



---

# TESTS DE RENDIMIENTO

---

Grupo 21



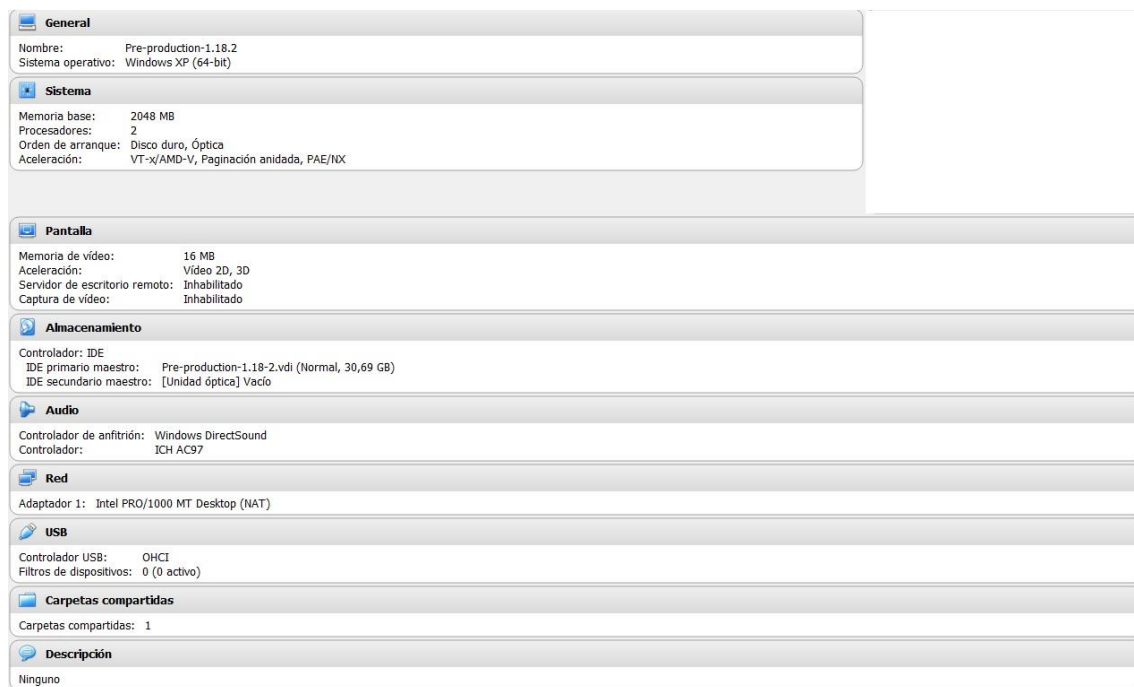
## Contenido

|        |   |    |
|--------|---|----|
| 1.     | Descripción del ordenador .....                   | 2  |
| 1.1.   | Descripción de máquina virtual.....               | 2  |
| 1.2.   | Descripción de máquina física.....                | 2  |
| 2.     | Casos de uso e informes.....                      | 3  |
| 2.1.   | Configuration (R.14) .....                        | 3  |
| 2.1.1. | Casos de uso .....                                | 3  |
| 2.1.2. | Pruebas realizadas.....                           | 3  |
| 2.1.3. | Conclusiones.....                                 | 5  |
| 2.2.   | Register (R.7.1) .....                            | 5  |
| 2.2.1. | Casos de uso .....                                | 5  |
| 2.2.2. | Pruebas realizadas.....                           | 5  |
| 2.2.3. | Conclusiones.....                                 | 7  |
| 2.3.   | List positions and display company (R.7.2) .....  | 7  |
| 2.3.1. | Casos de uso .....                                | 7  |
| 2.3.2. | Pruebas realizadas.....                           | 7  |
| 2.3.3. | Conclusiones.....                                 | 9  |
| 2.4.   | List companies and display positions (R.7.3)..... | 9  |
| 2.4.1. | Casos de uso .....                                | 9  |
| 2.4.2. | Pruebas realizadas.....                           | 9  |
| 2.4.3. | Conclusiones.....                                 | 10 |
| 2.5.   | Edit personal data (R.8.2).....                   | 10 |
| 2.5.1. | Casos de uso .....                                | 10 |
| 2.5.2. | Pruebas realizadas.....                           | 11 |
| 2.5.3. | Conclusiones.....                                 | 12 |
| 2.6.   | Dashboard (R.11.2 & 18) .....                     | 12 |
| 2.6.1. | Casos de uso .....                                | 12 |
| 2.6.2. | Pruebas realizadas.....                           | 12 |
| 2.6.3. | Conclusiones.....                                 | 14 |

