

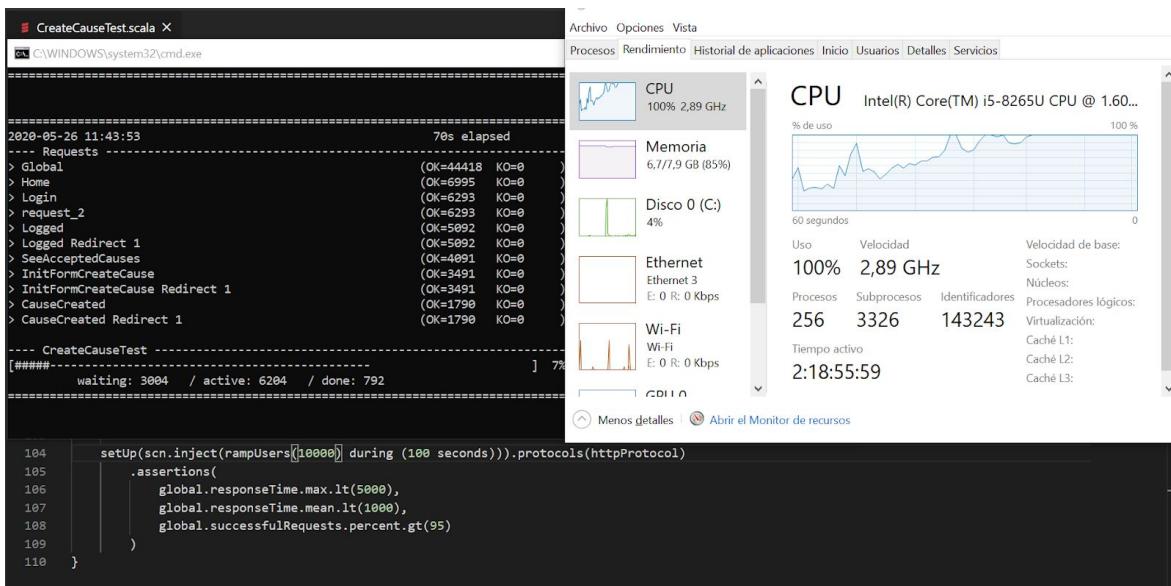
Test de rendimiento

Historia de Usuario 1	Yo COMO usuario QUIERO proponer causas benéficas en las que las personas puedan realizar donaciones durante la duración de la causa PARA ayudar a animales que tengan necesidades o no tengan dueño.
Escenario	Un admin crea una causa.
Nombre de Prueba	CreateCauseTest

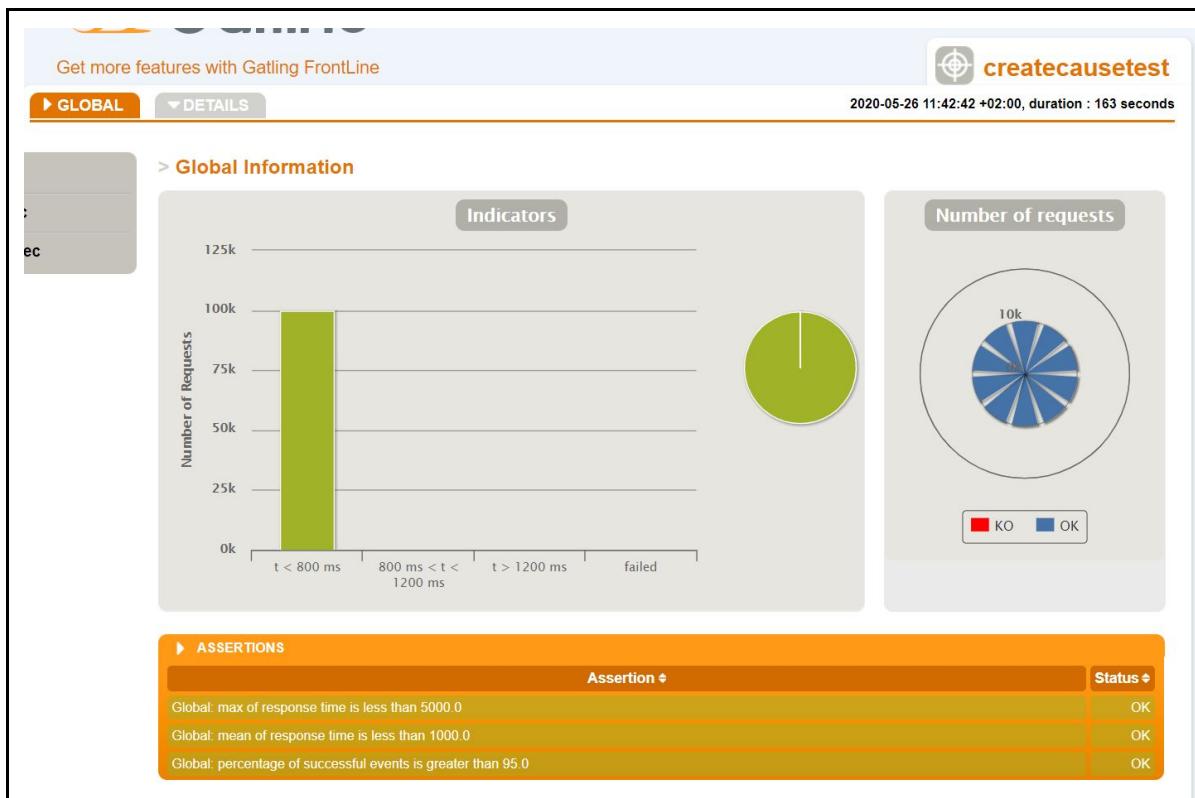
Máximo número de usuarios en los que el sistema funciona correctamente

Llegamos al límite en el que el sistema funciona adecuadamente probando con 10.000 usuarios en 100 segundos. Los resultados obtenidos son:

- Este es el resultado a la hora de aplicar la herramienta de gatling. Vemos que la CPU llega al 100% pero aún así el sistema responde correctamente.



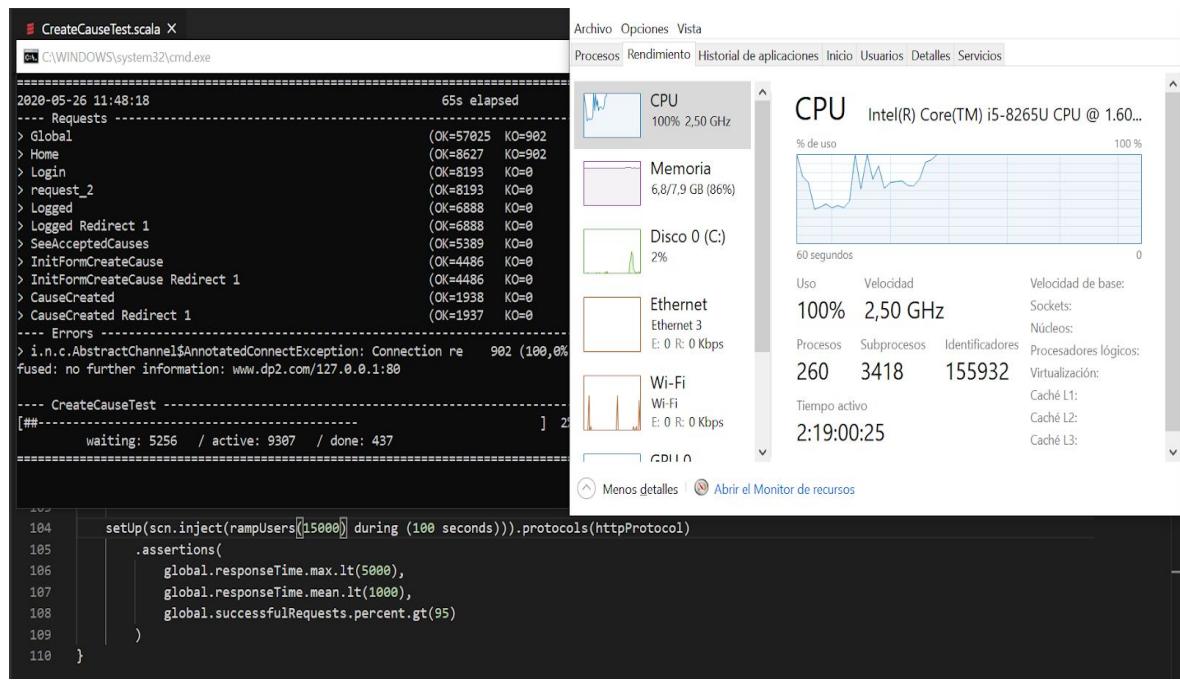
- Aquí vemos de manera más detallada que los “assertions” que hemos incluido (que el tiempo máximo de respuesta sea menos de 5 segundos, que el tiempo medio de respuesta sea de 1 segundo y que el 95% de los usuarios entren correctamente) ha sido realizado con éxito.



