



---

# TESTS DE RENDIMIENTO

---

Grupo 21












## Contenido

1.	Descripción del ordenador .....	2
1.1.	Descripción de máquina virtual.....	2
1.2.	Descripción de máquina física.....	3
2.	Casos de uso e informes.....	3
2.1.1.	Casos de uso .....	3
2.1.2.	Pruebas realizadas.....	3
2.1.3.	Conclusiones.....	6
2.1.4.	Casos de uso .....	6
2.1.5.	Pruebas realizadas.....	6
2.1.6.	Conclusiones.....	9
2.1.7.	Casos de uso .....	9
2.1.8.	Pruebas realizadas.....	9
2.1.9.	Conclusiones.....	12
2.1.10.	Casos de uso .....	12
2.1.11.	Pruebas realizadas.....	12
2.1.12.	Conclusiones.....	15
2.1.13.	Casos de uso .....	15
2.1.14.	Pruebas realizadas.....	15
2.1.15.	Conclusiones.....	17
2.1.16.	Casos de uso .....	18
2.1.17.	Pruebas realizadas.....	18
2.1.18.	Conclusiones.....	20
2.1.19.	Casos de uso .....	20
2.1.20.	Pruebas realizadas.....	21
2.1.21.	Conclusiones.....	23

# 1. Descripción del ordenador

Máquina virtual ejecutada con Oracle VM VirtualBox.

## 1.1. Descripción de máquina virtual

 <b>General</b>	
Nombre:	Pre-production-1.18.2
Sistema operativo:	Windows XP (64-bit)
Ubicación de archivo de preferencias:	/Users/elenamolina/Downloads/Pre-production-1.18.2
 <b>Sistema</b>	
Memoria base:	2048 MB
Procesadores:	2
Orden de arranque:	Disco duro, Óptica
Aceleración:	VT-x/AMD-V, Paginación anidada, PAE/NX
 <b>Pantalla</b>	
Memoria de vídeo:	16 MB
Controlador gráfico:	VBoxVGA
Aceleración:	Vídeo 2D, 3D
Servidor de escritorio remoto:	Inhabilitado
Grabando:	Inhabilitado
 <b>Almacenamiento</b>	
Controlador:	IDE
IDE primario maestro:	Pre-production-1.18-2.vdi (Normal, 30,69 GB)
IDE secundario maestro:	[Unidad óptica] Vacío
 <b>Audio</b>	
Controlador de anfitrión:	CoreAudio
Controlador:	ICH AC97
 <b>Red</b>	
Adaptador 1:	Intel PRO/1000 MT Desktop (NAT)
 <b>USB</b>	
Controlador USB:	OHCI
Filtros de dispositivos:	0 (0 activo)
 <b>Carpetas compartidas</b>	
Carpetas compartidas:	1
 <b>Descripción</b>	
	Ninguno

## 1.2. Descripción de máquina física

### MacBook Pro

#### Información del hardware:

Nombre del modelo:	MacBook Pro
Identificador del modelo:	MacBookPro15,2
Nombre del procesador:	Intel Core i5
Velocidad del procesador:	2,3 GHz
Cantidad de procesadores:	1
Cantidad total de núcleos:	4
Caché de nivel 2 (por núcleo):	256 KB
Caché de nivel 3:	6 MB
Memoria:	8 GB
Versión de la ROM de arranque:	220.250.366.0.0 (iBridge: 16.16.4507.0.0,0)
Número de serie (sistema):	C02XX0YXJHCD
UUID de hardware:	75267765-05EE-5CCD-8B82-E26B73F0FD84

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

## 2. 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 *AssignPositionAuditorCasePlan.jmx*.

### 2.1.1. 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”

### 2.1.2. Pruebas realizadas

#### Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/lo...	1000	10	7	22	4	93	0.00%	12.4/sec	41.5
/j_spring_s...	1000	41	23	78	6	817	0.00%	12.4/sec	41.4
/	1000	27	16	53	6	549	0.00%	12.4/sec	38.7
/audit/audit...	2000	19	15	30	9	532	0.00%	24.0/sec	120.0
/audit/audit...	1000	32	26	48	15	1657	0.00%	12.4/sec	64.1
TOTAL	6000	25	17	44	4	1657	0.00%	69.8/sec	290.1

