



# PET-CLINIC (RENDIMIENTO)

## GRUPO 13

Jesús Andrés Rico Catalán

Carlos Albalat Heredia

# Índice

Introducción .....	2
Rendimiento del sistema.....	3
HU-1,3,5: Diagnosis .....	3
HU2: Diagnosis .....	5
HU-4,14: Diagnosis .....	9
HU6: Diagnosis .....	11
HU7: Diagnosis .....	14
HU8: Diagnosis .....	17
HU9: Diagnosis .....	19
HU10: Diagnosis .....	20
HU11: Diagnosis .....	20
HU12: Diagnosis .....	21
HU13: Diagnosis .....	21
HU6 vs HU8: Diagnosis .....	21
Conclusión .....	25

## Introducción

Para comenzar se va a dejar constancia de los equipos utilizados para realizar las pruebas de rendimiento de la aplicación:

Procesador	Intel(R) Core(TM) i7-4510U CPU @ 2.00GHz 2.60 GHz
RAM instalada	8,00 GB (7,89 GB usable)

Procesador	Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz
RAM instalada	8,00 GB (7,83 GB usable)

Es importante mencionar que dependiendo de las especificaciones del equipo en donde se corran las pruebas de rendimiento pueden variar considerablemente los resultados, pero independientemente de los usuarios que soporte nuestro equipo se puede diagnosticar los cuellos de botellas que puede producir el sistema, así como analizar la eficiencia de nuestras HUs.

Para comprobar el rendimiento del sistema se han realizado bajo las “assertions” siguientes:

```
setUp(  
  transportarNeg.inject(rampUsers(280) during (100 seconds)),  
  transportarPos.inject(rampUsers(280) during (100 seconds))  
) .protocols(httpProtocol)  
  .assertions(  
    global.responseTime.max.lt(5000),  
    global.responseTime.mean.lt(1000),  
    global.successfulRequests.percent.gt(95)  
  )  
)
```

Consideramos que nuestro sistema tiene un rendimiento adecuado cuando se satisfacen el 95% de las peticiones y la media de los tiempos de respuestas de dichas peticiones no superan el segundo. El tiempo de respuesta máximo verifica que en 5 segundos todas las respuestas han sido satisfechas, cuando no se cumple dicha condición podemos analizar de manera sencilla donde están los cuellos de botella así como Timeout cuando el sistema es sometido a mucho estrés. Dependiendo del dispositivo en el que se ejecuta y la naturaleza de la prueba el rendimiento máximo de cada historia será diferente.

## Rendimiento del sistema

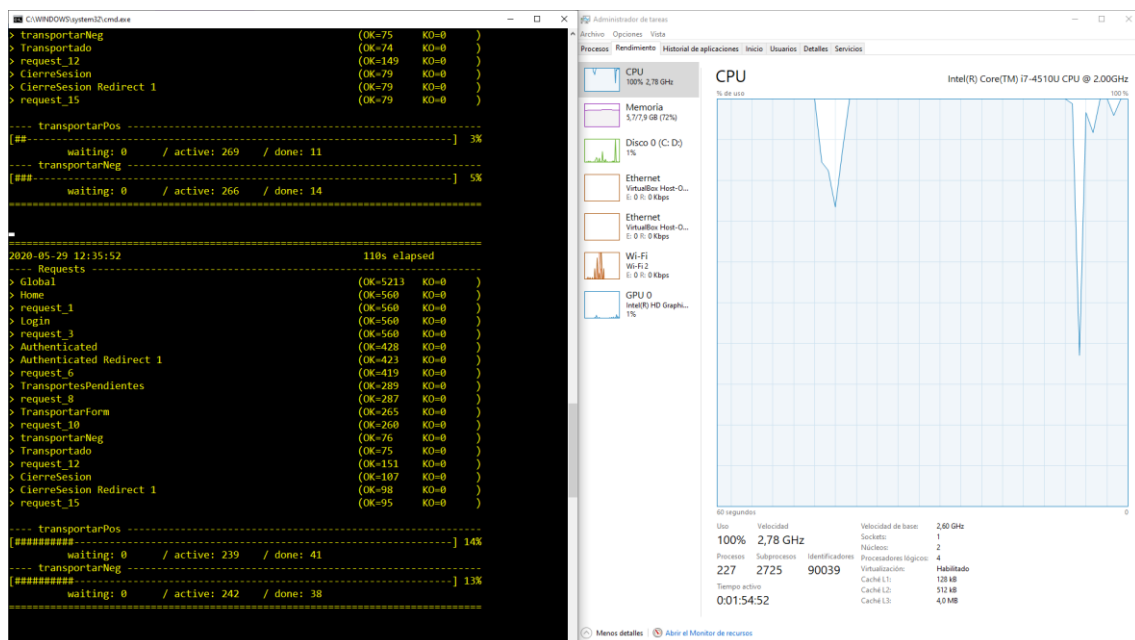
Algunos test de rendimiento incluyen varias HUs en el mismo test debido a la naturaleza de esta. Debido a que cada compañero posee un equipo con distintas especificaciones varía considerablemente el rendimiento de nuestra aplicación.

### HU-1,3,5: Diagnosis

El siguiente test engloba estas HUs:

- El secretario debe poder transportar animales consultando en la base de datos animales por transportar.
- El secretario debe poder listar de manera sencilla los animales que necesitan un transporte.
- El secretario solo puede solicitar un transporte con varios animales si esos animales pertenecen al mismo dueño.

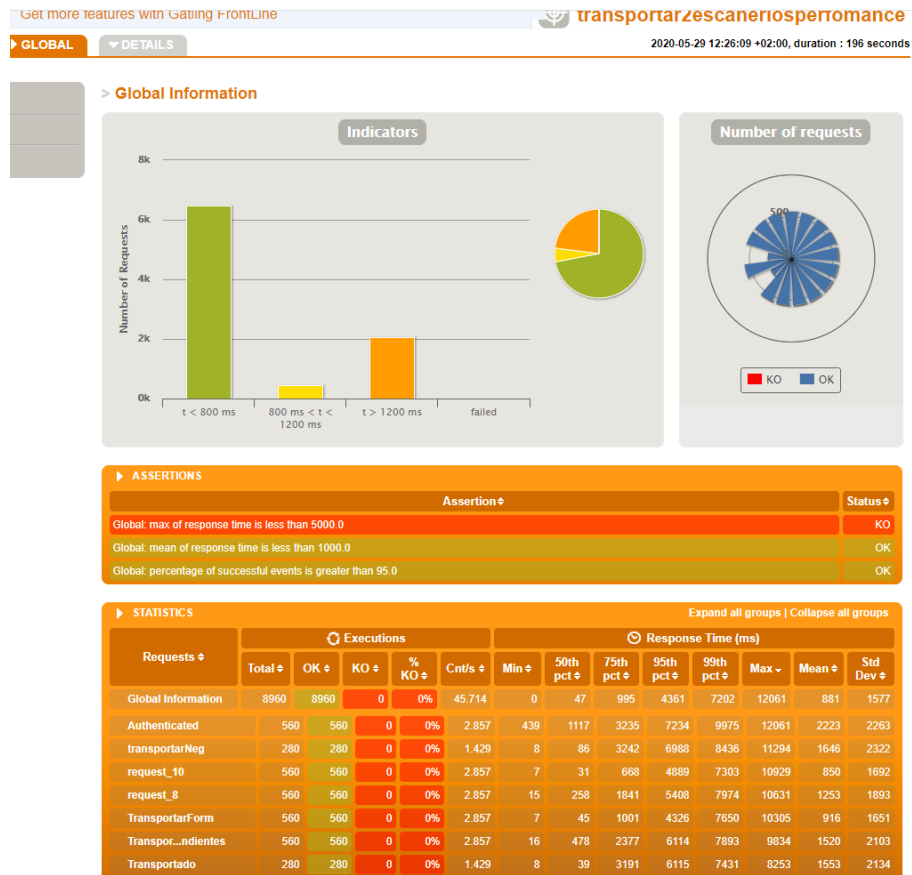
El máximo número de usuarios concurrentes que soporta de manera correcta es 560. Un aumento de 20 usuarios concurrente provoca una gran saturación provocando que el sistema para estas historias sea inviable usar la aplicación. La causa de que nuestro servidor deje de responder de manera óptima es debido a que la CPU no es capaz de procesar tantas peticiones concurrentemente propiciando cuellos de botellas y por lo tanto muchos TimeOuts como se puede ver en la siguiente captura:

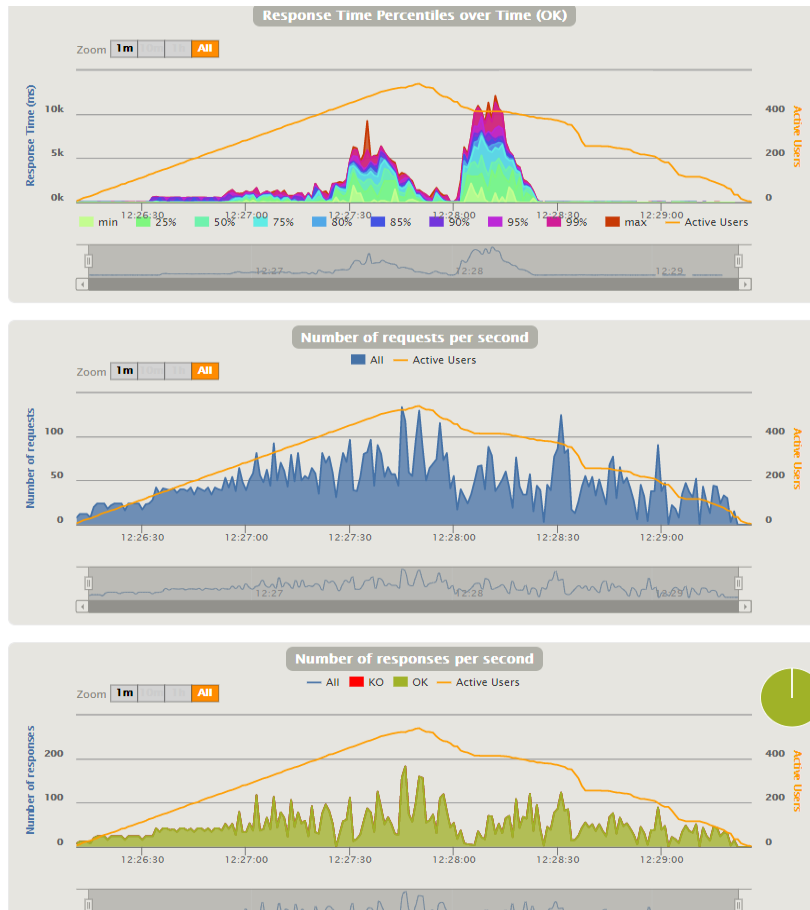


\*En este test se enfrenta un escenario positivo de transportar y otro negativo.

Cabe destacar que una vez se mejore las especificaciones de la CPU aumentará considerablemente el número de usuario concurrentes del sistema llegando a un punto en el

que habrá que mejorar la RAM o aumentar su capacidad ya que prácticamente llega al 80% de su funcionamiento.





En el report generado por Gatling podemos comprobar que el sistema satisface como mínimo el 95% de peticiones http que genera el test. También se verifica que la media de las peticiones es menor a un segundo aunque se pueden observar que al someter a cierto estrés hay una serie de peticiones que necesitan consultar la base de datos y por lo tanto un mayor procesamiento generando cuellos de botellas como son las autenticación y los post (edición) de la entidad transporte pero son casos puntuales.

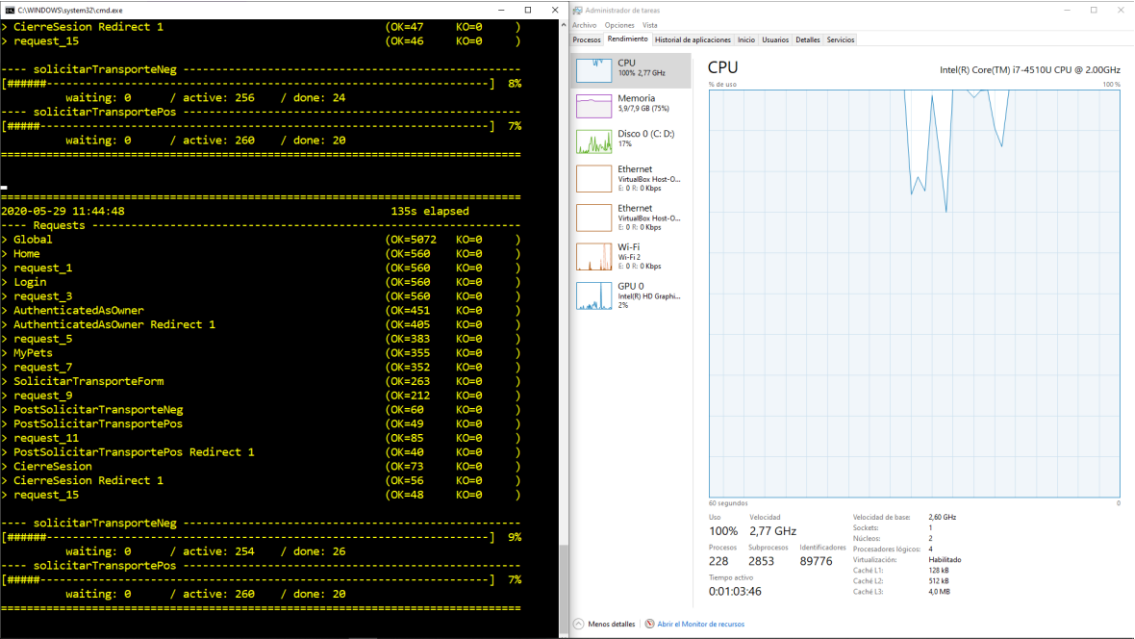
## HU2: Diagnosis

El siguiente test engloba esta HU:

- El propietario puede solicitar transportar a su animal.

El máximo número de usuarios concurrentes que se soporta es de 560. Un aumento de 35 usuarios concurrente provoca una gran saturación provocando que el sistema para esta historia sea inviable usar la aplicación. La causa de que nuestro servidor deje de responder de manera óptima es debido a que la CPU no es capaz de procesar tantas peticiones

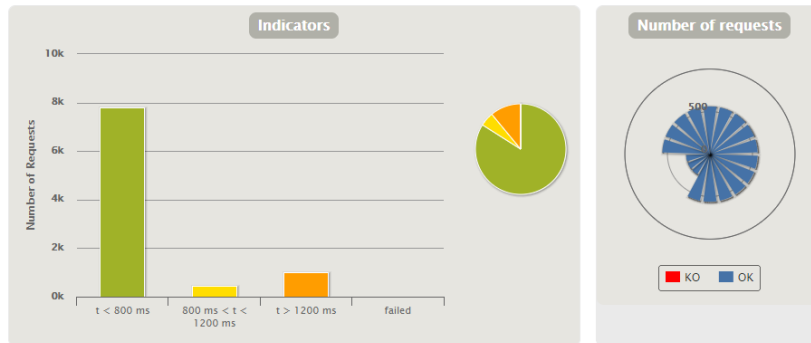
concurrentemente propiciando cuellos de botellas y por lo tanto muchos TimeOuts como se puede ver en la siguiente captura:



\*En este test se enfrenta un escenario positivo de solicitarTransporte y otro negativo.

Cabe destacar que una vez se mejore las especificaciones de la CPU aumentará considerablemente el número de usuario concurrentes del sistema llegando a un punto en el que habrá que mejorar la RAM o aumentar su capacidad ya que prácticamente llega al 80% de su funcionamiento.

## &gt; Global Information



## &gt; ASSERTIONS

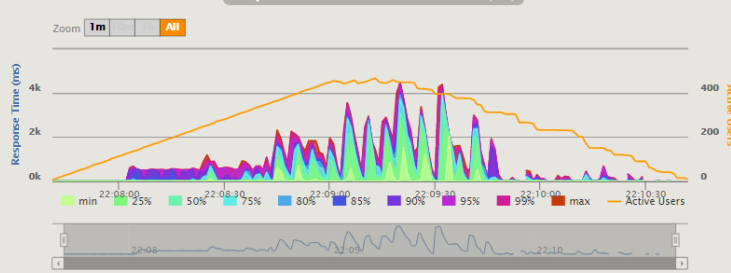
Assertion	Status
Global: max of response time is less than 5000.0	OK
Global: mean of response time is less than 1000.0	OK
Global: percentage of successful events is greater than 95.0	OK

## &gt; STATISTICS

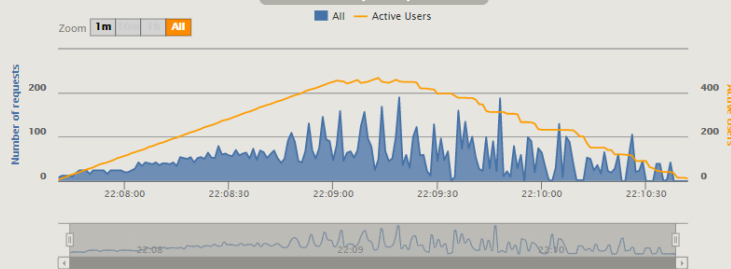
Expand all groups | Collapse all groups

Requests ↕	🔄 Executions					⌚ Response Time (ms)							
	Total ↕	OK ↕	KO ↕	% KO ↕	Cnt/s ↕	Min ↕	50th pct ↕	75th pct ↕	95th pct ↕	99th pct ↕	Max ↕	Mean ↕	Std Dev ↕
Global Information	9240	9240	0	0%	50.769	0	47	481	2072	3097	4454	391	696
PostSoli...ortePos	280	280	0	0%	1.538	51	221	671	2118	2943	4454	535	710
request_3	560	560	0	0%	3.077	3	9	118	786	2017	4425	160	405
Authenti...dAsOwner	560	560	0	0%	3.077	374	1096	2007	3495	4072	4377	1403	990
PostSoli...orteNeg	280	280	0	0%	1.538	8	14	299	1514	2708	4061	314	592
Solicita...orteForm	560	560	0	0%	3.077	8	165	978	2476	3103	3998	612	821
request_5	560	560	0	0%	3.077	3	65	281	775	1563	3974	208	338
request_15	560	560	0	0%	3.077	2	6	179	719	1929	3963	164	398