Mínimo número de usuarios concurrentes con los que el sistema no funciona adecuadamente

Llegamos a este mínimo probando con 15.000 usuarios concurrentes en 100 segundos. Los resultados obtenidos son:

- Ejecutando la herramienta de gatling, vemos que empiezan a aparecer fallos además de el estado de la CPU al 100% durante gran parte del proceso.



- Aquí podemos ver de forma más detallada que no superamos que el tiempo máximo de solicitud sea de 5 segundos además de que empiezan a haber ciertos usuarios afectados que no consiguen realizar ciertas acciones.

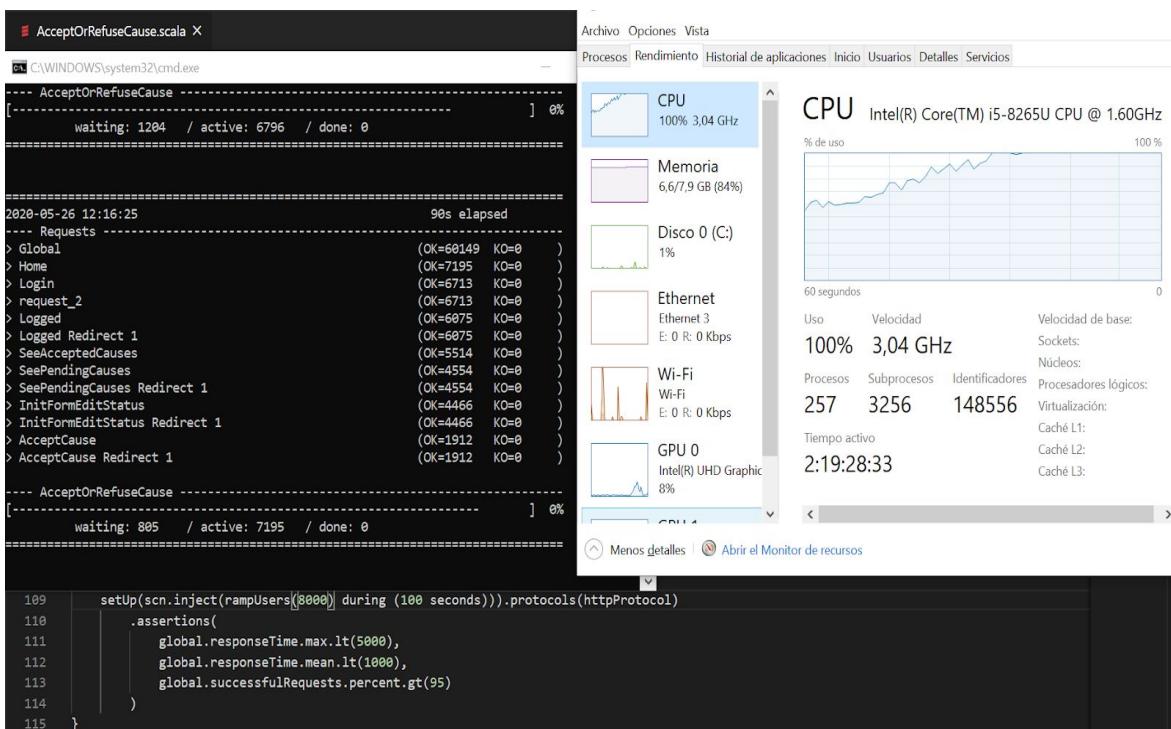


Historia de Usuario 2	Yo COMO admin QUIERO poder aceptar o rechazar causas benéficas PARA evitar fraudes
Escenario	Admin acepta y deniega causas pendientes
Nombre de prueba	acceptOrRefuseCause

Máximo número de usuarios en los que el sistema funciona correctamente

Estos son los resultados obtenidos al realizar el test de rendimiento con 8.000 usuarios concurrentes en 100 segundos:

- Como podemos observar, la CPU está al 100% pero no da problemas en cuanto al funcionamiento del sistema y funciona correctamente.



- Aquí podemos ver de manera más detallada que el sistema funciona correctamente cumpliendo nuestras condiciones.

more features with Gatling FrontLine

acceptorrefusecause

2020-05-26 12:14:54 +02:00, duration : 203 seconds

DETAILS

> Global Information

Indicators

Number of requests

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

STATISTICS

Expand all groups | Collapse all groups

Mínimo número de usuarios concurrentes con los que el sistema no funciona adecuadamente

Probando nuestro sistema con 8.500 resultados llegamos a este mínimo en el que el sistema empieza a no funcionar adecuadamente. Estos son los resultados obtenidos:

- Cómo podemos ver ocurre un fallo al ejecutar gatling además de que la CPU está al 100%.

AcceptOrRefuseCause.scala

```

2020-05-26 12:10:15          120s elapsed
--- Requests ---
> Global                           (OK=98165 KO=208  )
> Home                            (OK=8292 KO=208  )
> Login                            (OK=8500 KO=0   )
> request_2                         (OK=8500 KO=0   )
> Logged                           (OK=8500 KO=0   )
> Logged Redirect 1                (OK=8500 KO=0   )
> SeeAcceptedCauses                (OK=8192 KO=0   )
> SeePendingCauses                 (OK=7387 KO=0   )
> SeePendingCauses Redirect 1      (OK=7386 KO=0   )
> InitFormEditStatus               (OK=9844 KO=0   )
> InitFormEditStatus Redirect 1    (OK=9844 KO=0   )
> AcceptCause                       (OK=4588 KO=0   )
> AcceptCause Redirect 1           (OK=4588 KO=0   )
> RejectCause                      (OK=2031 KO=0   )
> RejectCause Redirect 1           (OK=2029 KO=0   )
--- Errors ---
> i.n.c.AbstractChannel$AnnotatedConnectException: Connection re  208 (100,%
fused: no further information: www.dp2.com/127.0.0.1:80
---- AcceptOrRefuseCause -----
[#####] 17%
    waiting: 0      / active: 6983     / done: 1517
====

109 setUp scn.inject(rampUsers[8500] during (100 seconds)).protocols(httpProtocol)
110   .assertions(
111     global.responseTime.max.lt(5000),
112     global.responseTime.mean.lt(1000),
113     global.successfulRequests.percent.gt(95)
114   )
115 }

```

CPU Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz

% uso

60 segundos

Uso: 99% | Velocidad: 3,07 GHz | Velocidad de base: 1.60GHz

Procesos: 253 | Subprocesos: 3305 | Identificadores: 146209

Tiempo activo: 2:19:22:20

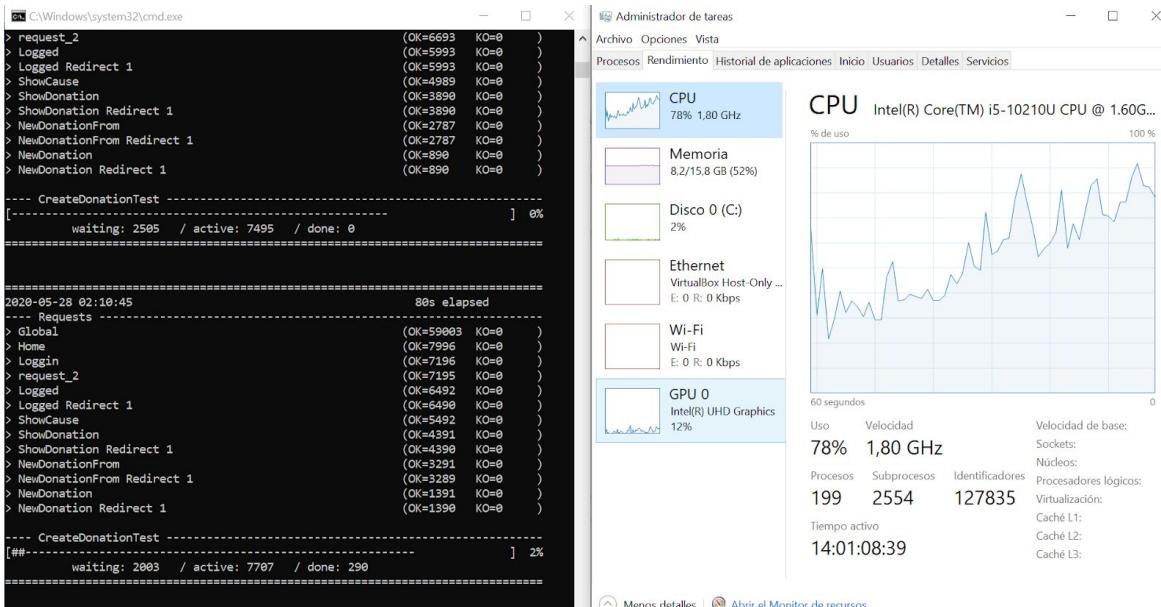
Núcleos: 4 | Procesadores lógicos: 4 | Virtualización: Off