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

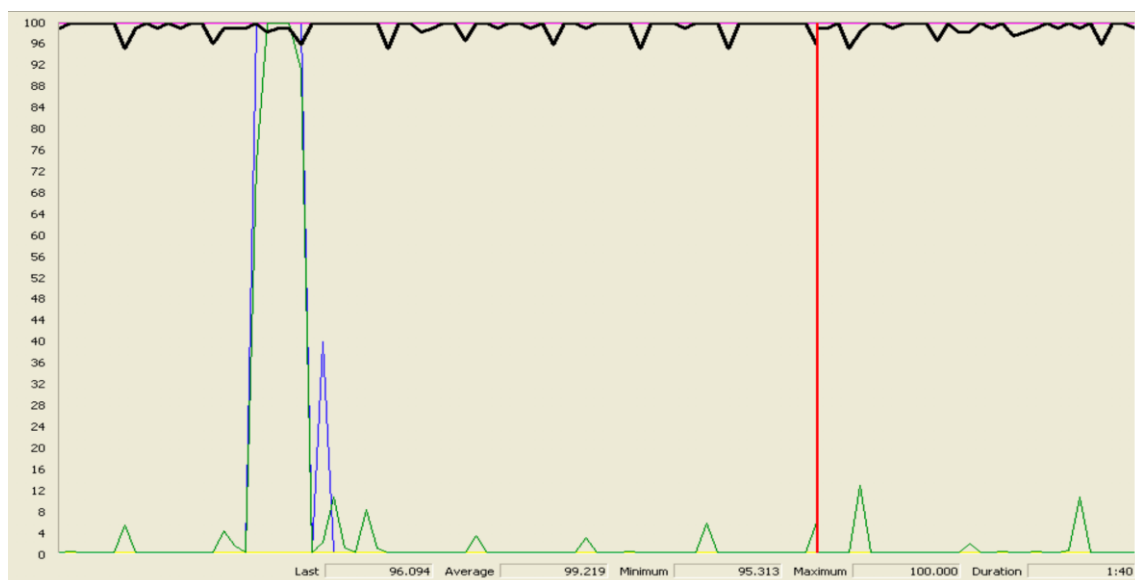
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/lo...	2000	25	10	38	4	3343	0.00%	23.2/sec	77.5
/j_spring_s...	2000	90	33	203	10	2648	0.00%	23.1/sec	77.3
/	2000	50	21	96	6	1945	0.00%	23.1/sec	72.1
/audit/audit...	4000	47	24	81	7	2821	0.00%	44.2/sec	221.0
/audit/audit...	2000	98	43	205	13	2391	0.00%	22.7/sec	118.0
TOTAL	12000	59	25	114	4	3343	0.00%	128.6/sec	536.3

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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/lo...	2500	124	15	266	4	4381	0.00%	29.4/sec	98.6
/j_spring_s...	2500	206	55	550	10	4517	0.00%	29.3/sec	98.2
/	2500	104	28	236	6	3738	0.00%	29.3/sec	91.4
/audit/audit...	5000	97	30	206	8	4868	0.00%	57.0/sec	285.4
/audit/audit...	2500	181	54	435	12	4791	0.00%	29.6/sec	154.0
TOTAL	15000	135	35	312	4	4868	0.00%	164.6/sec	686.8

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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/lo...	3000	329	135	849	5	8862	0.00%	25.6/sec	86.0
/j_spring_s...	3000	650	398	1534	15	6743	0.00%	25.6/sec	86.0
/	3000	319	138	858	8	5817	0.00%	25.6/sec	80.0
/audit/audit...	6000	308	147	808	9	7144	0.00%	49.8/sec	249.5
/audit/audit...	3000	645	402	1529	17	6040	0.00%	25.6/sec	133.5
TOTAL	18000	427	209	1099	5	8862	0.00%	147.4/sec	615.7

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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/l...	424	923	303	2678	1	10883	1.42%	28.6/sec	94.0
/j_spring_...	317	1450	290	4301	1	8722	3.47%	22.8/sec	81.5
/	211	556	43	2016	8	6216	0.00%	18.1/sec	61.9
/audit/aud...	271	755	487	1933	1	4310	1.11%	21.7/sec	108.0
/audit/aud...	155	2263	1512	5243	51	9073	1.94%	16.1/sec	83.0
TOTAL	1378	1106	362	3582	1	10883	1.67%	89.5/sec	350.1

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 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*.

### 2.1.4. Casos de uso

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

4. Loguearse
5. Ir a “Mis Audits”
6. Crear un audit
7. Editar ese audit
8. Mostrar ese audit
9. Eliminar dicho audit

### 2.1.5. Pruebas realizadas

#### Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/loge...	1000	35	10	59	5	611	0.00%	9.5/sec	31.6
/j_spring_se...	1000	50	26	93	12	946	0.00%	9.5/sec	31.5
/	1000	33	19	57	7	845	0.00%	9.5/sec	29.5
/audit/auditorf...	1000	24	16	39	8	487	0.00%	9.5/sec	33.0
/audit/auditorf...	1000	19	16	30	8	123	0.00%	9.5/sec	41.9
/audit/auditorf...	1000	20	18	34	10	74	0.00%	9.5/sec	38.8
/audit/auditorf...	1000	34	28	53	15	198	0.00%	9.6/sec	35.7
/audit/auditorf...	1000	37	29	62	13	429	0.00%	9.6/sec	45.0
TOTAL	8000	31	21	52	5	946	0.00%	70.6/sec	265.9