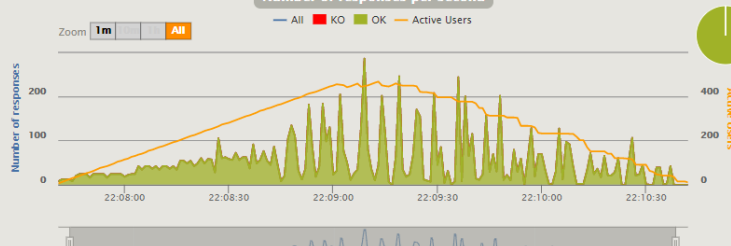
## Response Time Percentiles over Time (OK)



## Number of requests per second

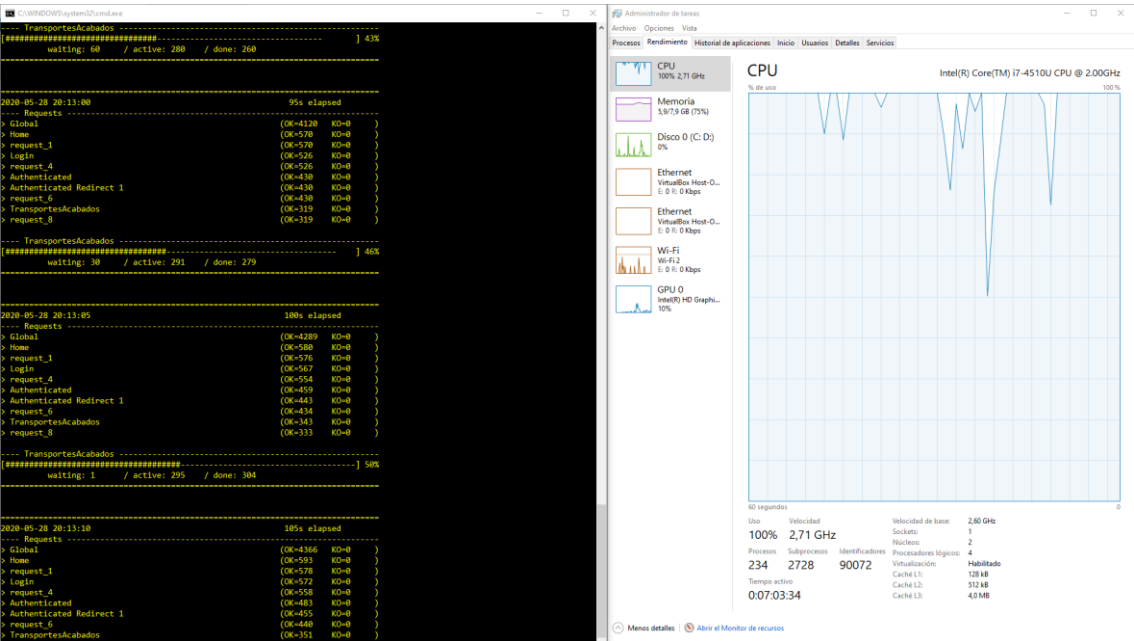


## Number of responses per second





En el report generado por Gatling podemos comprobar que el sistema satisface como mínimo el 95% de peticiones http que genera el test. También se verifica que la media de las peticiones es menor a un segundo y que ninguna petición sup



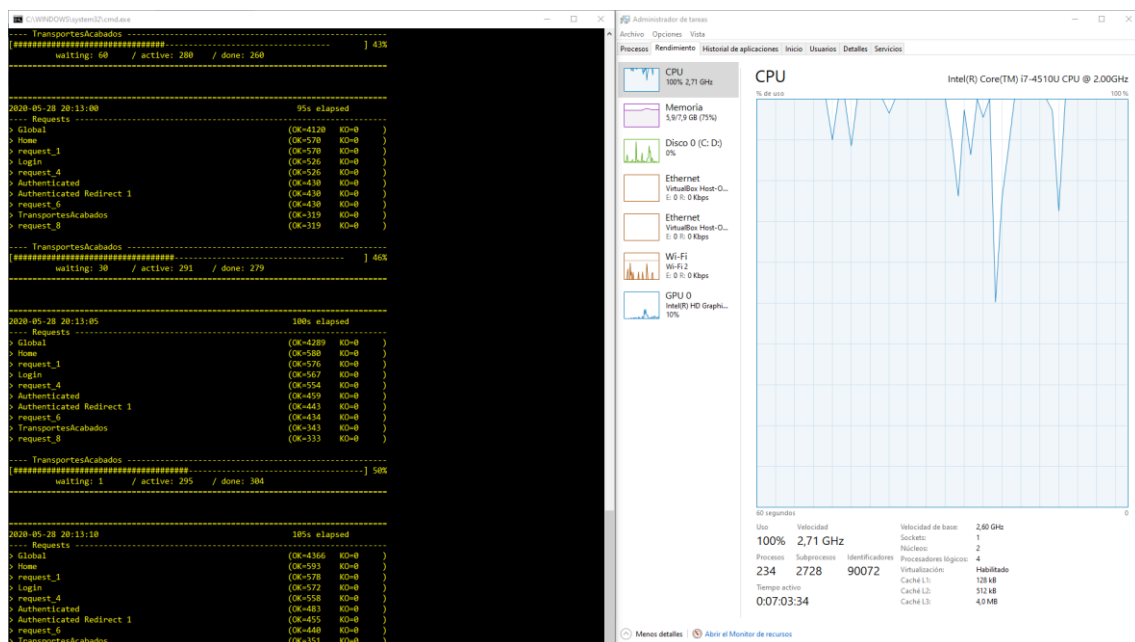
era un tiempo de respuesta mayor a 5 segundos. Por lo tanto, se podría decir que el comportamiento es idóneo ya que apenas se producen cuellos de botella.

## HU-4,14: Diagnosis

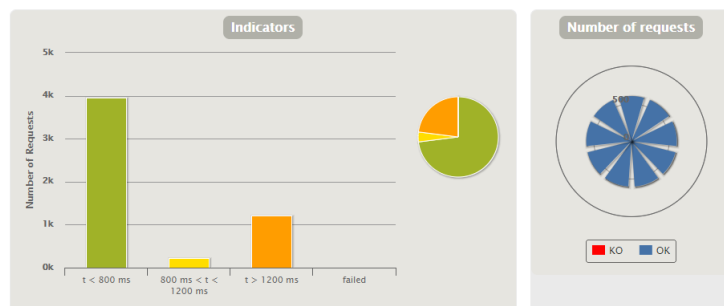
El siguiente test engloba esta HU:

- El propietario debe poder ver todos sus transportes acabados.
- El secretario debe poder ver los animales que han sido transportados.

Debido a la naturaleza de la HU que solo consulta la DB y no realiza ningún tipo de inserción o eliminación, el máximo número de usuarios concurrentes es algo mayor a las otras historias de transportes, esta es de 600 usuarios. Un aumento de 30 usuarios concurrente provoca una gran saturación provocando que el sistema para esta historia sea inviable usar la aplicación. La causa de que nuestro servidor deje de responder de manera óptima es debido a que la CPU no es capaz de procesar tantas peticiones concurrentemente propiciando cuellos de botellas y por lo tanto muchos TimeOuts, como se puede ver en la siguiente captura:

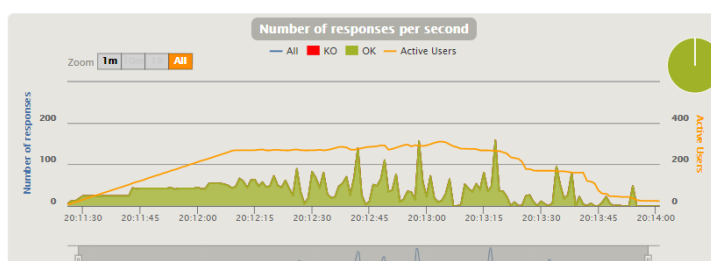
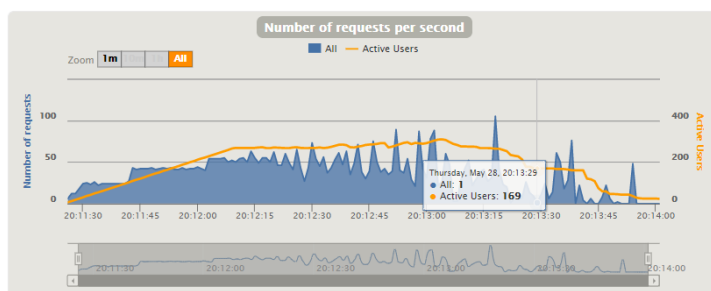
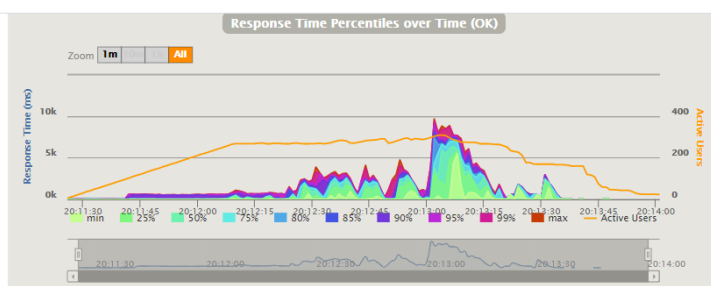


> Global Information



ASSERTIONS	
Assertion	Status
Global: max of response time is less than 5000.0	KO
Global: mean of response time is less than 1000.0	OK
Global: percentage of successful events is greater than 95.0	OK

Requests	Executions											Response Time (ms)		
	Total	OK	KO	% KO	Cnt/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev	
Global Information	5400	5400	0	0%	34.615	3	52	941	3316	6678	9661	747	1354	
Authenticated	600	600	0	0%	3.846	464	1241	2423	5790	8820	9661	1790	1701	
Authenti...direct 1	600	600	0	0%	3.846	3	159	1213	3901	6066	7725	833	1345	
TransportesAcabados	600	600	0	0%	3.846	7	50	965	4355	7399	8106	811	1514	
Login	600	600	0	0%	3.846	3	28	1030	2924	5102	8615	647	1206	
Home	600	600	0	0%	3.846	4	18	612	2613	7288	8810	637	1411	



En el report generado por Gatling podemos comprobar que el sistema satisface como mínimo el 95% de peticiones http que genera el test. También se verifica que la media de las peticiones

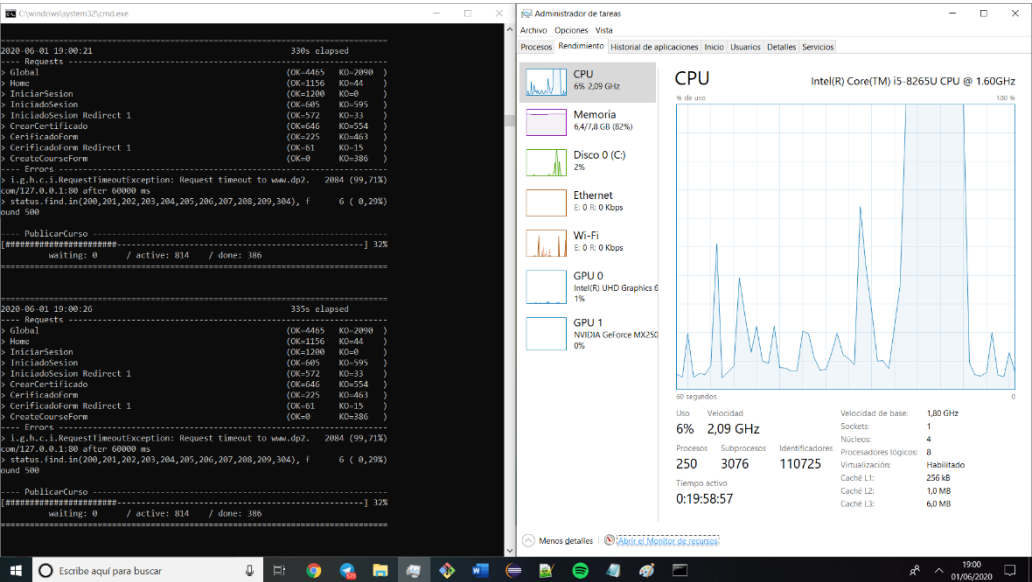
es menor a un segundo aunque se pueden observar que al someter a cierto estrés hay una serie de peticiones que necesitan consultar la base de datos y por lo tanto un mayor procesamiento generando cuellos de botellas como son las autenticación y la query de transportes acabados pero son casos puntuales.

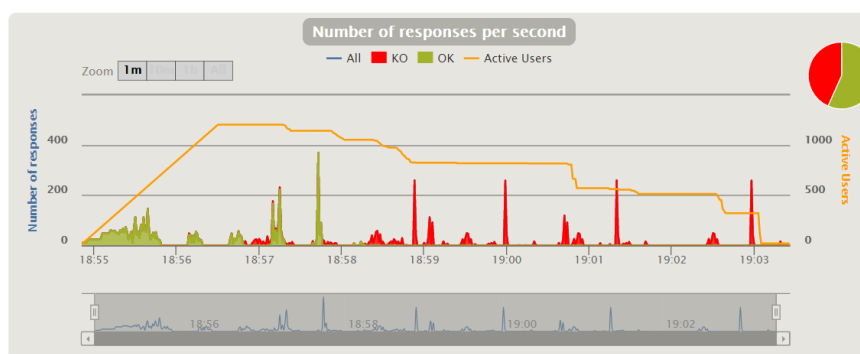
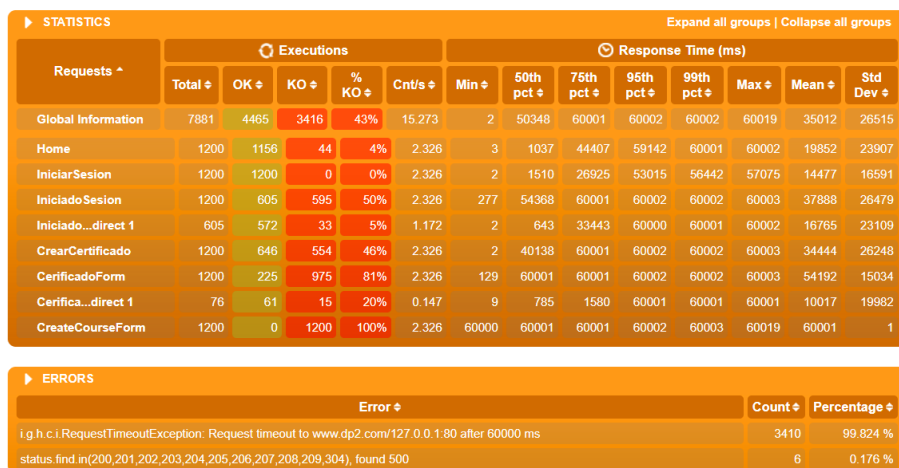
## HU6: Diagnosis

Este test de rendimiento se centra en la siguiente HU:

- El secretario debe poder publicar cursos.