## 1. Descripción del ordenador

Máquina virtual ejecutada con Oracle VM VirtualBox.

### 1.1. Descripción de máquina virtual



### 1.2. Descripción de máquina física

Edición de Windows

Windows 10 Education

© 2018 Microsoft Corporation. Todos los derechos reservados.

Sistema

Procesador: Intel(R) Core(TM) i7-4700MQ CPU @ 2.40GHz 2.40 GHz  
Memoria instalada (RAM): 16,0 GB  
Tipo de sistema: Sistema operativo de 64 bits, procesador x64  
Lápiz y entrada táctil: La entrada táctil o manuscrita no está disponible para esta pantalla

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.

## 2. Casos de uso e informes

### 2.1. Configuration (R.14)

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

#### 2.1.1. 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

#### 2.1.2. Pruebas realizadas

##### Prueba 1:

- 150 usuarios
- Loop de 10 iteraciones

| Label                        | # Samples | Average | Median | 90% Line | Min | Max   | Error % | Throughput | KB/sec |
|------------------------------|-----------|---------|--------|----------|-----|-------|---------|------------|--------|
| /security/login.do           | 1500      | 616     | 105    | 1312     | 8   | 24791 | 0.00%   | 9.6/sec    | 30.3   |
| /j_spring_security_check     | 1500      | 879     | 212    | 1717     | 12  | 32537 | 0.00%   | 9.6/sec    | 37.1   |
| /                            | 1500      | 371     | 87     | 632      | 7   | 23222 | 0.00%   | 9.6/sec    | 34.9   |
| /configuration/administra... | 3000      | 435     | 102    | 1016     | 9   | 18635 | 0.00%   | 18.0/sec   | 74.7   |
| /configuration/administra... | 3000      | 604     | 153    | 1581     | 10  | 19384 | 0.00%   | 19.0/sec   | 87.7   |
| /configuration/administra... | 1500      | 394     | 92     | 942      | 9   | 10173 | 0.00%   | 9.6/sec    | 54.4   |
| TOTAL                        | 12000     | 542     | 120    | 1232     | 7   | 32537 | 0.00%   | 72.8/sec   | 315.4  |