Caché L1: 16 KB | Caché L2: 1 MB | Caché L3: 8 MB

Monitoreo de recursos

- De manera más detallada podemos ver que el sistema empieza a no cumplir ciertos aspectos que hacen que no su funcionalidad ya no sea la óptima para los distintos usuarios.

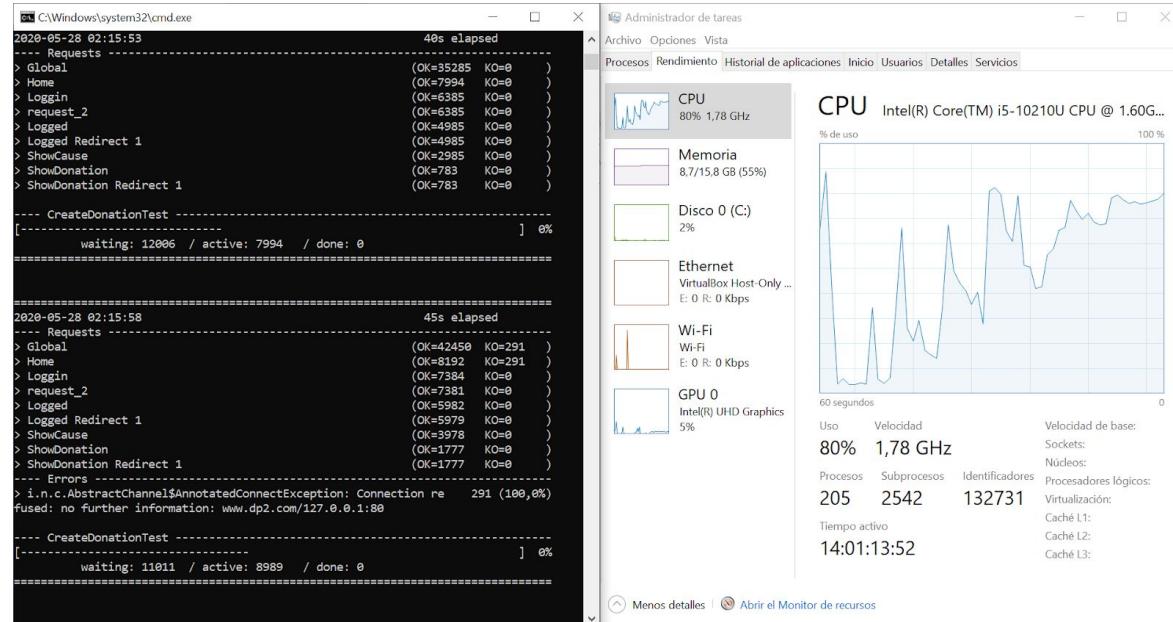


Historia de Usuario 3	Yo COMO usuario QUIERO poder realizar donaciones a causas benéficas PARA apoyar las causas.
Escenario	Un admin crea una donación.
Nombre de Prueba	CreateDonationTest
Máximo número de usuarios en los que el sistema funciona correctamente	
<p>Probando con 10000 usuarios en 100 segundos podemos ver que sobre los 7700 la CPU empieza a dar picos cercanos al 80% .</p> <p>- El sistema funciona correctamente con unos 8000 usuarios aún teniendo el uso de la CPU por 80%. Cabe destacar que si aumenta por encima del 85% en mi servidor empieza a fallar, nunca llega al 100%.</p>  <p>The terminal window shows the Gatling test results for 'CreateDonationTest'. It lists various requests and their outcomes (OK or KO). The test runs for 80 seconds, starting at 2020-05-28 02:10:45. The last section shows the test is still running with 2% completion. The Task Manager shows the CPU usage for an Intel Core i5-10210U CPU at 1.60GHz. The CPU usage fluctuates between 78% and 82%, peaking around 80% and staying below 100% throughout the 100-second period.</p>	
<p>- Como vemos en la imagen de Gatling los 3 assertion se cumplen correctamente.</p>  <p>The Gatling report shows the following assertions:</p> <ul style="list-style-type: none"> 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 	

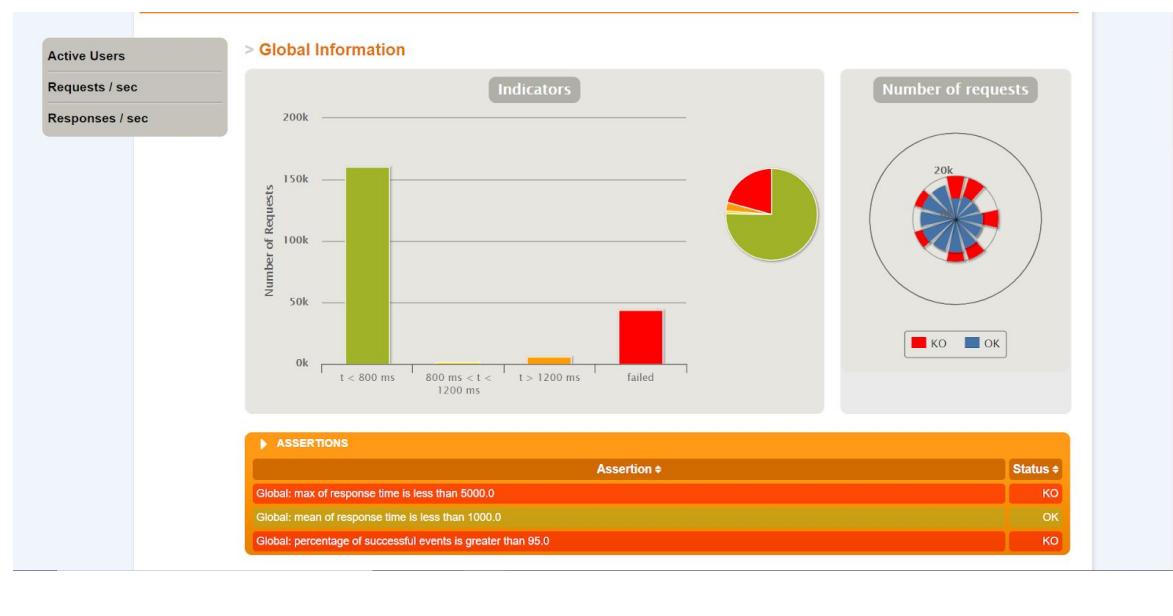
Mínimo número de usuarios concurrentes con los que el sistema no funciona adecuadamente

El mínimo se obtiene probando 20000 usuarios concurrentes en 100 segundos.