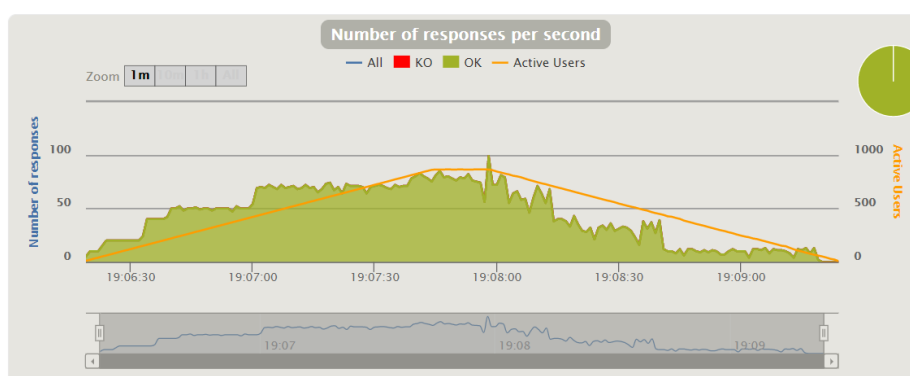
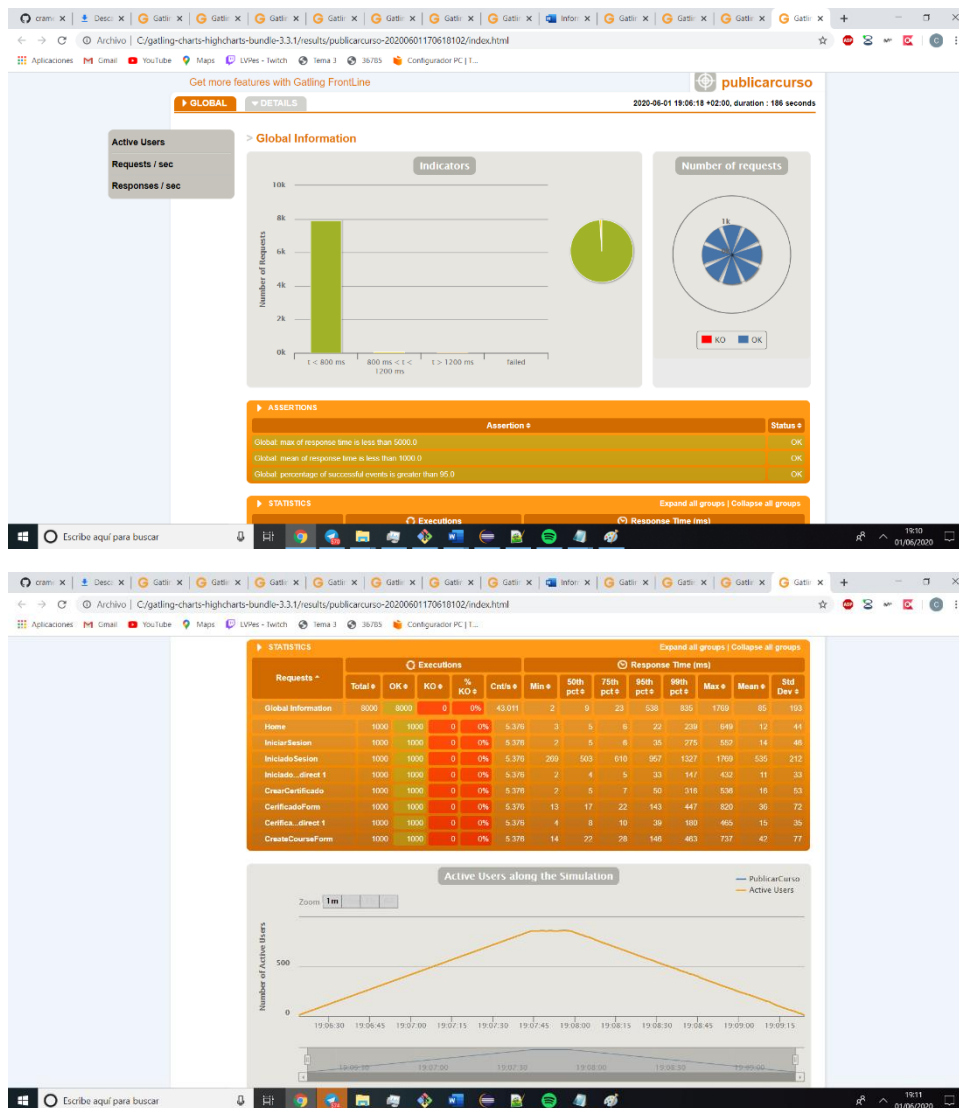
El primer parámetro a identificar es el mínimo de usuarios concurrentes que no puede soportar esta HU junto a una captura de pantalla sobre el rendimiento del sistema mostrando el cuello de botella en la CPU y/o RAM.





Esta captura fue realizada mientras se realizaba un 'stress test' de 1200 usuarios al sistema. Se puede identificar en la consola de Gatling que se producen errores de TimeOut en su mayoría, junto a otro error 500 que aparece menos. En el rendimiento del sistema se aprecia como a pesar de haber cargado ya todos los usuarios hay un pico del 100% (mencionar que al arrancar el test para cargar los usuarios se mantiene al 100%), dando lugar a un cuello de botella en la CPU. Cabe mencionar que se puede observar también el archivo index.html generado por Gatling para verificar que efectivamente el comportamiento es malísimo, dejando sin respuesta a un gran porcentaje de peticiones.

Una vez identificado el mínimo de usuarios que soporta este escenario (HU), se ha ido disminuyendo el número de usuarios concurrentes hasta llegar a diagnosticar el segundo parámetro solicitado, el máximo número de usuarios concurrentes para los cuales se obtiene un buen rendimiento por parte del sistema. En este con 1000 usuarios se realiza un rendimiento magnífico como se puede apreciar en la siguiente captura del index.html proporcionado por Gatling.



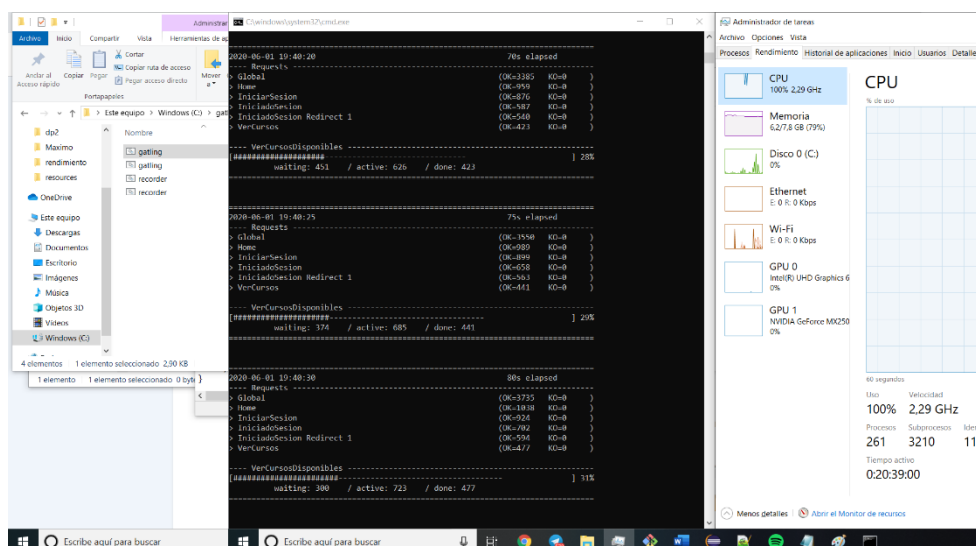
Se observa que satisface las 3 'assertions' indicadas en el archivo .scala para la ejecución de este test y que todas las peticiones son resueltas de manera satisfactoria en un tiempo menor a 0,8 s (800ms)

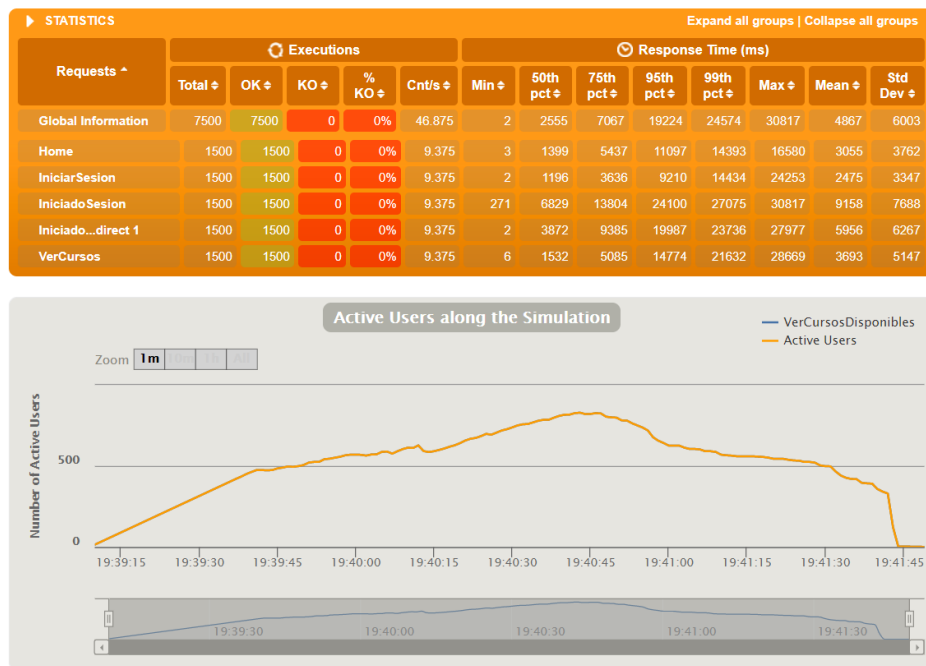
## HU7: Diagnosis

Este test de rendimiento se centra en la siguiente HU:

- El propietario de un animal debe poder listar los cursos disponibles en el sistema.

El primer parámetro a identificar es el mínimo de usuarios concurrentes que no puede soportar esta HU junto a una captura de pantalla sobre el rendimiento del sistema mostrando el cuello de botella en la CPU y/o RAM.





Esta captura fue realizada mientras se realizaba un 'stress test' de 1500 usuarios al sistema. En este caso no se han identificado ningún error, es decir, todas las peticiones han sido contestadas. Sin embargo, se ve como la CPU está saturada, manteniéndose durante todo el test en un 100%, dando lugar a un tiempo de respuesta nada óptimo. En esta situación el rendimiento no es malísimo, pero no es lo que queremos ya que en el reporte generado por Gatling se puede observar que el 63% de las peticiones tarda más de 1,2 s. Siendo éste un rendimiento bastante mejorable. No cumple las dos 'assertions' del tiempo de respuesta.