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

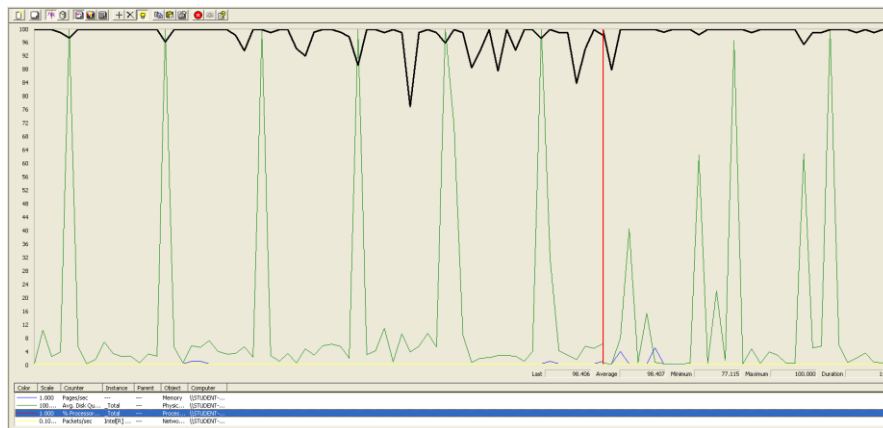
| Label                        | # Samples | Average | Median | 90% Line | Min | Max   | Error % | Throughput | kB/sec |
|------------------------------|-----------|---------|--------|----------|-----|-------|---------|------------|--------|
| /security/login.do           | 2000      | 468     | 167    | 1176     | 7   | 8671  | 0.00%   | 13.8/sec   | 43.9   |
| /j_spring_security_check     | 2000      | 907     | 455    | 2207     | 11  | 12058 | 0.00%   | 13.9/sec   | 53.9   |
| /                            | 2000      | 421     | 158    | 958      | 7   | 10941 | 0.00%   | 13.9/sec   | 50.7   |
| /configuration/administra... | 4000      | 441     | 162    | 1113     | 8   | 9058  | 0.00%   | 26.9/sec   | 108.2  |
| /configuration/administra... | 4000      | 631     | 267    | 1607     | 9   | 9425  | 0.00%   | 27.4/sec   | 140.7  |
| /configuration/administra... | 2000      | 390     | 146    | 1039     | 9   | 7359  | 0.00%   | 14.0/sec   | 79.0   |
| TOTAL                        | 16000     | 541     | 209    | 1374     | 7   | 12058 | 0.00%   | 105.3/sec  | 456.3  |

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

| Label                        | # Samples | Average | Median | 90% Line | Min | Max   | Error % | Throughput | kB/sec |
|------------------------------|-----------|---------|--------|----------|-----|-------|---------|------------|--------|
| /security/login.do           | 2500      | 826     | 355    | 2192     | 7   | 12659 | 0.00%   | 14.7/sec   | 46.6   |
| /j_spring_security_check     | 2500      | 1400    | 934    | 3255     | 12  | 13448 | 0.00%   | 14.8/sec   | 57.6   |
| /                            | 2500      | 679     | 341    | 1623     | 8   | 11125 | 0.00%   | 14.8/sec   | 54.1   |
| /configuration/administra... | 5000      | 632     | 308    | 1537     | 8   | 20234 | 0.00%   | 28.8/sec   | 116.0  |
| /configuration/administra... | 5000      | 908     | 498    | 2239     | 9   | 12635 | 0.00%   | 29.4/sec   | 151.0  |
| /configuration/administra... | 2500      | 631     | 285    | 1542     | 9   | 11769 | 0.00%   | 15.0/sec   | 85.2   |
| TOTAL                        | 20000     | 827     | 422    | 2070     | 7   | 20234 | 0.00%   | 112.4/sec  | 487.9  |

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

| Label                        | # Samples | Average | Median | 90% Line | Min | Max   | Error % | Throughput | KB/sec |
|------------------------------|-----------|---------|--------|----------|-----|-------|---------|------------|--------|
| /security/login.do           | 305       | 1464    | 862    | 3190     | 14  | 14320 | 0.33%   | 19.2/sec   | 60.7   |
| /spring_security/check       | 234       | 1443    | 864    | 3432     | 14  | 8745  | 0.43%   | 17.3/sec   | 66.8   |
| /                            | 163       | 425     | 108    | 879      | 10  | 6798  | 0.00%   | 12.1/sec   | 44.0   |
| /configuration/administra... | 239       | 875     | 338    | 2271     | 11  | 6246  | 0.64%   | 20.3/sec   | 81.4   |
| /configuration/administra... | 269       | 1530    | 581    | 4857     | 11  | 7752  | 0.00%   | 26.1/sec   | 136.5  |
| /configuration/administra... | 43        | 1341    | 1190   | 1934     | 29  | 3207  | 2.33%   | 8.8/sec    | 48.8   |
| TOTAL                        | 1263      | 1222    | 573    | 3126     | 10  | 14320 | 0.40%   | 78.8/sec   | 318.8  |

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).