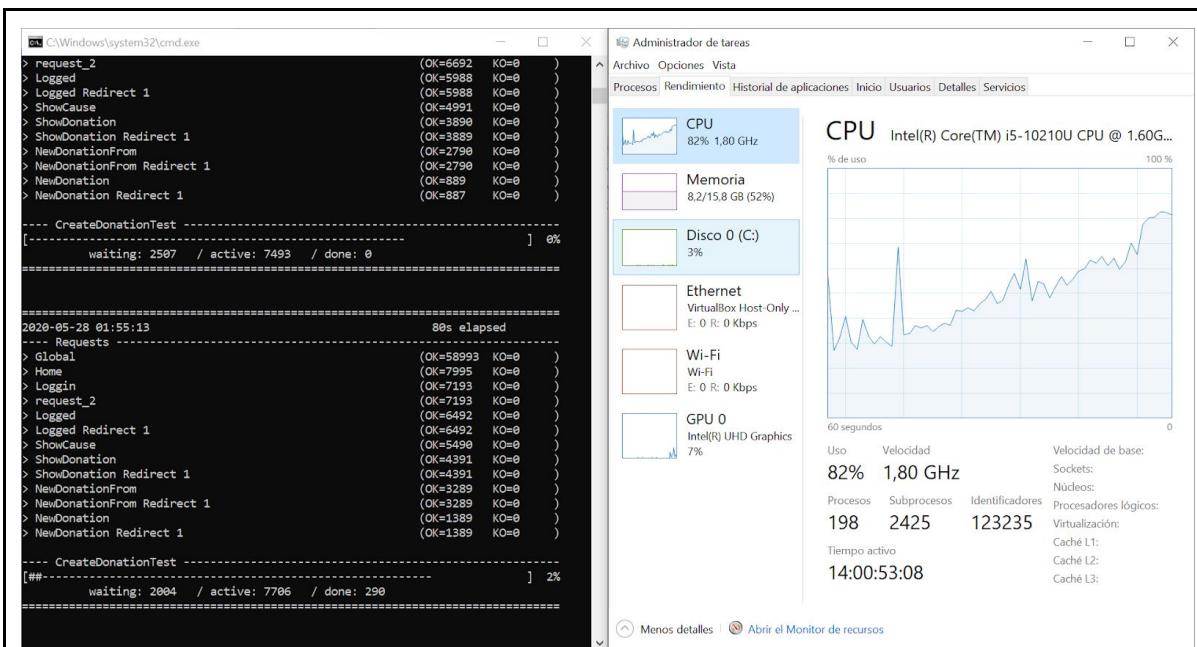
- Viendo esta captura nos podemos dar cuenta que aunque la CPU no esté al 100% y mucho menos la gráfica ya empezamos a ver los primeros KO sobre los 9000 usuarios concurrentes.



- En esta vista podemos ver que menos el tiempo medio todos los demás Assertion fallan, por tanto podemos decir que este sistema no funciona adecuadamente.



Historia de Usuario 4	Yo COMO usuario QUIERO poder realizar donaciones anónimas a causas benéficas PARA apoyar las causas.
Escenario	Un admin crea una donación anónima.
Nombre de Prueba	CreateDonationTest
Máximo número de usuarios en los que el sistema funciona correctamente	
Esta historia de usuario es prácticamente idéntica a la anterior exceptuando que se hará de forma anónima, por tanto podemos reciclar los test cambiando el valor de Anonymous a 'true', ya que antes realizamos las pruebas con este valor en false. Aun así, los resultados deben ser los mismos ya que este cambio no supone un aumento o retraso del rendimiento.	
<pre>object NewDonation{ val newDonation=exec(http("NewDonation") .post("/cause/3/donations/new") .headers(headers_3) .formParam("money", "1000") .formParam("anonymous", "false") .formParam("id", "") .formParam("_csrf", "651bd1a6-8ffb-44bc-9420-f73ea91a8449")) .pause(11) }</pre>	
En la captura vemos el valor en 'false' el cual cambiaremos y lo pondremos a 'true'.	
Probando con 10000 usuarios en 100 segundos podemos ver que sobre los 7700 la CPU empieza a dar picos cercanos al 80% .	
<ul style="list-style-type: none"> - Volvemos a ver que aunque ahora las donaciones sean anónimas el sistema sigue funcionando correctamente con los valores de 10000 usuarios en 100 segundos. 	



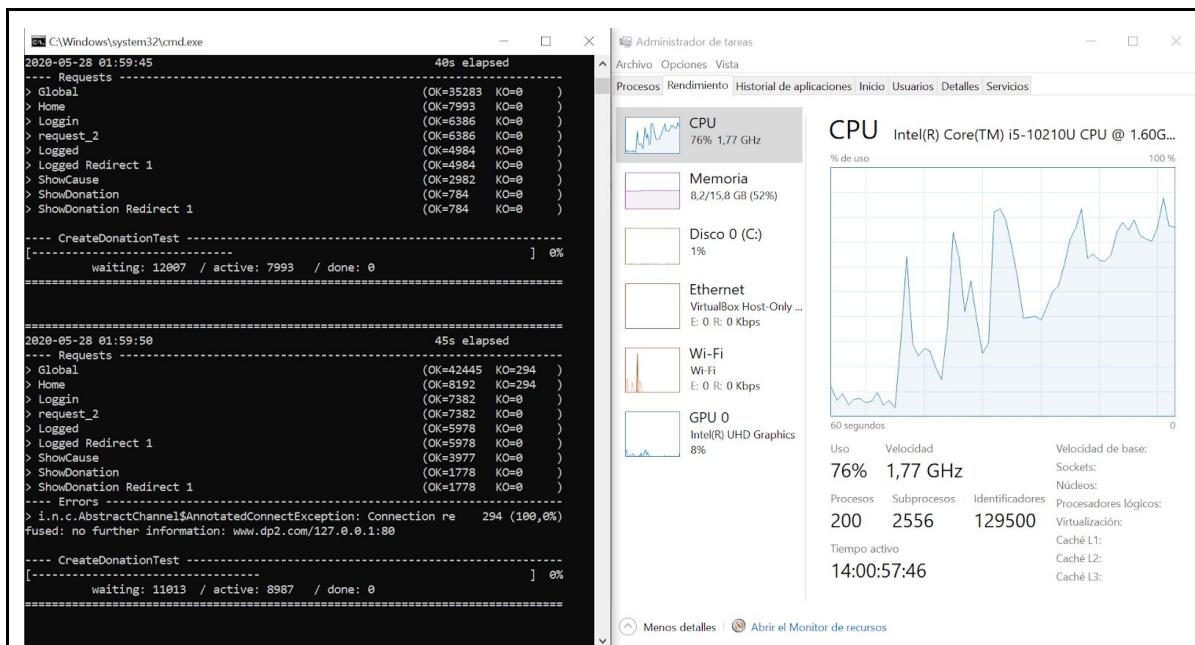
- En gatling corroboramos que los assertions se cumplen, por tanto, el sistema funciona correctamente.



Mínimo número de usuarios concurrentes con los que el sistema no funciona adecuadamente

El mínimo se obtiene probando 20000 usuarios concurrentes en 100 segundos.

- Como era de esperar, sobre los 9000 usuarios el sistema empieza a fallar y obtenemos los primeros KO, acercando el rendimiento al 80%.



- Los assertion no se cumplen por tanto podemos decir que con estos datos el sistema deja de funcionar como deseamos.