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

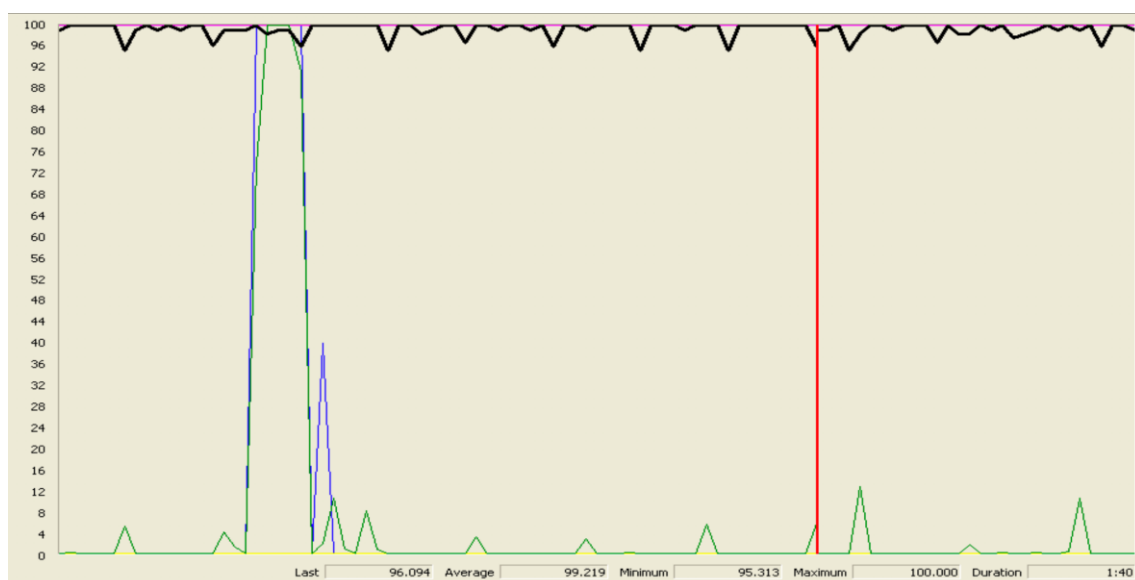
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/loge...	2000	263	73	668	5	5043	0.00%	15.2/sec	50.7
/j_spring_se...	2000	490	253	1213	11	6400	0.00%	15.1/sec	50.6
/	2000	245	98	634	8	5434	0.00%	15.0/sec	46.8
/audit/auditorf...	2000	233	81	605	8	4956	0.00%	15.0/sec	52.1
/audit/auditorf...	2000	211	73	547	9	3898	0.00%	14.9/sec	65.9
/audit/auditorf...	2000	204	73	519	9	4260	0.00%	14.8/sec	60.3
/audit/auditorf...	2000	411	169	1028	12	9207	0.00%	14.8/sec	55.5
/audit/auditorf...	2000	237	96	601	13	5105	0.00%	14.9/sec	69.7
TOTAL	16000	287	104	749	5	9207	0.00%	110.8/sec	418.8

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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login...	2500	363	137	900	5	6921	0.00%	16.5/sec	55.2
/j_spring_se...	2500	689	436	1615	12	5587	0.00%	16.4/sec	55.2
/f	2500	349	173	892	9	4114	0.00%	16.4/sec	51.3
/audit/auditorf...	2500	333	150	861	9	6412	0.00%	16.4/sec	57.4
/audit/auditorf...	2500	304	135	731	8	6527	0.00%	16.5/sec	72.7
/audit/auditorf...	2500	290	127	739	9	4492	0.00%	16.4/sec	67.1
/audit/auditorf...	2500	632	371	1505	13	5972	0.00%	16.4/sec	61.6
/audit/auditorf...	2500	328	157	790	13	5070	0.00%	16.5/sec	77.4
TOTAL	20000	411	197	1054	5	6921	0.00%	123.5/sec	467.1

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

/security/login...	1227	606	145	1746	0	11182	0.08%	17.7/sec	59.2
_spring_se...	1200	1041	451	2756	10	17941	0.00%	17.5/sec	58.7
	1176	554	168	1556	5	9575	0.09%	17.1/sec	53.4
udit/auditorf...	1159	549	131	1549	7	11979	0.00%	17.8/sec	62.1
udit/auditorf...	1129	499	91	1548	3	9450	0.09%	17.9/sec	78.8
udit/auditorf...	1085	496	126	1480	8	7363	0.00%	17.4/sec	71.1
udit/auditorf...	1029	1102	496	3058	14	8448	0.10%	17.5/sec	65.9
udit/auditorf...	960	610	197	1715	18	10413	0.00%	16.8/sec	78.7
TOTAL	8965	681	203	1969	0	17941	0.04%	126.9/sec	476.1