Una vez identificado el mínimo de usuarios que soporta este escenario (HU), se ha ido disminuyendo el número de usuarios concurrentes hasta llegar a diagnosticar el segundo parámetro solicitado, el máximo número de usuarios concurrentes para los cuales se obtiene un buen rendimiento por parte del sistema. En este con 1200 usuarios se realiza un rendimiento magnífico como se puede apreciar en la siguiente captura del index.html proporcionado por Gatling.

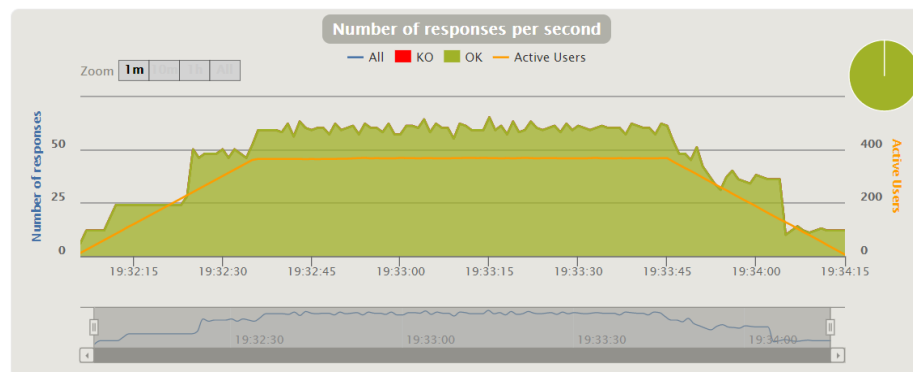
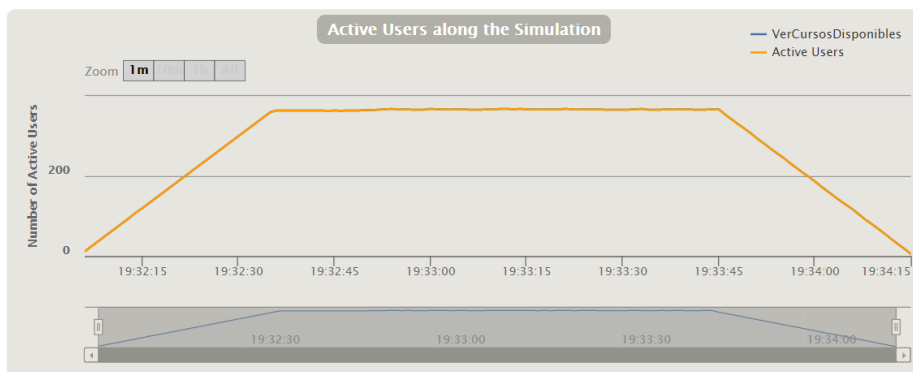




STATISTICS

Expand all groups | Collapse all groups

Requests	Executions					Response Time (ms)							
	Total	OK	KO	% KO	Cnt/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev
Global Information	6000	6000	0	0%	46.154	2	6	15	530	610	844	103	195
Home	1200	1200	0	0%	9.231	2	5	6	13	24	50	6	4
IniciarSesion	1200	1200	0	0%	9.231	2	5	6	10	21	70	6	4
IniciadoSesion	1200	1200	0	0%	9.231	274	496	530	610	725	844	485	93
Iniciado...direct 1	1200	1200	0	0%	9.231	2	4	5	9	19	47	5	3
VerCursos	1200	1200	0	0%	9.231	7	12	14	22	34	87	14	5



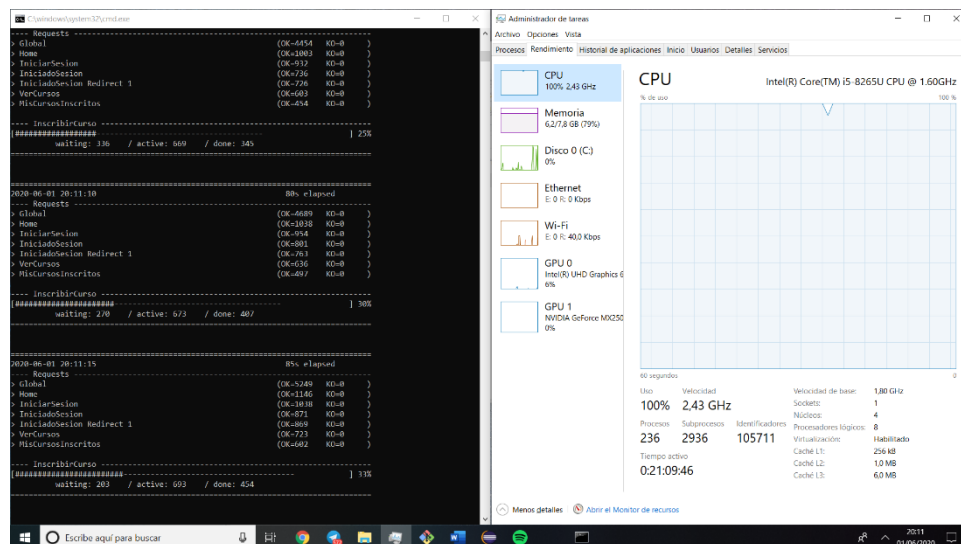
Se observa que satisface las 3 'assertions' indicadas en el archivo .scala para la ejecución de este test y que todas las peticiones son resueltas de manera satisfactoria en un tiempo menor a 0,8 s (800ms).

## HU8: Diagnosis

Este test de rendimiento se centra en la siguiente HU:

- El propietario de un animal debe poder inscribirse en un curso si no ha comenzado

El primer parámetro a identificar es el mínimo de usuarios concurrentes que no puede soportar esta HU junto a una captura de pantalla sobre el rendimiento del sistema mostrando el cuello de botella en la CPU y/o RAM.



Esta captura fue realizada mientras se realizaba un 'stress test' de 1500 usuarios al sistema. En este caso no se han identificado ningún error, es decir, todas las peticiones han sido contestadas. Sin embargo, se ve como la CPU está saturada, manteniéndose durante todo el test en un 100%, dando lugar a un tiempo de respuesta nada óptimo. En esta situación el rendimiento no es malísimo, pero no es lo que queremos ya que en el reporte generado por Gatling se puede observar que el 30% de las peticiones tarda más de 1,2 s y un 10% tarda entre 0,8s y 1,2s. Siendo un éste un rendimiento bastante mejorable, ya que no se cumplen dos 'assertions' de Gatling.

Una vez identificado el mínimo de usuarios que soporta este escenario (HU), se ha ido disminuyendo el número de usuarios concurrentes hasta llegar a diagnosticar el segundo parámetro solicitado, el máximo número de usuarios concurrentes para los cuales se obtiene un buen rendimiento por parte del sistema.



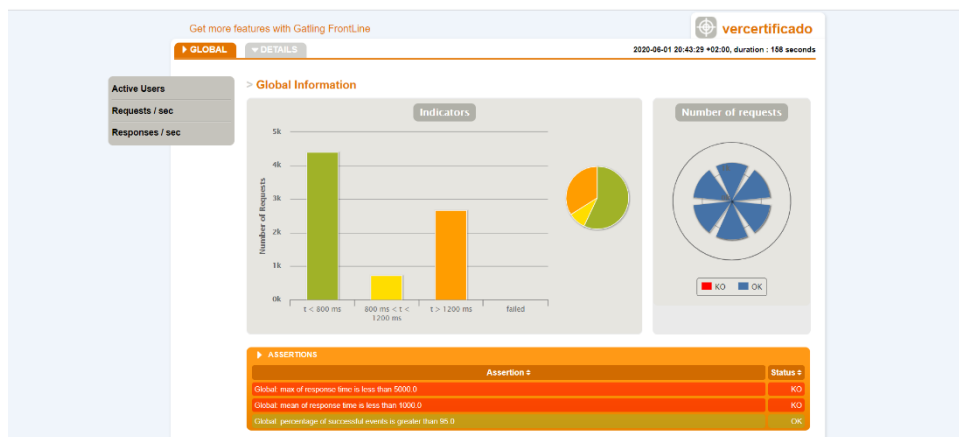
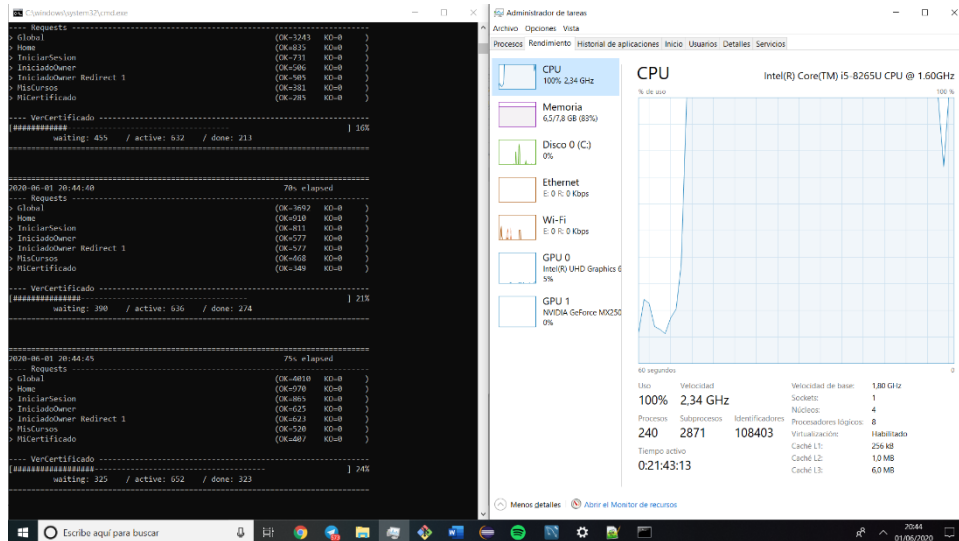
En este con 1100 usuarios se realiza un rendimiento que puede ser mejorable, pero sigue siendo bastante bueno ya que un 94% de las peticiones son respondidas en menos de 0,8s y se cumplen todas las 'assertions' de gatling.