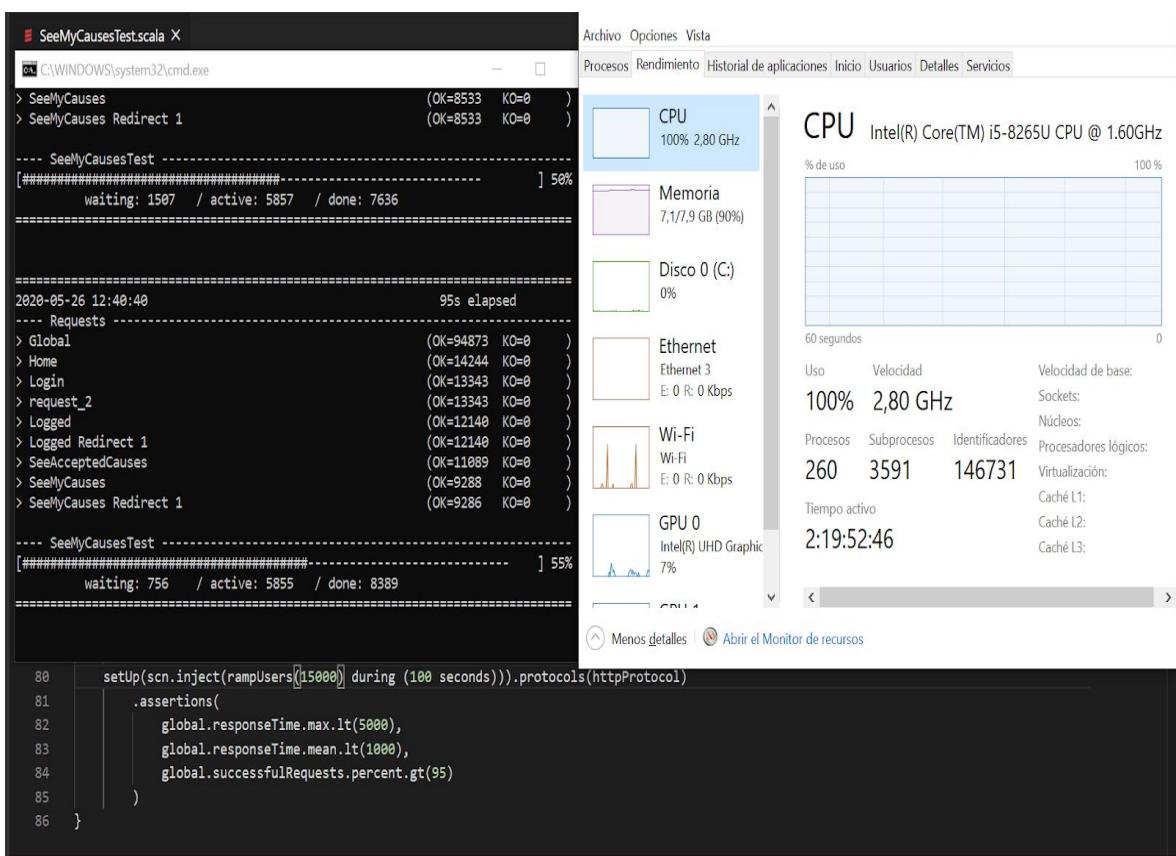


Historia de Usuario 5	Yo COMO usuario QUIERO ver mi lista de causas y el estado en que se encuentran estas PARA ver si han sido aceptadas, rechazadas o pendientes.
Escenario	Admin ve las causas que ha creado.
Nombre prueba	seeMyCausesTest

Máximo número de usuarios en los que el sistema funciona correctamente

Estos son los resultados obtenidos al realizar el test de rendimiento con 15.000 usuarios concurrentes en 100 segundos:

- Cómo podemos ver no hay errores a la hora de ejecutar la aplicación de gatling.



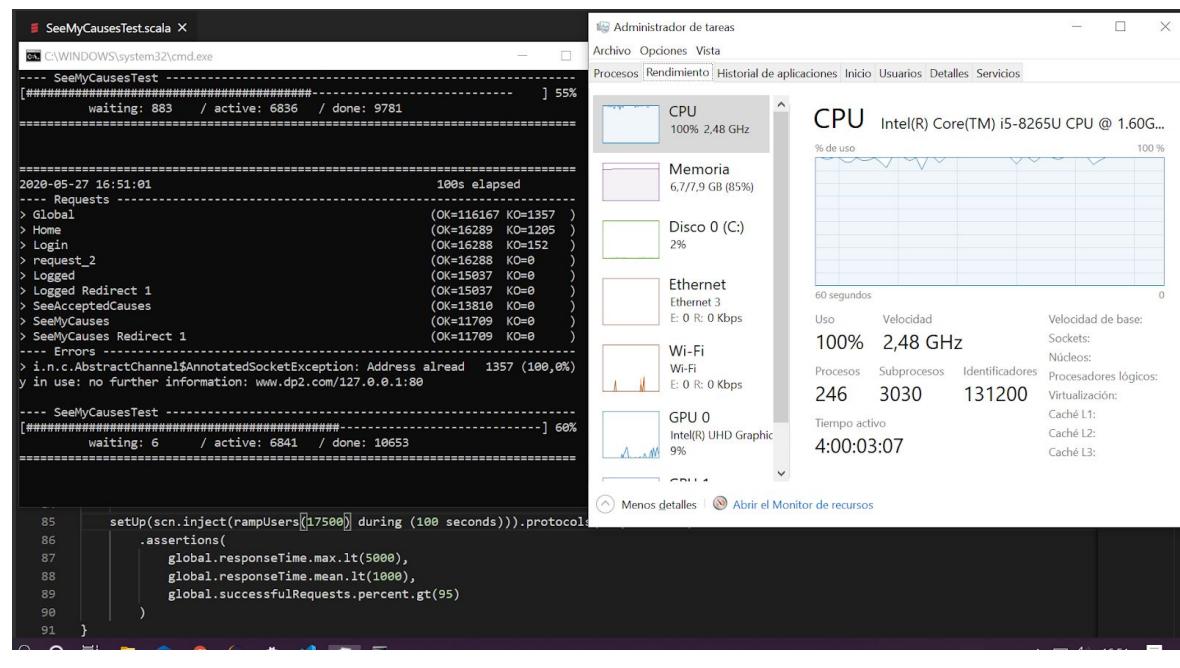
- Aquí vemos más detalladamente que el sistema funciona correctamente sin fallos algunos que perjudiquen la funcionalidad.



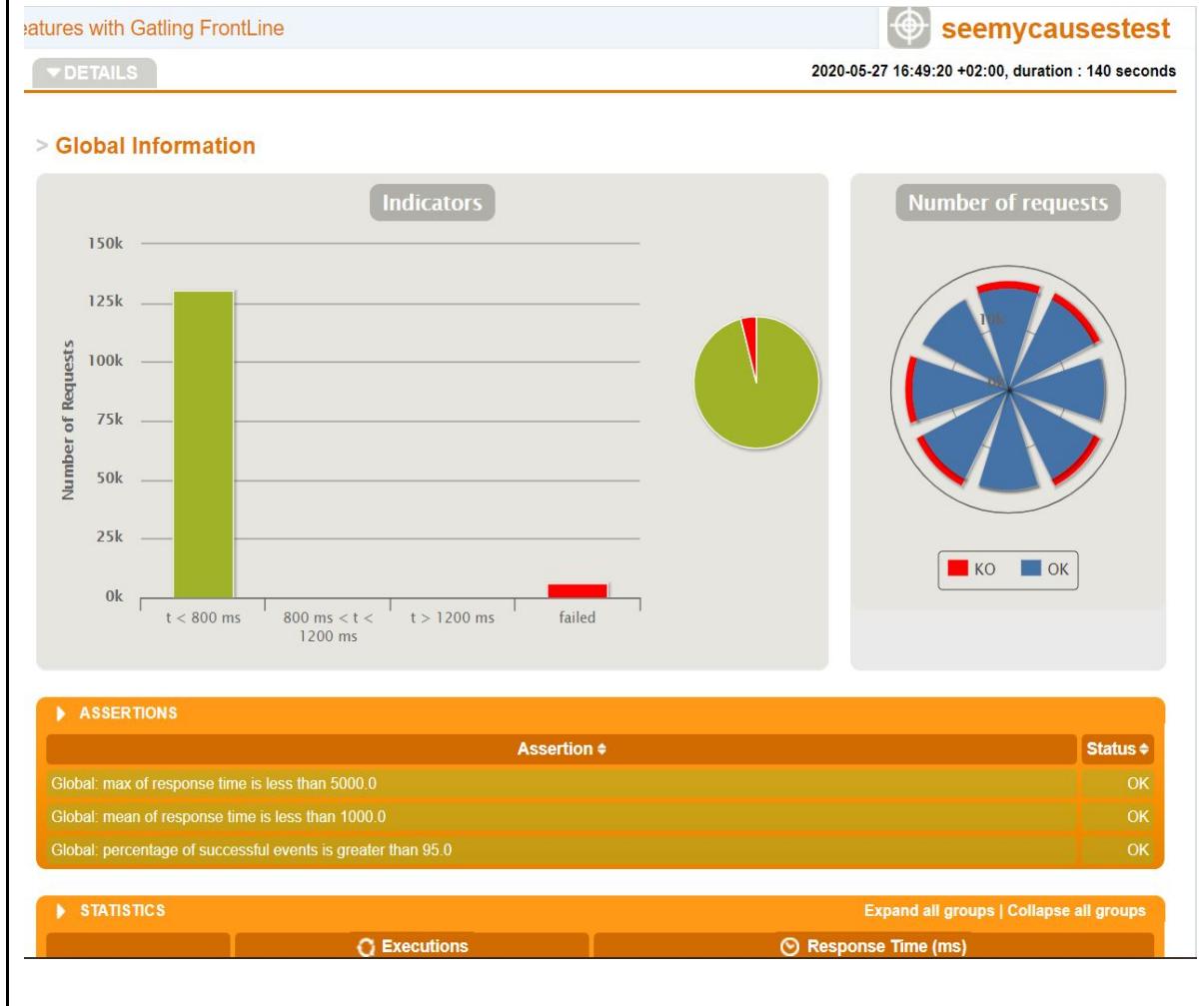
Mínimo número de usuarios concurrentes con los que el sistema no funciona adecuadamente

Estos son los resultados obtenidos al realizar el test de rendimiento con 17.500 usuarios concurrentes en 100 segundos:

- Aquí podemos ver que al usar gatling empiezan a saltar fallos y la CPU asciende al 100%.