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 eliminar 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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login...	410	994	236	3134	1	9159	0.49%	28.5/sec	94.2
/j_spring_se...	350	1003	384	2866	1	4914	3.71%	25.0/sec	89.1
/f	249	247	110	641	11	3965	0.00%	19.1/sec	65.3
/audit/aud...	241	332	101	1092	1	3562	0.83%	21.2/sec	73.4
/audit/aud...	221	537	178	1866	1	3813	0.45%	21.1/sec	93.7
/audit/aud...	187	676	286	1855	0	4558	3.74%	20.6/sec	82.3
/audit/aud...	144	1978	1939	3675	0	4913	3.47%	17.5/sec	64.5
/audit/aud...	62	1393	1132	2685	0	3698	11.29%	10.5/sec	45.2
TOTAL	1864	813	252	2557	0	9159	1.98%	119.0/sec	434.9

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).

#### 2.1.6. 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*.

#### 2.1.7. 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”

#### 2.1.8. Pruebas realizadas

##### Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/lo...	6500	254	46	598	4	10466	0.00%	5.2/min	.4
/j_spring_s...	6500	477	171	1147	11	11019	0.00%	5.2/min	.4
/	6500	227	63	536	6	13610	0.00%	5.2/min	.4
/configurati...	6500	266	68	597	8	9990	0.00%	5.2/min	.4
TOTAL	26000	306	79	719	4	13610	0.00%	21.0/min	1.5

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

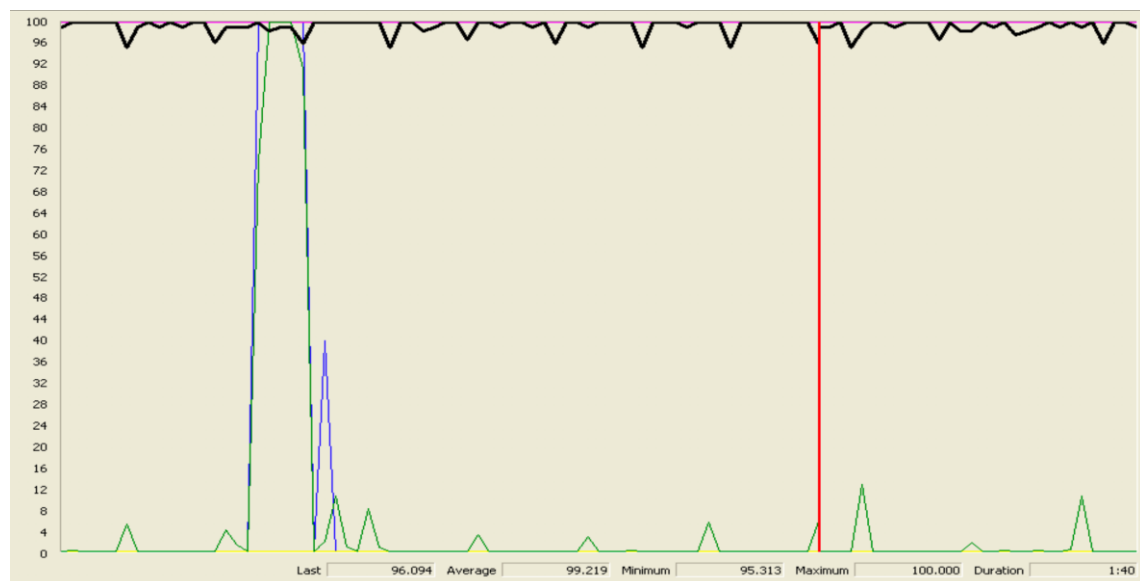
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/lo...	5500	298	73	707	4	10466	0.00%	32.4/sec	134.1
/j_spring_s...	5500	557	228	1335	11	11019	0.00%	32.4/sec	146.3
/	5500	265	85	614	7	13610	0.00%	32.4/sec	138.9
/configurati...	5500	311	95	702	8	9990	0.00%	32.4/sec	143.5
TOTAL	22000	358	111	852	4	13610	0.00%	127.4/sec	553.0

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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/lo...	3000	249	42	720	4	5510	0.00%	17.9/sec	73.8
/j_spring_s...	3000	566	225	1635	4	7143	0.00%	17.9/sec	80.6
/	3000	271	64	782	6	6168	0.00%	17.9/sec	76.5
/configurati...	3000	267	68	745	8	4644	0.00%	17.8/sec	78.6
TOTAL	12000	338	75	1006	4	7143	0.00%	70.0/sec	303.2