### 2.1.3. 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.

## 2.2. Register (R.7.1)

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

### 2.2.1. Casos de uso

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

1. Registrarse
2. Loguearse

### 2.2.2. Pruebas realizadas

#### Prueba 1:

- 150 usuarios
- Loop de 10 iteraciones

| Label                  | # Samples | Average | Median | 90% Line | Min | Max   | Error % | Throughput | KB/sec |
|------------------------|-----------|---------|--------|----------|-----|-------|---------|------------|--------|
| /registerfactor.do     | 3000      | 1169    | 306    | 2763     | 11  | 30391 | 0.00%   | 23.7/sec   | 177.6  |
| /welcomeindex.do       | 1500      | 746     | 223    | 1910     | 8   | 20196 | 0.00%   | 12.7/sec   | 36.3   |
| /securitylogin.do      | 1500      | 640     | 190    | 1744     | 7   | 14893 | 0.00%   | 12.9/sec   | 39.5   |
| /spring_security_check | 1500      | 1112    | 474    | 2780     | 13  | 14104 | 0.00%   | 12.9/sec   | 43.4   |
| /                      | 1500      | 550     | 172    | 1430     | 8   | 9554  | 0.00%   | 13.0/sec   | 40.7   |
| TOTAL                  | 9000      | 898     | 270    | 2213     | 7   | 30391 | 0.00%   | 69.1/sec   | 316.3  |

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

| Label                  | # Samples | Average | Median | 90% Line | Min | Max   | Error % | Throughput | KB/sec |
|------------------------|-----------|---------|--------|----------|-----|-------|---------|------------|--------|
| /registerfactor.do     | 4000      | 766     | 289    | 1947     | 10  | 16555 | 0.00%   | 29.2/sec   | 219.5  |
| /welcomeindex.do       | 2000      | 704     | 248    | 1785     | 7   | 21748 | 0.00%   | 14.8/sec   | 42.4   |
| /securitylogin.do      | 2000      | 737     | 259    | 1961     | 7   | 14264 | 0.00%   | 14.9/sec   | 45.7   |
| /spring_security_check | 2000      | 1469    | 689    | 3693     | 5   | 37624 | 0.00%   | 14.9/sec   | 50.0   |
| /                      | 2000      | 646     | 216    | 1683     | 7   | 15859 | 0.00%   | 14.8/sec   | 46.6   |
| TOTAL                  | 12000     | 848     | 306    | 2152     | 5   | 37624 | 0.00%   | 85.5/sec   | 391.8  |

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

| Label                  | # Samples | Average | Median | 90% Line | Min | Max   | Error % | Throughput | KB/sec |
|------------------------|-----------|---------|--------|----------|-----|-------|---------|------------|--------|
| /registerfactor.do     | 3303      | 1105    | 581    | 2700     | 1   | 19590 | 0.79%   | 36.2/sec   | 270.2  |
| /welcomeindex.do       | 1582      | 1162    | 545    | 2873     | 1   | 19584 | 0.82%   | 18.3/sec   | 52.3   |
| /securitylogin.do      | 1543      | 1082    | 504    | 2488     | 2   | 18635 | 0.45%   | 18.2/sec   | 56.3   |
| /spring_security_check | 1475      | 2076    | 1289   | 4672     | 1   | 17509 | 1.09%   | 17.5/sec   | 59.3   |
| /                      | 1435      | 983     | 470    | 2523     | 1   | 16199 | 0.77%   | 17.6/sec   | 55.3   |
| TOTAL                  | 9338      | 1246    | 636    | 3069     | 1   | 19590 | 0.76%   | 102.2/sec  | 475.8  |

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).

