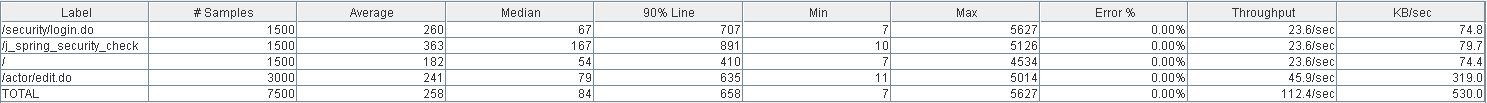
En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse
2. Ver datos personales
3. Editar datos personales

### Pruebas realizadas

**Prueba 1:**

* 150 usuarios
* Loop de 10 iteraciones



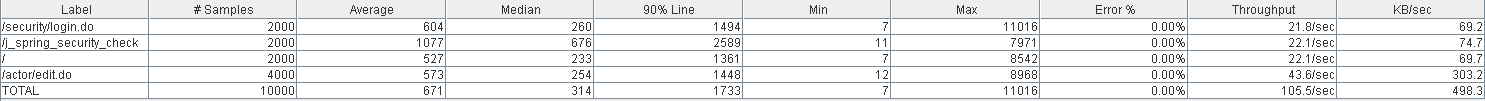
No hay errores HTTP y el tiempo medio de espera total es de 2643ms = 2,643s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 891ms = 0,891s.

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

**Prueba 2:**

* 200 usuarios
* Loop de 10 iteraciones

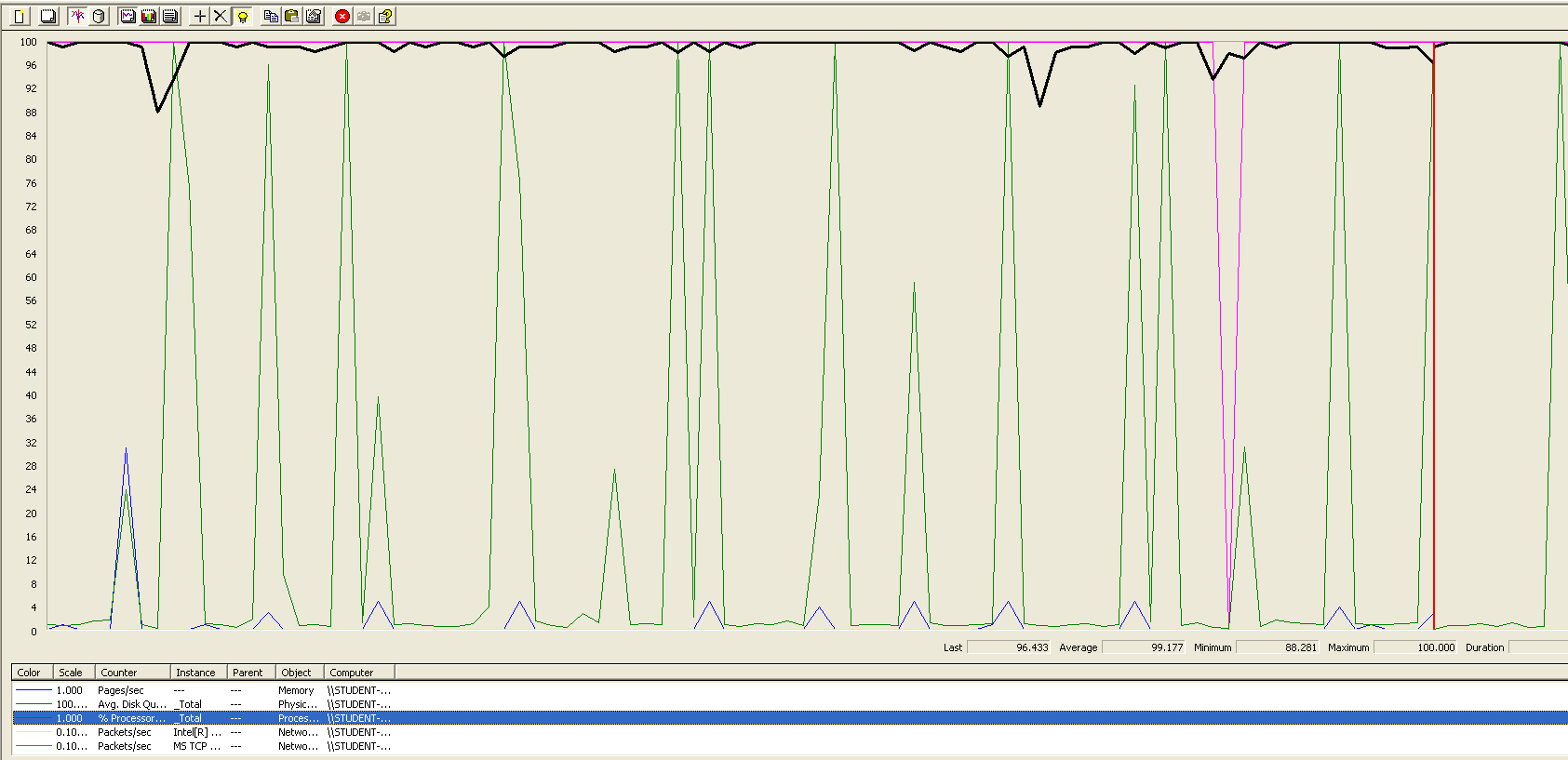


No hay errores HTTP y el tiempo medio de espera total es de 6892ms = 6,892s.

El mayor tiempo de espera se da al realizarse la operación de listar los puestos siendo de 2589ms = 2,589s.

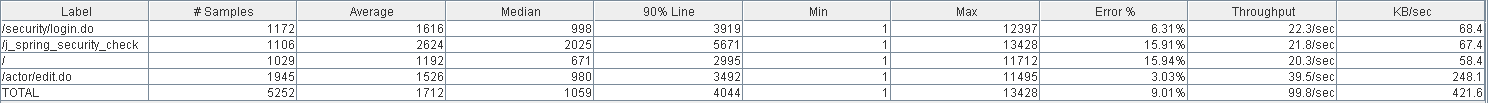
Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

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



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