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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/lo...	9000	350	54	929	4	13986	0.00%	7.2/min	.5
/j_spring_s...	9000	658	189	1808	10	18173	0.00%	7.2/min	.5
/	9000	327	69	796	6	15228	0.00%	7.2/min	.5
/configurati...	9000	346	72	885	8	15043	0.00%	7.2/min	.5
TOTAL	36000	420	85	1109	4	18173	0.00%	29.0/min	2.1

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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/lo...	13051	498	122	1423	3	13986	0.07%	10.5/min	.7
/j_spring_s...	12946	944	357	2653	0	19593	0.14%	10.4/min	.8
/	12709	461	125	1237	6	15228	0.00%	10.2/min	.7
/configurati...	12684	470	134	1299	0	15043	0.03%	10.2/min	.7
TOTAL	51390	594	166	1711	0	19593	0.06%	41.2/min	3.0

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).

#### 2.1.9. 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*.

#### 2.1.10. Casos de uso

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

3. Loguearse
4. Acceder al menú y pulsar en Register a provider

#### 2.1.11. Pruebas realizadas

##### Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/l...	1000	17	9	35	5	330	0.00%	11.4/sec	46.9
/j_spring_...	1000	94	36	106	12	2063	0.00%	11.3/sec	51.0
/	1000	45	29	71	7	1932	0.00%	11.3/sec	48.5
/register/a...	2000	29	21	50	6	1212	0.00%	22.3/sec	193.6
/welcome/...	1000	20	14	37	7	87	0.00%	11.4/sec	48.7
TOTAL	6000	39	21	57	5	2063	0.00%	63.4/sec	365.2

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

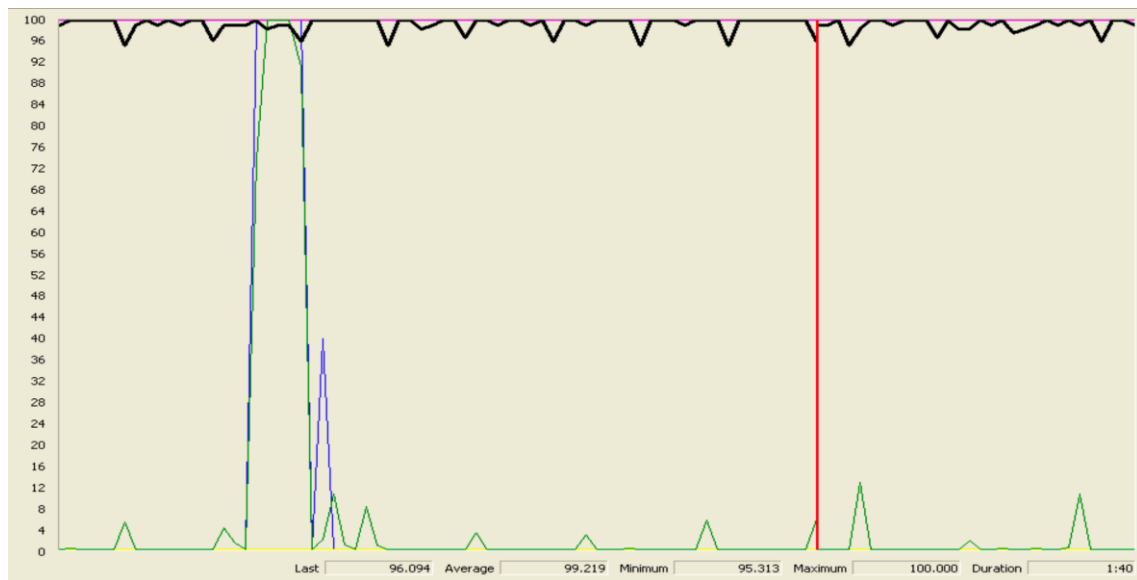
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/lo...	11000	154	26	418	4	7989	0.00%	8.0/sec	33.2
/j_spring_s...	11000	352	117	917	9	9043	0.00%	8.0/sec	34.0
/	11000	177	47	445	5	6670	0.00%	8.0/sec	32.1
/register/ad...	22000	172	47	426	5	9816	0.00%	16.1/sec	139.8
/welcome/i...	11000	151	37	395	6	8288	0.00%	8.0/sec	32.1
TOTAL	66000	197	48	512	4	9816	0.00%	48.1/sec	270.4

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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput
/security/logi...	8500	185	34	515	4	7989	0.00%	7.3/sec
/j_spring_se...	8500	413	152	1105	10	9043	0.00%	7.3/sec
/	8500	208	56	537	5	6670	0.00%	7.3/sec
/register/ad...	17000	202	55	510	5	9816	0.00%	14.6/sec
/welcome/in...	8500	176	43	472	6	8288	0.00%	7.3/sec