- En esta foto podemos ver más detalladamente que aunque se cumplen las condiciones que hemos establecido para evaluar el sistema, hay usuarios que no son capaces de realizar ciertas acciones.

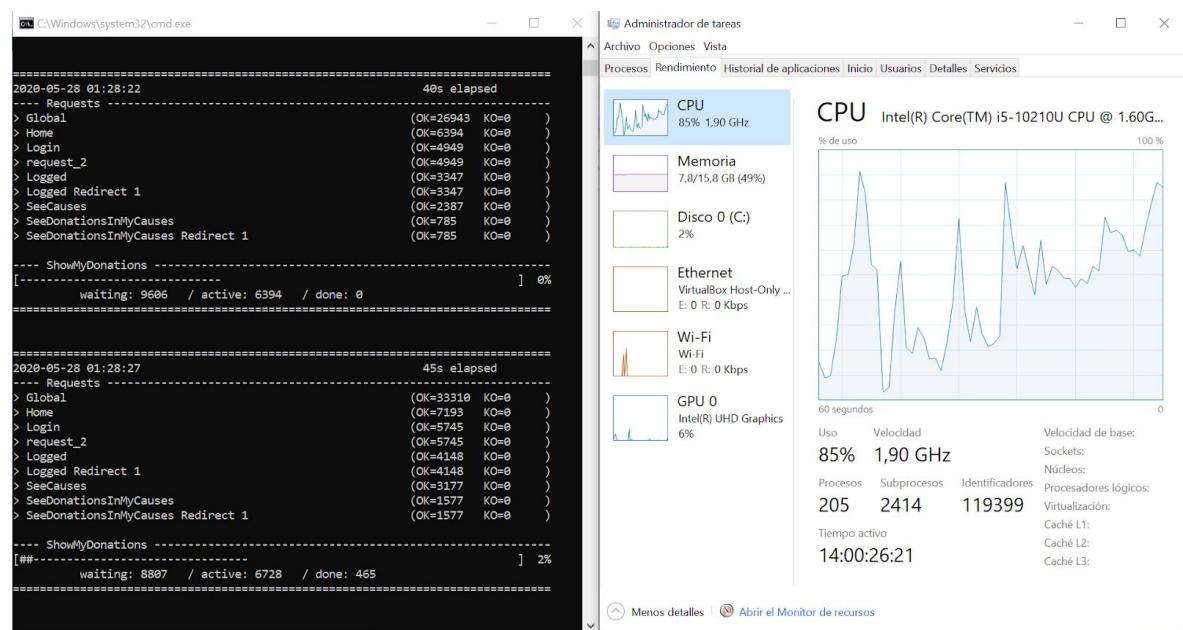


Historia de Usuario 6	Yo COMO usuario QUIERO ver la lista de donaciones de mis causas propuestas PARA llevar un seguimiento de la evolución de mis causas.
Escenario	Un Owner revisa las donaciones realizadas en sus causas.
Nombre de Prueba	ShowMyDonations

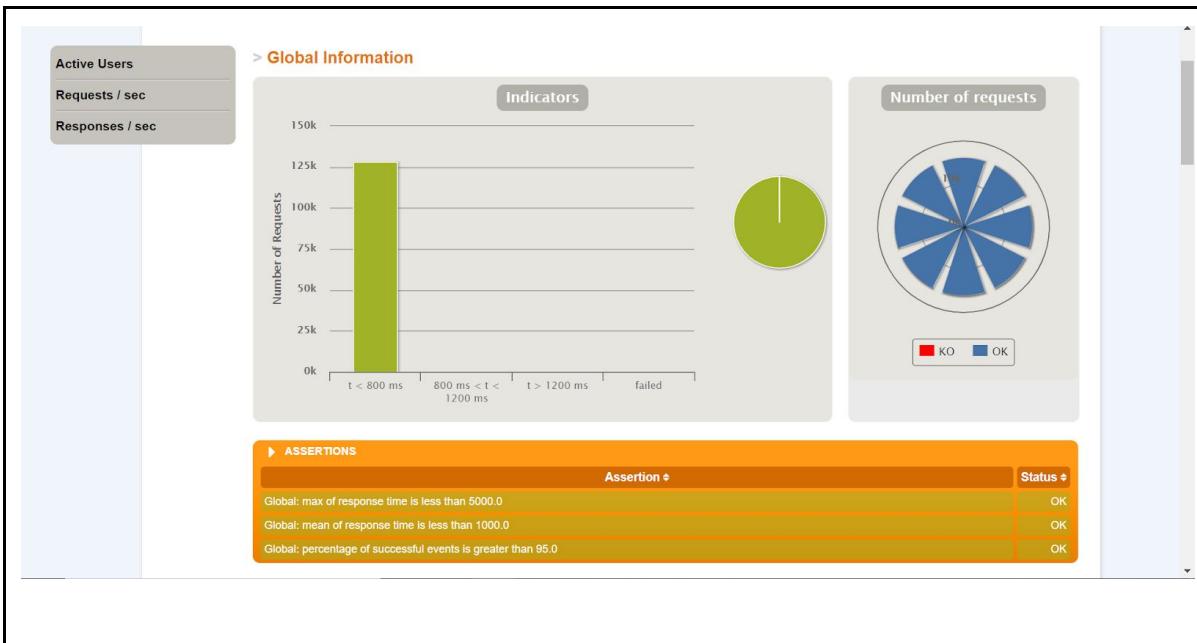
Máximo número de usuarios en los que el sistema funciona correctamente

Probando con 16000 usuarios en 100 segundos podemos ver que sobre los 7000 la CPU empieza a dar picos cercanos al 90% .

- El sistema empieza a saturarse con 7000 usuarios activos y se acerca al 85% aunque luego dará picos más altos sobre los 9000 usuarios.



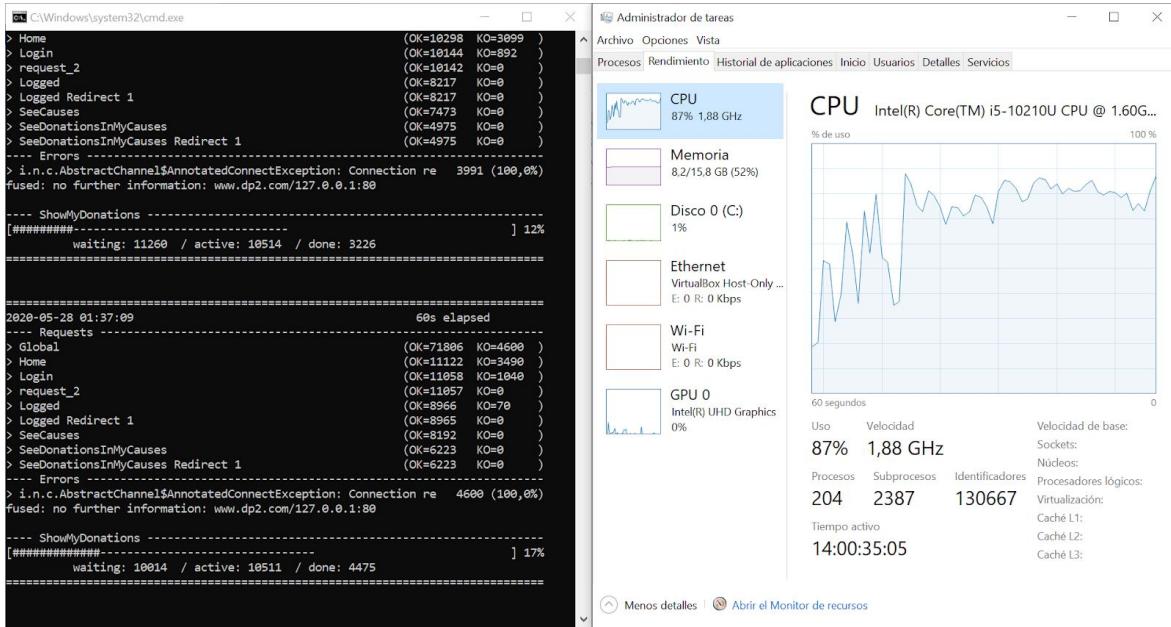
-En gatling vemos que aunque la CPU se acerque al 100% no hay ningún assertion fallido y todas las peticiones se han realizado satisfactoriamente.