### 2.2.3. 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.

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

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

### 2.3.1. Casos de uso

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

1. Listar las Positions
2. Ver la Company

### 2.3.2. Pruebas realizadas

#### Prueba 1:

- 150 usuarios
- Loop de 10 iteraciones

| Label              | # Samples | Average | Median | 90% Line | Min | Max  | Error % | Throughput | kB/sec |
|--------------------|-----------|---------|--------|----------|-----|------|---------|------------|--------|
| /positionlist.do   | 1500      | 393     | 44     | 987      | 10  | 8117 | 0.00%   | 37.3/sec   | 153.7  |
| /companydisplay.do | 1500      | 390     | 36     | 580      | 10  | 6258 | 0.00%   | 38.7/sec   | 177.5  |
| TOTAL              | 3000      | 392     | 40     | 782      | 10  | 8117 | 0.00%   | 71.7/sec   | 312.3  |

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

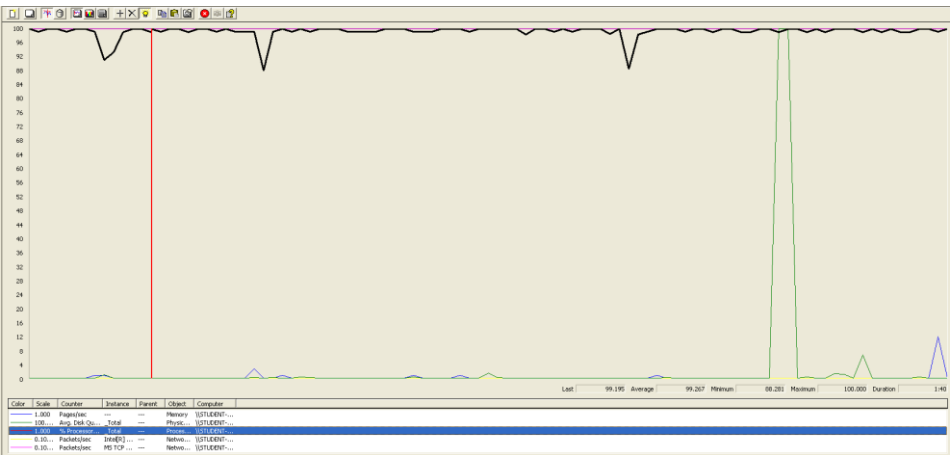
| Label              | # Samples | Average | Median | 90% Line | Min | Max   | Error % | Throughput | KB/sec |
|--------------------|-----------|---------|--------|----------|-----|-------|---------|------------|--------|
| /posi@onlist.do    | 3000      | 779     | 324    | 2068     | 1   | 9230  | 0.20%   | 60.9/sec   | 210.1  |
| /companydisplay.do | 3000      | 737     | 303    | 1992     | 1   | 16700 | 0.13%   | 55.4/sec   | 254.6  |
| TOTAL              | 6000      | 758     | 314    | 1993     | 1   | 16700 | 0.17%   | 99.2/sec   | 432.7  |

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

| Label              | # Samples | Average | Median | 90% Line | Min | Max   | Error % | Throughput | KB/sec |
|--------------------|-----------|---------|--------|----------|-----|-------|---------|------------|--------|
| /posi@onlist.do    | 1564      | 1059    | 388    | 2580     | 1   | 28630 | 8.89%   | 50.9/sec   | 198.0  |
| /companydisplay.do | 1361      | 895     | 333    | 2392     | 1   | 9608  | 10.65%  | 51.8/sec   | 221.0  |
| TOTAL              | 2925      | 983     | 348    | 2505     | 1   | 28630 | 9.71%   | 95.1/sec   | 386.9  |

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).

### 2.3.3. 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.

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

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

### 2.4.1. Casos de uso

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

1. Listar las Companies
2. Ver sus Positions

### 2.4.2. Pruebas realizadas

#### Prueba 1:

- 150 usuarios
- Loop de 10 iteraciones

| Label                     | # Samples | Average | Median | 90% Line | Min | Max  | Error % | Throughput | kB/sec |
|---------------------------|-----------|---------|--------|----------|-----|------|---------|------------|--------|
| /companylist.do           | 1500      | 443     | 156    | 1324     | 9   | 8042 | 0.00%   | 52.1/sec   | 192.1  |
| /positionlistByCompany... | 1500      | 303     | 114    | 744      | 9   | 3257 | 0.00%   | 53.9/sec   | 201.0  |
| TOTAL                     | 3000      | 373     | 132    | 1039     | 9   | 8042 | 0.00%   | 104.2/sec  | 396.2  |