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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/lo...	3000	216	24	609	4	7989	0.00%	4.4/sec	18.1
/j_spring_s...	3000	465	81	1422	10	9043	0.00%	4.4/sec	18.5
/	3000	240	39	639	6	6670	0.00%	4.4/sec	17.5
/register/ad...	6000	234	41	595	5	9816	0.00%	8.7/sec	76.1
/welcome/i...	3000	210	34	569	7	8288	0.00%	4.4/sec	17.5
TOTAL	18000	267	39	738	4	9816	0.00%	26.1/sec	146.5

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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/l...	4185	584	59	1706	4	19803	0.79%	4.9/sec	20.1
/j_spring_...	4152	1126	221	3426	11	23836	0.77%	4.9/sec	21.9
/	4120	558	78	1548	7	17361	0.22%	4.8/sec	20.7
/register/a...	8215	565	72	1604	5	22316	0.15%	9.6/sec	83.7
/welcome/...	4099	545	63	1486	7	22482	0.15%	4.8/sec	20.6
TOTAL	24771	658	81	1881	4	23836	0.37%	28.8/sec	165.8

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).

### 2.1.12. 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*.

### 2.1.13. Casos de uso

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

5. Loguearse
6. Acceder al menú de “Administrador” y pulsar “Change Brand”
7. Pulsar Launch Process

### 2.1.14. Pruebas realizadas

#### Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/lo...	1000	9	7	16	4	88	0.00%	17.6/sec	72.5
/j_spring_s...	1000	25	19	42	10	215	0.00%	17.6/sec	79.2
/	1000	16	13	28	6	205	0.00%	17.6/sec	75.3
/configurati...	1000	7	7	10	4	36	0.00%	17.5/sec	72.9
/configurati...	1000	10	9	15	6	42	0.00%	17.4/sec	74.4
TOTAL	5000	13	10	25	4	215	0.00%	80.9/sec	345.0

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



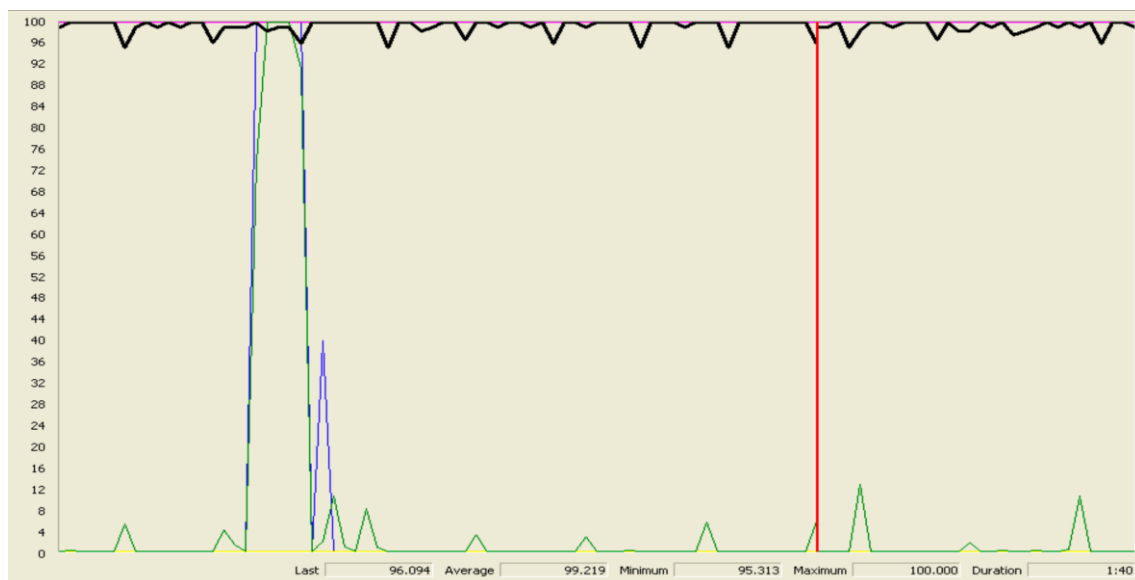
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/lo...	2000	19	8	29	4	1231	0.00%	34.2/sec	141.3
/j_spring_s...	2000	50	29	109	6	676	0.00%	34.1/sec	154.0
/	2000	29	19	52	5	670	0.00%	34.1/sec	146.2
/configurati...	2000	13	7	22	3	312	0.00%	34.1/sec	141.9
/configurati...	2000	10	9	16	5	71	0.00%	33.9/sec	145.1
TOTAL	10000	24	12	45	3	1231	0.00%	157.7/sec	674.1

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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/lo...	4500	251	33	753	4	4978	0.00%	18.4/sec	76.0
/j_spring_s...	4500	523	156	1503	6	7554	0.00%	18.4/sec	83.0
/	4500	272	56	788	5	5940	0.00%	18.4/sec	78.8
/configurati...	4500	241	25	744	3	7368	0.00%	18.4/sec	76.7
/configurati...	4500	251	27	779	5	6865	0.00%	18.4/sec	78.8
TOTAL	22500	307	45	937	3	7554	0.00%	90.3/sec	386.2