## HU9: Diagnosis

Este test de rendimiento se centra en la siguiente HU:

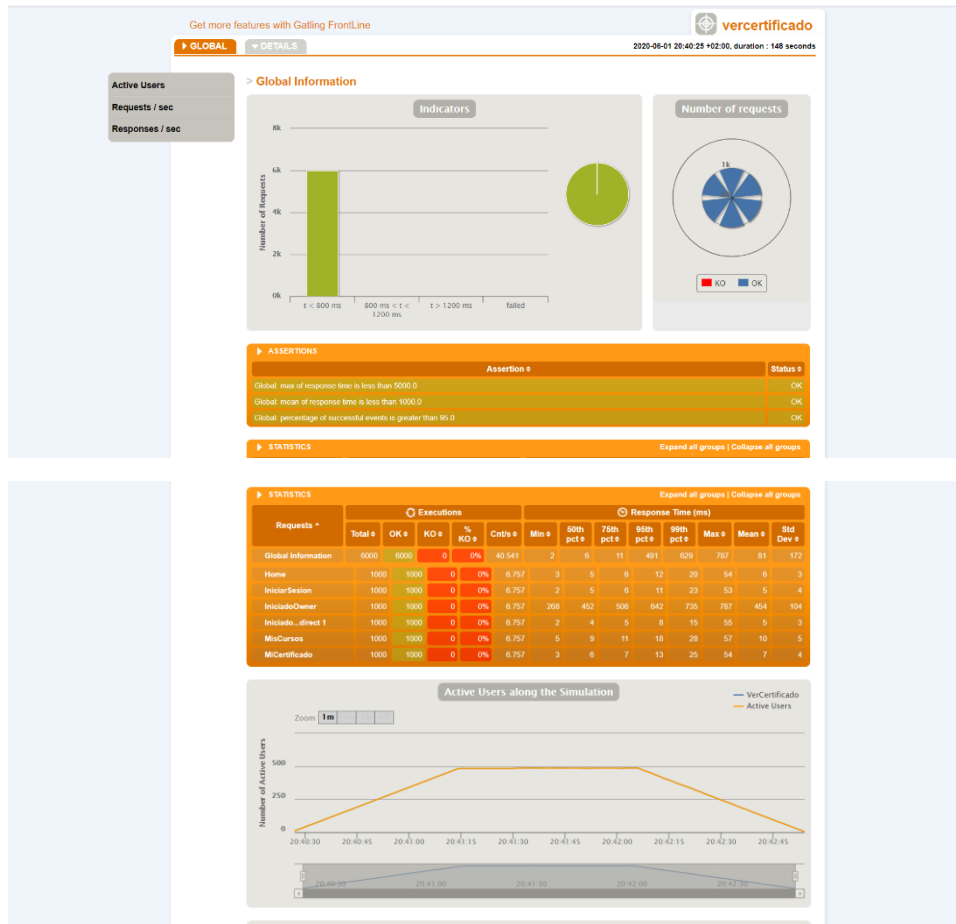
- El propietario debe poder obtener un certificado del curso finalizado.

El primer parámetro a identificar es el mínimo de usuarios concurrentes que no puede soportar esta HU junto a una captura de pantalla sobre el rendimiento del sistema mostrando el cuello de botella en la CPU y/o RAM.



Esta captura fue realizada mientras se realizaba un 'stress test' de 1300 usuarios al sistema. En este caso no se han identificado ningún error, es decir, todas las peticiones han sido contestadas. Sin embargo, se ve como la CPU está saturada, manteniéndose durante todo el test en un 100%, dando lugar a un tiempo de respuesta nada óptimo. En esta situación el rendimiento no es malísimo, pero no es lo que queremos ya que en el reporte generado por Gatling se puede observar que el 34% de las peticiones tarda más de 1,2 s y un 9% tarda entre 0,8s y 1,2s. Siendo un éste un rendimiento bastante mejorable, ya que no se cumplen dos 'assertions' de Gatling.

Una vez identificado el mínimo de usuarios que soporta este escenario (HU), se ha ido disminuyendo el número de usuarios concurrentes hasta llegar a diagnosticar el segundo parámetro solicitado, el máximo número de usuarios concurrentes para los cuales se obtiene un buen rendimiento por parte del sistema.



En este con 1000 usuarios se realiza un rendimiento excelente con un 100% de las peticiones respondidas en menos de 0,8s.

## HU10: Diagnosis

Esta HU es la siguiente:

- El propietario de un animal exótico debe poder obtener un seguro al terminar el curso.

Esta HU, se ha considerado no someterle a test de rendimiento ya que es ver tus cursos en los que estás inscrito, pero con un atributo más dicha tabla. A nivel de rendimiento es exactamente lo mismo que la HU7, ya que los pasos son exactamente los mismos.

## HU11: Diagnosis

Esta HU es la siguiente:

- El propietario de un animal exótico, una vez realizado el curso debe poder consultar el seguro otorgado.

Esta HU, se ha considerado no someterle a test de rendimiento ya que los pasos son exactamente igual que la HU9. Teniendo además el mismo número de atributos la entidad Certificado que la entidad Seguro. Es lo mismo, únicamente que un seguro te aparecerá en el caso de que tengas una mascota exótica además de haber finalizado el curso de manera satisfactoria.

## HU12: Diagnosis

Esta HU es la siguiente:

- El propietario debe tener un descuento del 10% para siguientes inscripciones a cursos.

Esta HU, se ha considerado no someterle a test de rendimiento un motivo similar a la HU10. Ya que un propietario si ha finalizado dos cursos o más, le aparecerá un nuevo atributo en la tabla de los cursos disponibles (HU7) llamado precio rebajado.

## HU13: Diagnosis

Esta HU es la siguiente:

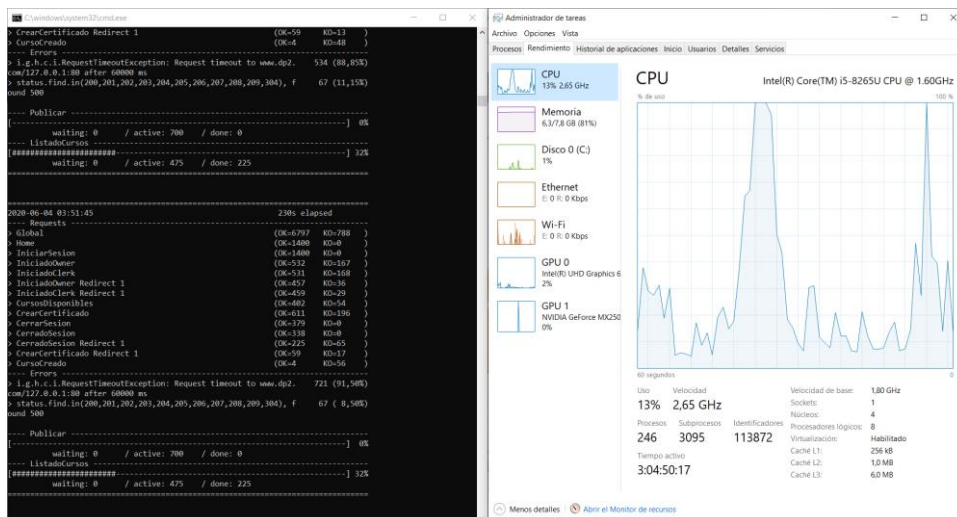
- El propietario debe poder ver toda la información de un curso antes de inscribirse.

Esta HU, se ha considerado no someterle a test de rendimiento ya que en el listado de cursos disponibles aparece toda la información relativa a ese curso. Ya que este consta de muy pocos atributos.