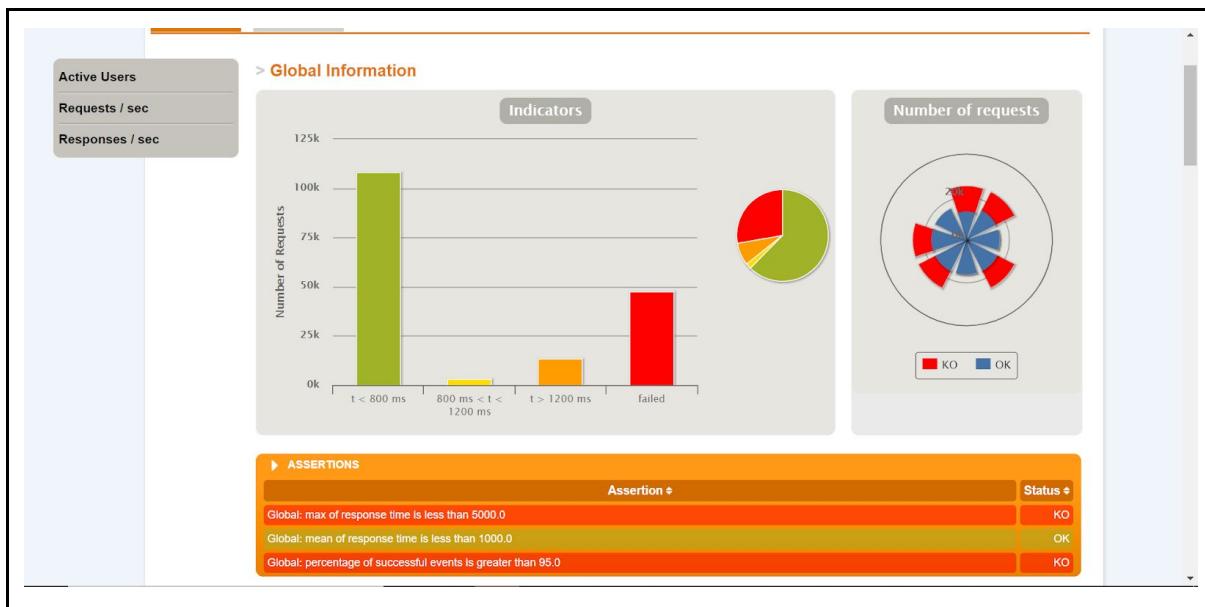
Mínimo número de usuarios concurrentes con los que el sistema no funciona adecuadamente

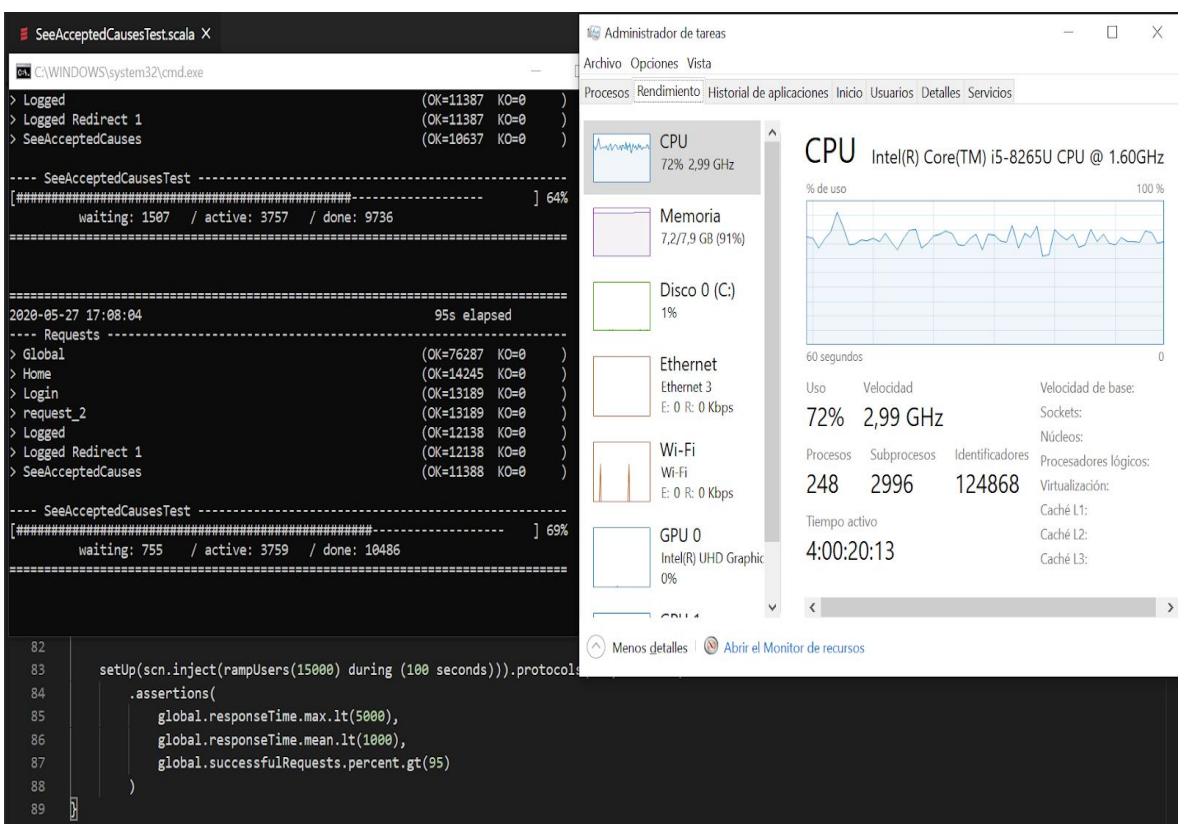
El mínimo se obtiene probando 25000 usuarios concurrentes en 100 segundos.

- Podemos observar que al llegar a los 10000 usuarios concurrentes el sistema comienza a lanzar excepciones y hay algunas peticiones que no pueden ser atendidas correctamente.