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

| Label                     | # Samples | Average | Median | 90% Line | Min | Max  | Error % | Throughput | kB/sec |
|---------------------------|-----------|---------|--------|----------|-----|------|---------|------------|--------|
| /companylist.do           | 2000      | 707     | 350    | 1888     | 10  | 7302 | 0.00%   | 53.6/sec   | 197.7  |
| /positionlistByCompany... | 2000      | 669     | 328    | 1677     | 9   | 6390 | 0.00%   | 54.5/sec   | 203.3  |
| TOTAL                     | 4000      | 683     | 338    | 1793     | 9   | 6390 | 0.00%   | 107.1/sec  | 397.4  |

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

| Label                     | # Samples | Average | Median | 90% Line | Min | Max   | Error % | Throughput | KB/sec |
|---------------------------|-----------|---------|--------|----------|-----|-------|---------|------------|--------|
| /companylist.do           | 1242      | 1202    | 661    | 2923     | 1   | 17290 | 0.24%   | 52.6/sec   | 194.0  |
| /positionlistByCompany... | 1037      | 1202    | 680    | 3105     | 1   | 12180 | 0.48%   | 47.5/sec   | 177.0  |
| TOTAL                     | 2279      | 1202    | 668    | 3047     | 1   | 17290 | 0.35%   | 96.5/sec   | 357.6  |

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).

### 2.4.3. 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.

## 2.5. Edit personal data (R.8.2)

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

### 2.5.1. 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

## 2.5.2. Pruebas realizadas

### Prueba 1:

- 150 usuarios
- Loop de 10 iteraciones

| Label                  | # Samples | Average | Median | 90% Line | Min | Max  | Error % | Throughput | kB/sec |
|------------------------|-----------|---------|--------|----------|-----|------|---------|------------|--------|
| /securitylogin.do      | 1500      | 260     | 67     | 707      | 7   | 5627 | 0.00%   | 23.6/sec   | 74.8   |
| /spring_security_check | 1500      | 363     | 167    | 891      | 10  | 5126 | 0.00%   | 23.6/sec   | 79.7   |
| /                      | 1500      | 192     | 54     | 410      | 7   | 4534 | 0.00%   | 23.6/sec   | 74.4   |
| /actonfedit.do         | 3000      | 241     | 79     | 635      | 11  | 5014 | 0.00%   | 45.9/sec   | 319.0  |
| TOTAL                  | 7500      | 258     | 84     | 658      | 7   | 5627 | 0.00%   | 112.4/sec  | 530.0  |

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

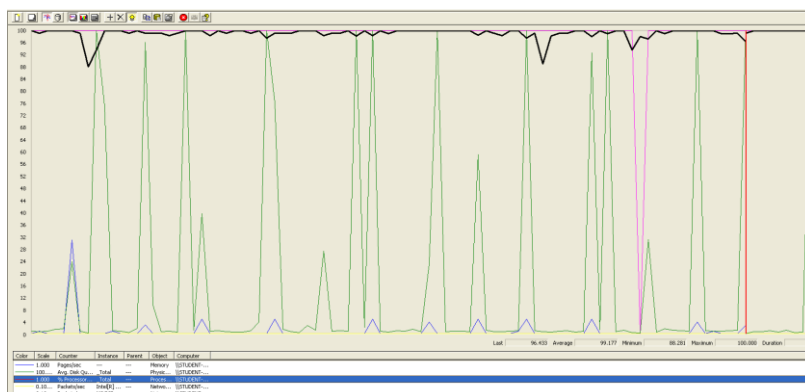
| Label                  | # Samples | Average | Median | 90% Line | Min | Max   | Error % | Throughput | kB/sec |
|------------------------|-----------|---------|--------|----------|-----|-------|---------|------------|--------|
| /securitylogin.do      | 2000      | 604     | 260    | 1434     | 7   | 11016 | 0.00%   | 21.8/sec   | 69.2   |
| /spring_security_check | 2000      | 1077    | 676    | 2599     | 11  | 7971  | 0.00%   | 22.1/sec   | 74.7   |
| /                      | 2000      | 527     | 233    | 1361     | 7   | 8542  | 0.00%   | 22.1/sec   | 69.7   |
| /actonfedit.do         | 4000      | 573     | 254    | 1448     | 12  | 8968  | 0.00%   | 43.6/sec   | 303.2  |
| TOTAL                  | 10000     | 671     | 314    | 1733     | 7   | 11016 | 0.00%   | 105.5/sec  | 496.3  |