## HU6 vs HU8: Diagnosis

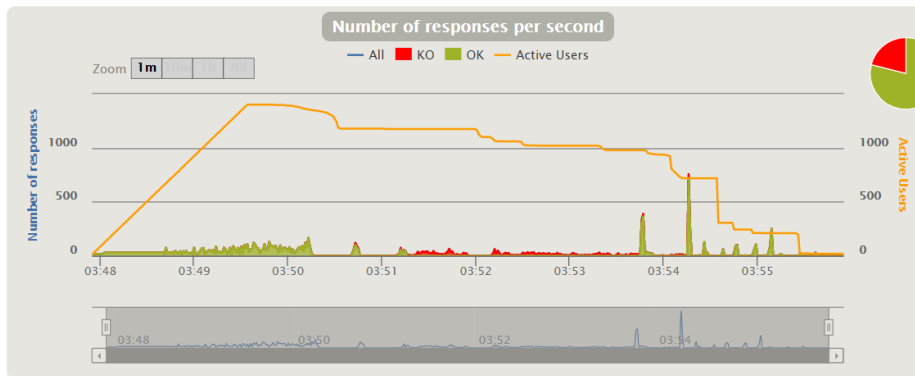
Este test se basa en comparar el rendimiento de la HU6 (Publicar un curso) frente a la HU8 (Inscribirse en un curso). Mencionar que la HU6 la realiza el rol Clerk y la HU8 la realiza el rol Owner.

El primer parámetro a identificar es el mínimo de usuarios concurrentes que no puede soportar esta HU junto a una captura de pantalla sobre el rendimiento del sistema mostrando el cuello de botella en la CPU y/o RAM.



STATISTICS		Expand all groups   Collapse all groups											
Requests ^	Executions					Response Time (ms)							
	Total	OK	KO	% KO	Cnt/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev
Global Information	12191	9623	2568	21%	25.345	0	10161	31861	60001	60002	60022	19937	22827
Home	1400	1400	0	0%	2.911	4	1423	10417	15679	18256	21228	5153	5907
IniciarSesion	1400	1400	0	0%	2.911	3	5142	10060	15199	18585	23122	5495	5597
IniciadoClerk	700	531	169	24%	1.455	786	14283	56152	60001	60001	60002	26238	22104
IniciadoOwner	700	532	168	24%	1.455	685	14172	55581	60001	60001	60002	26164	22041
Iniciado...direct 1	532	457	75	14%	1.106	6	8729	12970	60001	60001	60002	14937	18755
Iniciado...direct 1	531	459	72	14%	1.104	5	9818	13761	60001	60001	60002	15132	18354
CursoDisponibles	700	402	298	43%	1.455	1550	15883	60000	60001	60002	60004	30706	24133
CrearCertificado	1400	692	708	51%	2.911	4	54041	60001	60001	60002	60003	38421	23746
CerrarSesion	1400	1083	317	23%	2.911	2	19852	56325	60001	60002	60022	27273	23614
CerradoSesion	1400	1244	156	11%	2.911	0	19332	20201	60001	60001	60022	17304	18944
CerradoS...direct 1	1244	1131	113	9%	2.586	2	378	1080	60000	60001	60002	6402	17079
CrearCer...direct 1	84	59	25	30%	0.175	6	8704	31075	60000	60000	60001	16750	16525
CursoCreado	700	233	467	67%	1.455	3	60000	60001	60001	60002	60002	40602	26592

ERRORS			Error		Count	Percentage
			i.g.h.c.i.RequestTimeoutException: Request timeout to www.dp2.com/127.0.0.1:80 after 60000 ms		2379	92.64 %
			status.find.in(200,201,202,203,204,205,206,207,208,209,304), found 403		116	4.517 %
			status.find.in(200,201,202,203,204,205,206,207,208,209,304), found 500		73	2.843 %



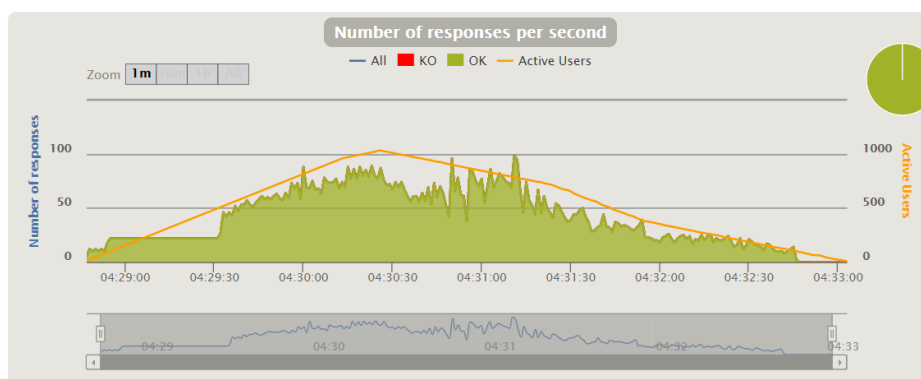
Esta captura fue realizada mientras se realizaba un 'stress test' de 700 usuarios por cada escenario al sistema (1400 en total). Se puede identificar en la consola de Gatling que se producen errores de TimeOut en su mayoría, junto a otro error 500 que aparece menos. En el rendimiento del sistema se aprecia como a pesar de haber cargado ya todos los usuarios hay un pico del 100% (mencionar que al arrancar el test para cargar los usuarios se mantiene al 100%), dando lugar a un cuello de botella en la CPU. Recordar que se puede observar también el archivo index.html generado por Gatling para verificar que efectivamente el comportamiento es malísimo, dejando sin respuesta a un gran porcentaje de peticiones.

Una vez identificado el mínimo de usuarios que soporta este escenario (HU), se ha ido disminuyendo el número de usuarios concurrentes hasta llegar a diagnosticar el segundo parámetro solicitado, el máximo número de usuarios concurrentes para los cuales se obtiene un buen rendimiento por parte del sistema.





▶ STATISTICS														Expand all groups   Collapse all groups			
Requests ^	🔄 Executions					🕒 Response Time (ms)											
	Total ▾	OK ▾	KO ▾	% KO ▾	Cnt/s ▾	Min ▾	50th pct ▾	75th pct ▾	95th pct ▾	99th pct ▾	Max ▾	Mean ▾	Std Dev ▾				
Global Information	10450	10450	0	0%	40.661	0	6	18	575	887	1566	84	201				
Home	1100	1100	0	0%	4.28	3	6	7	26	168	360	11	28				
IniciarSesion	1100	1100	0	0%	4.28	3	5	7	28	168	335	11	27				
IniciadoClerk	550	550	0	0%	2.14	273	558	683	987	1360	1566	600	203				
IniciadoOwner	550	550	0	0%	2.14	280	556	684	976	1342	1441	596	198				
Iniciado...direct 1	550	550	0	0%	2.14	2	5	11	144	441	532	25	69				
Iniciado...direct 1	550	550	0	0%	2.14	2	5	10	110	331	511	21	56				
CursosDisponibles	550	550	0	0%	2.14	8	14	24	249	564	1270	48	111				
CrearCertificado	1100	1100	0	0%	4.28	2	16	23	218	614	1314	46	125				
CerrarSesion	1100	1100	0	0%	4.28	2	5	8	137	445	674	26	79				
CerradoSesion	1100	1100	0	0%	4.28	0	1	1	39	345	530	12	55				
CerradoS...direct 1	1100	1100	0	0%	4.28	2	4	6	48	287	644	16	56				
CrearCer...direct 1	550	550	0	0%	2.14	5	8	11	59	333	654	21	57				
CursoCreado	550	550	0	0%	2.14	5	8	12	179	495	1333	34	111				



En este con 550 usuarios cada escenario (1100 en total) se realiza un rendimiento excelente con un 100% de las peticiones respondidas en menos de 0,8s.

## Conclusión

En primer lugar, consideramos que para tener una visión objetiva de nuestro sistema a nivel de rendimiento se debería de haber hecho en un solo dispositivo, ya que dependiendo de las especificaciones donde se corran los test puede variar considerablemente. Aun así, es de mucha utilidad para identificar el comportamiento del sistema cuando se somete a mucho estrés. El máximo número de usuarios concurrentes que soporta nuestra aplicación es de 560. Con este número de usuarios garantizamos que más del 95% de las peticiones se satisfacen y que la media del número de peticiones totales no superan el segundo, dando lugar a un funcionamiento más que correcto.

Por último, se han realizado pruebas en el que los usuarios concurrentes son introducidos en un corto periodo de tiempo, es decir bajando de esos 100 segundos. Esto ha significado para el sistema una bajada de rendimiento muy considerable, ya que el sistema debe satisfacer más peticiones en un periodo de tiempo más corto dando a lugar a un cuello de botella grande en las peticiones que interactúen con la base datos provocando TimeOuts. Es digno mencionar lo interesante que nos ha parecido la herramienta Gatling ya que teníamos experiencia con Jmeter en la cual hay una serie de mejoras sustanciales como la fácil configuración y la paralelización de peticiones proporcionando una mayor velocidad de ejecución de todo el test. También proporciona un “report” muy completo donde se puede realizar un análisis de manera sencilla y verificar con “assertions” posibles cuellos de botellas y posibles mejoras de rendimientos en métodos concretos.