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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/l...	3000	1009	508	2412	5	24664	0.00%	22.5/sec	93.1
/j_spring_...	3000	1911	1217	4429	5	20918	0.00%	22.6/sec	102.5
/	3000	928	422	2271	8	15649	0.00%	22.7/sec	97.2
/configura...	3000	822	388	2064	4	12287	0.00%	22.7/sec	94.7
/configura...	3000	823	394	2076	6	15174	0.00%	22.7/sec	97.6
TOTAL	15000	1099	536	2775	4	24664	0.00%	108.6/sec	465.3

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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/l...	3586	1163	675	2767	1	19640	0.31%	23.3/sec	96.1
/j_spring_...	3575	2295	1701	4894	0	19338	0.64%	22.9/sec	103.2
/	3552	1127	667	2684	1	13245	0.14%	22.7/sec	97.5
/configura...	3547	1162	661	2853	1	15593	0.17%	22.7/sec	94.8
/configura...	3541	1121	653	2553	0	13443	0.25%	22.7/sec	97.2
TOTAL	17801	1375	809	3326	0	19640	0.30%	110.8/sec	473.6

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).

#### 2.1.15. 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*.

#### 2.1.16. Casos de uso

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

8. No loguearse
9. Acceder al menú y pulsar en Register a provider

#### 2.1.17. Pruebas realizadas

##### Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/register/ac...	2000	19	16	34	6	176	0.00%	44.9/sec	349.9
/welcome/i...	1000	9	8	12	5	56	0.00%	23.1/sec	69.0
TOTAL	3000	15	11	30	5	176	0.00%	64.9/sec	401.9

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 = 0,034s.

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

##### Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

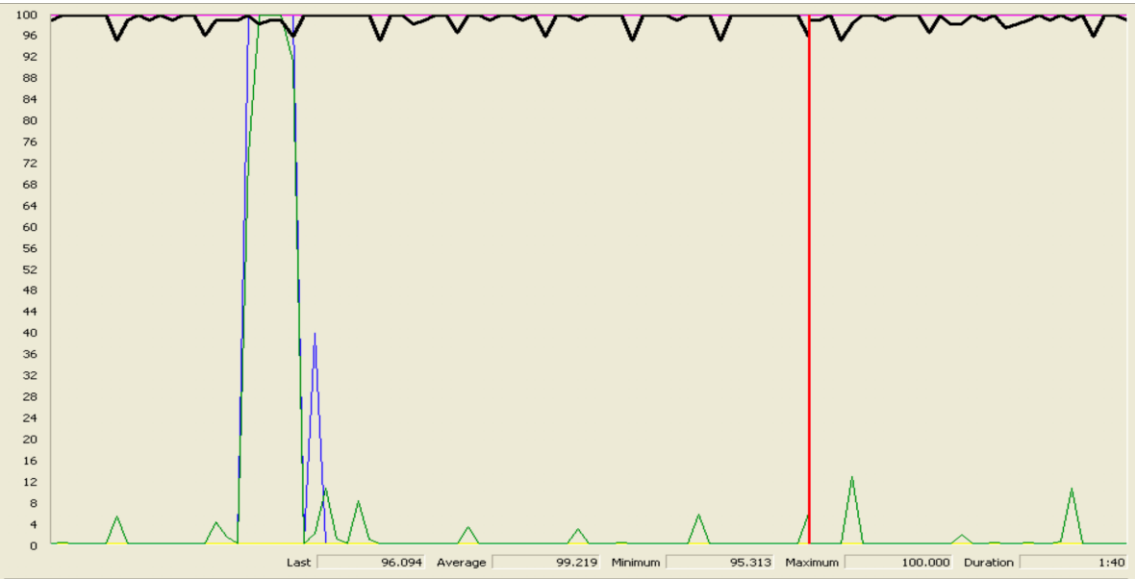
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/register/ac...	4000	49	19	72	5	1734	0.00%	76.9/sec	600.4
/welcome/i...	2000	15	9	25	4	611	0.00%	39.9/sec	119.8
TOTAL	6000	38	15	55	4	1734	0.00%	112.5/sec	697.7

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 = 0,072s.

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
/register/ac...	9000	96	27	179	5	8232	0.00%	46.6/sec	363.8
/welcome/i...	4500	57	13	116	4	4488	0.00%	23.6/sec	70.9
TOTAL	13500	83	21	155	4	8232	0.00%	69.7/sec	432.2

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 = 0,179secs.

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
/register/ac...	15000	188	38	478	5	10796	0.00%	46.6/sec	363.8
/welcome/i...	7500	170	21	441	4	6908	0.00%	23.4/sec	70.4
TOTAL	22500	182	33	467	4	10796	0.00%	69.6/sec	431.8

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 = 0,478secs.