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

| Label                    | # Samples | Average | Median | 90% Line | Min | Max   | Error % | Throughput | KB/sec |
|--------------------------|-----------|---------|--------|----------|-----|-------|---------|------------|--------|
| /securitylogin.do        | 1172      | 1816    | 998    | 3919     | 1   | 12387 | 6.31%   | 22.3/sec   | 88.4   |
| /j_spring_security_check | 1106      | 2624    | 2025   | 5671     | 1   | 13428 | 15.91%  | 21.9/sec   | 67.4   |
| /                        | 1029      | 1192    | 671    | 2995     | 1   | 11712 | 15.94%  | 20.3/sec   | 58.4   |
| /factoredit.do           | 1945      | 1528    | 980    | 3492     | 1   | 11495 | 3.03%   | 39.5/sec   | 248.1  |
| TOTAL                    | 5252      | 1712    | 1059   | 4044     | 1   | 13428 | 9.01%   | 99.9/sec   | 421.6  |

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).

### 2.5.3. 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.

## 2.6. Dashboard (R.11.2 & 18)

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

### 2.6.1. Casos de uso

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

1. Loguearse
2. Ver Dashboard

### 2.6.2. Pruebas realizadas

#### Prueba 1:

- 150 usuarios
- Loop de 10 iteraciones

| Label                      | # Samples | Average | Median | 90% Line | Min | Max   | Error % | Throughput | KB/sec |
|----------------------------|-----------|---------|--------|----------|-----|-------|---------|------------|--------|
| /securitylogin.do          | 1500      | 1006    | 555    | 2530     | 9   | 11436 | 0.00%   | 16.4/sec   | 51.9   |
| /j_spring_security_check   | 1500      | 2020    | 1472   | 4280     | 18  | 21301 | 0.00%   | 17.0/sec   | 65.9   |
| /                          | 1500      | 1006    | 495    | 2517     | 10  | 19239 | 0.00%   | 17.0/sec   | 62.1   |
| /dashboard/administrato... | 1500      | 983     | 543    | 2344     | 29  | 10897 | 0.00%   | 17.0/sec   | 107.4  |
| TOTAL                      | 6000      | 1254    | 707    | 3124     | 9   | 21301 | 0.00%   | 64.3/sec   | 273.5  |

No hay errores HTTP y el tiempo medio de espera total es de 11671ms = 11,671s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 4280ms = 4,280s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable debido a que la dashboard realiza muchas peticiones a la base de datos.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

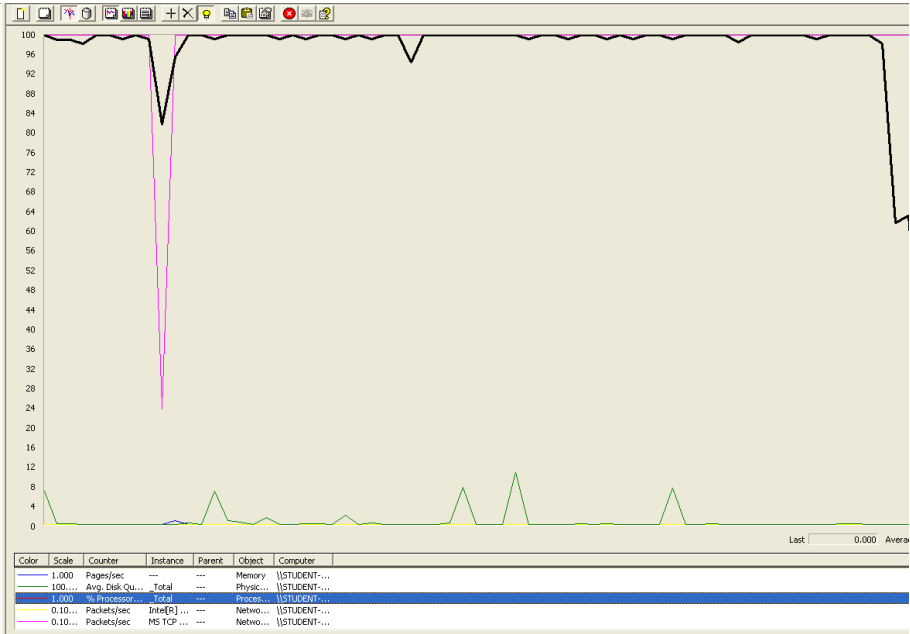
| Label                      | # Samples | Average | Median | 90% Line | Min | Max   | Error % | Throughput | kB/sec |
|----------------------------|-----------|---------|--------|----------|-----|-------|---------|------------|--------|
| /security/login.do         | 2000      | 1747    | 947    | 4222     | 9   | 24263 | 0.00%   | 15.1/sec   | 47.9   |
| /j_spring_security_check   | 2000      | 3391    | 2363   | 7763     | 15  | 28950 | 0.00%   | 15.4/sec   | 59.9   |
| /                          | 2000      | 1658    | 878    | 4206     | 10  | 18560 | 0.00%   | 15.4/sec   | 58.3   |
| /dashboard/administrato... | 2000      | 1686    | 917    | 4262     | 29  | 22029 | 0.00%   | 15.6/sec   | 98.2   |
| TOTAL                      | 8000      | 2121    | 1182   | 5197     | 9   | 28950 | 0.00%   | 59.8/sec   | 254.8  |

No hay errores HTTP y el tiempo medio de espera total es de 20453ms = 20,453s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 7763ms = 7,7763s.

Por lo tanto, la ejecución ha sido satisfactoria pero el tiempo ha sido 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

| Label                      | # Samples | Average | Median | 90% Line | Min | Max   | Error % | Throughput | kB/sec |
|----------------------------|-----------|---------|--------|----------|-----|-------|---------|------------|--------|
| /security/login.do         | 2500      | 2573    | 1621   | 5732     | 9   | 28812 | 0.00%   | 14.2/sec   | 45.2   |
| /j_spring_security_check   | 2500      | 4743    | 3710   | 9044     | 16  | 52308 | 0.00%   | 14.5/sec   | 56.4   |
| /                          | 2500      | 2439    | 1503   | 5351     | 10  | 31607 | 0.00%   | 14.5/sec   | 53.0   |
| /dashboard/administrato... | 2500      | 2441    | 1600   | 5123     | 28  | 35332 | 0.00%   | 14.4/sec   | 90.9   |
| TOTAL                      | 10000     | 3049    | 2033   | 6654     | 9   | 52308 | 0.00%   | 56.0/sec   | 238.5  |

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

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 9044ms = 9,044s.

Por lo tanto, la ejecución ha sido satisfactoria pero el tiempo ha sido muy elevado.

El cuello de botella se sigue produciendo en la CPU como en la anterior prueba.

#### Prueba 4:

- 300 usuarios
- Loop de 10 iteraciones

| Label                  | # Samples | Average | Median | 90% Line | Min | Max  | Error % | Throughput | kB/sec |
|------------------------|-----------|---------|--------|----------|-----|------|---------|------------|--------|
| /securitylogin.do      | 254       | 400     | 244    | 825      | 57  | 8238 | 0.00%   | 22.3/sec   | 70.7   |
| /spring_security_check | 7         | 32      | 2      | 3        | 1   | 217  | 100.00% | 31.4/sec   | 48.7   |
| TOTAL                  | 261       | 391     | 237    | 822      | 1   | 8238 | 2.68%   | 21.1/sec   | 66.0   |

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).

#### 2.6.3. 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 250 y 300 usuarios simultáneos.