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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/register/a...	885	498	108	1418	10	6728	4.63%	106.1/sec	798.3
/welcome/...	225	307	39	996	5	2390	0.00%	42.6/sec	128.1
TOTAL	1110	459	93	1354	5	6728	3.69%	133.1/sec	879.4

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).

#### 2.1.18. 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*.

#### 2.1.19. Casos de uso

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

10. Loguearse
11. Listar mis items

12. Crear una item
13. Editar ese item
14. Ver ese ítem
15. Eliminar ese item

## 2.1.20. Pruebas realizadas

### Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/lo...	1000	11	7	21	4	146	0.00%	9.1/sec	29.7
/j_spring_s...	1000	26	20	43	6	253	0.00%	9.0/sec	29.3
/	1000	18	14	32	7	231	0.00%	9.0/sec	27.3
/item/provid...	1000	13	11	22	6	80	0.00%	9.0/sec	33.1
/item/provid...	1000	12	10	18	6	46	0.00%	9.1/sec	37.4
/item/provid...	1000	22	19	35	12	96	0.00%	8.9/sec	35.2
/item/provid...	2000	17	15	26	9	118	0.00%	17.4/sec	77.1
/item/provid...	1000	24	20	39	7	112	0.00%	8.9/sec	35.1
TOTAL	9000	18	15	31	4	253	0.00%	72.5/sec	274.8

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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/fl...	2500	58	14	109	4	4048	0.00%	21.1/sec	69.0
/j_spring_...	2500	90	39	209	11	1647	0.00%	21.1/sec	75.1
/	2500	47	25	89	6	1066	0.00%	21.1/sec	70.2
/item/provi...	2500	44	21	89	6	970	0.00%	21.0/sec	77.7
/item/provi...	2500	33	18	59	5	1313	0.00%	21.0/sec	87.0
/item/provi...	2500	72	39	138	11	1379	0.00%	21.1/sec	83.5
/item/provi...	5000	48	27	85	7	1196	0.00%	41.2/sec	183.3
/item/provi...	2500	77	44	162	10	1308	0.00%	21.1/sec	83.5
TOTAL	22500	57	27	113	4	4048	0.00%	173.8/sec	672.4

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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/lo...	2000	51	13	94	4	3005	0.00%	16.4/sec	53.7
/j_spring_s...	2000	89	37	239	9	1155	0.00%	16.5/sec	54.0
/j	2000	48	24	103	6	1075	0.00%	16.5/sec	50.2
/item/provid...	2000	53	23	121	6	1365	0.00%	16.6/sec	61.1
/item/provid...	2000	48	19	105	5	1119	0.00%	16.6/sec	68.7
/item/provid...	2000	77	38	188	10	1132	0.00%	16.5/sec	65.3
/item/provid...	4000	38	25	69	8	941	0.00%	32.2/sec	143.2
/item/provid...	2000	65	36	132	11	1221	0.00%	16.5/sec	65.3
TOTAL	18000	56	26	121	4	3005	0.00%	135.4/sec	514.6

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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/l...	6095	247	60	695	3	8242	0.03%	15.5/sec	51.0
/j_spring_...	6093	471	200	1263	0	8169	0.25%	15.5/sec	55.3
/j	6078	236	67	644	0	6393	0.03%	15.5/sec	51.6
/item/provi...	6076	237	63	661	0	7333	0.07%	15.5/sec	57.2
/item/provi...	6072	235	59	686	1	5272	0.05%	15.5/sec	63.9
/item/provi...	6069	452	180	1250	0	5684	0.30%	15.5/sec	61.1
/item/provi...	12101	226	63	634	7	5802	0.01%	30.6/sec	136.3
/item/provi...	6050	428	174	1193	10	6191	0.00%	15.4/sec	61.0
TOTAL	54634	306	88	875	0	8242	0.08%	135.3/sec	523.8

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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/lo...	3000	421	122	1010	5	10050	0.00%	17.9/sec	58.6
/j_spring_s...	3000	845	368	2121	11	11284	0.00%	17.6/sec	57.8
/	3000	435	135	1020	7	13749	0.00%	17.6/sec	53.7
/item/provid...	3000	396	140	971	8	11875	0.00%	17.7/sec	65.3
/item/provid...	3000	351	111	797	6	11098	0.00%	17.7/sec	73.3
/item/provid...	3000	708	324	1653	12	15224	0.00%	17.5/sec	69.4
/item/provid...	6000	320	112	796	9	12619	0.00%	34.5/sec	153.7
/item/provid...	3000	664	299	1498	11	21480	0.00%	17.7/sec	70.2
TOTAL	27000	496	177	1179	5	21480	0.00%	146.9/sec	559.5

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).

#### 2.1.21. 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.