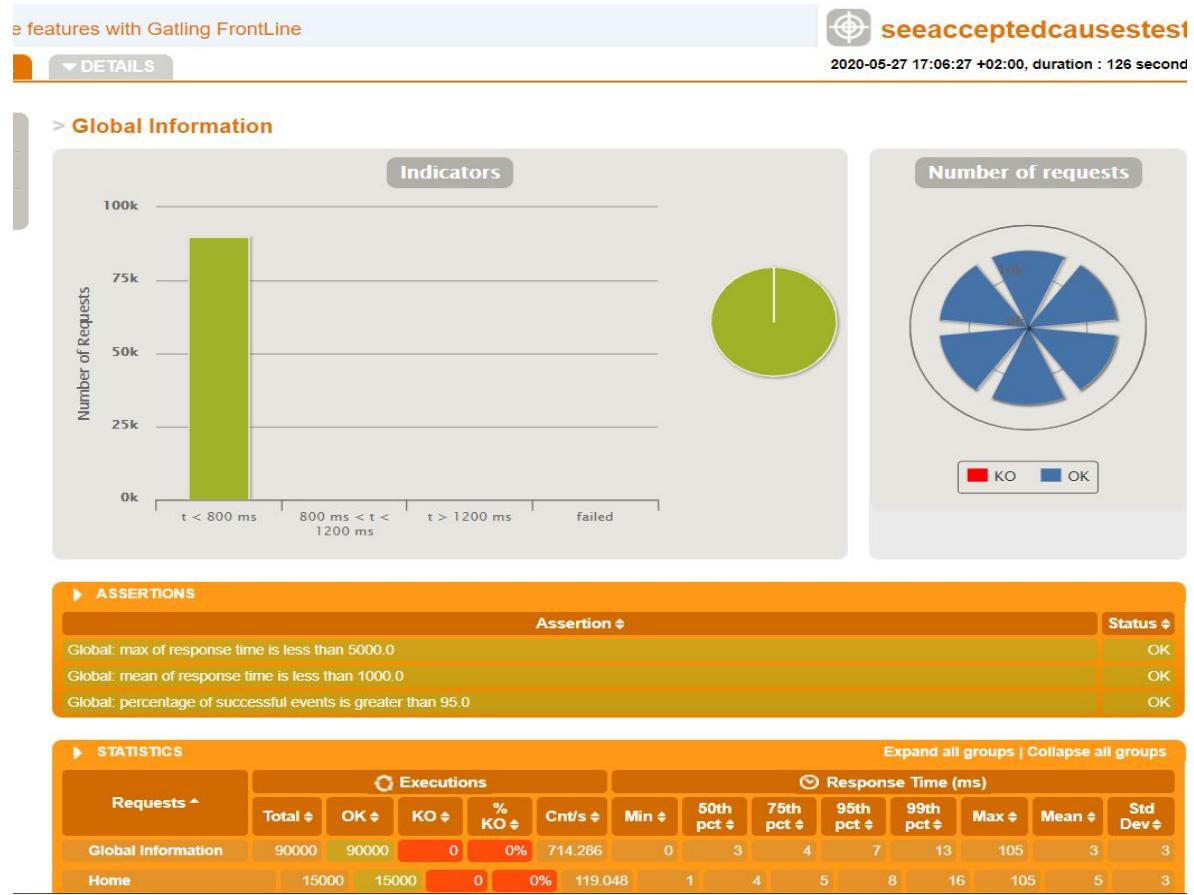


- En gatling corroboramos que el sistema falla tanto en el tiempo máximo para obtener una respuesta como en el porcentaje de peticiones exitosas.



Historia de Usuario 7	Yo COMO usuario QUIERO ver la lista de causas PARA poder realizar mis donaciones.
Escenario	Un admin entra en el apartado de causas del sistema
Nombre de prueba	seeAcceptedCausesTest
Máximo número de usuarios en los que el sistema funciona correctamente	
Estos son los resultados obtenidos al realizar el test de rendimiento con 20.000 usuarios concurrentes en 100 segundos:	
<p>- Aquí podemos ver que salta un error a la hora de ejecutar Gatling.</p>  <pre>SeeAcceptedCausesTest.scala X C:\WINDOWS\system32\cmd.exe > Logged (OK=11387 KO=0) > Logged Redirect 1 (OK=11387 KO=0) > SeeAcceptedCauses (OK=10637 KO=0) --- SeeAcceptedCausesTest --- [##### waiting: 1507 / active: 3757 / done: 9736] 64% ===== 2020-05-27 17:08:04 95s elapsed --- Requests --- > Global (OK=76287 KO=0) > Home (OK=14245 KO=0) > Login (OK=13189 KO=0) > request_2 (OK=13189 KO=0) > Logged (OK=12158 KO=0) > Logged Redirect 1 (OK=12138 KO=0) > SeeAcceptedCauses (OK=11388 KO=0) --- SeeAcceptedCausesTest --- [##### waiting: 755 / active: 3759 / done: 10486] 69%</pre> <p>Administrator de tareas</p> <p>CPU: 72% 2,99 GHz</p> <p>Memoria: 7,2/7,9 GB (91%)</p> <p>Disco 0 (C): 1%</p> <p>Ethernet: Ethernet 3 E: 0 R: 0 Kbps</p> <p>Wi-Fi: Wi-Fi E: 0 R: 0 Kbps</p> <p>GPU 0: Intel(R) UHD Graphic 9%</p> <p>Uso: 72% Velocidad: 2,99 GHz Tiempo activo: 4:00:20:13</p> <p>Velocidad de base: Sockets: Núcleos: Procesadores lógicos: Virtualización: Caché L1: Caché L2: Caché L3:</p>	

- De manera más detallada, vemos que no se cumplen una de las condiciones por lo que el sistema no funcionaría correctamente.



Mínimo número de usuarios concurrentes con los que el sistema no funciona adecuadamente

Estos son los resultados obtenidos al realizar el test de rendimiento con 20.000 usuarios concurrentes en 100 segundos:

- Aquí podemos ver que salta un error a la hora de ejecutar Gatling.

The terminal window displays the execution of a Scala test named 'SeeAcceptedCausesTest'. The test includes assertions for response times and success rates. The Administrator tasks window shows CPU usage for an Intel i5-8265U CPU at 1.60GHz, with a graph and detailed statistics.

```

# SeeAcceptedCausesTest.scala
C:\WINDOWS\system32\cmd.exe
> SeeAcceptedCauses
(OK=12184 KO=0)

--- SeeAcceptedCausesTest
[##### waiting: 4009 / active: 5014 / done: 10977 ] 54%
=====
2020-05-27 17:14:40
-- Requests --
> Global (OK=89025 KO=706 )
> Home (OK=16290 KO=706 )
> Login (OK=15589 KO=6 )
> request_2 (OK=15589 KO=0 )
> Logged (OK=14187 KO=0 )
> Logged Redirect 1 (OK=14187 KO=0 )
> SeeAcceptedCauses (OK=13183 KO=0 )

--- Errors --
i.n.c.AbstractChannel$AnnotatedSocketException: Address already in use: no further information: www.dp2.com/127.0.0.1:80
y [##### waiting: 3004 / active: 5017 / done: 11979 ] 59%
=====

83     setUp(scn.inject(rampUsers(20000) during (100 seconds))).protocol,
84     .assertions(
85       global.responseTime.max.lt(5000),
86       global.responseTime.mean.lt(1000),
87       global.successfulRequests.percent.gt(95)
88     )
89 }

CPU Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz
% de uso 100 %
60 segundos
0
Uso 100% Velocidad 3,37 GHz
Procesos 245 Subprocesos 3078 Identificadores 127396
Tiempo activo 4:00:26:66
Velocidad de base: Sockets:
Núcleos: Procesadores lógicos:
Virtualización: Caché L1:
Caché L2:
Caché L3:

```

- De manera más detallada, vemos que no se cumplen una de las condiciones por lo que el sistema no funcionaría correctamente.



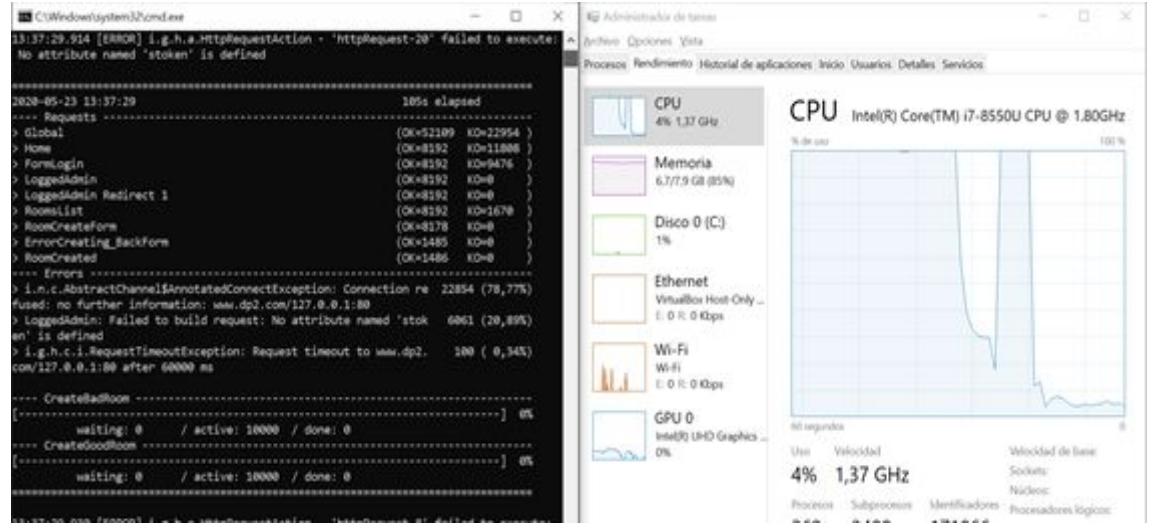
Historia de Usuario 8	Yo COMO admin QUIERO que se muestre a los usuarios solo las causas que he aceptado PARA evitar confusiones con causas que han sido rechazadas o pendientes.
Leyendo la historia de usuario 7, esta historia en cuanto a su test se refiere sería igual, el funcionamiento del test es idéntico dado que evalúan el mismo punto ambos. Ambos recorren el mismo “camino” dentro del sistema por lo que sería repetitiva esta prueba de rendimiento. Por ello, los valores de la historia de usuario 7 se darían también en este caso.	

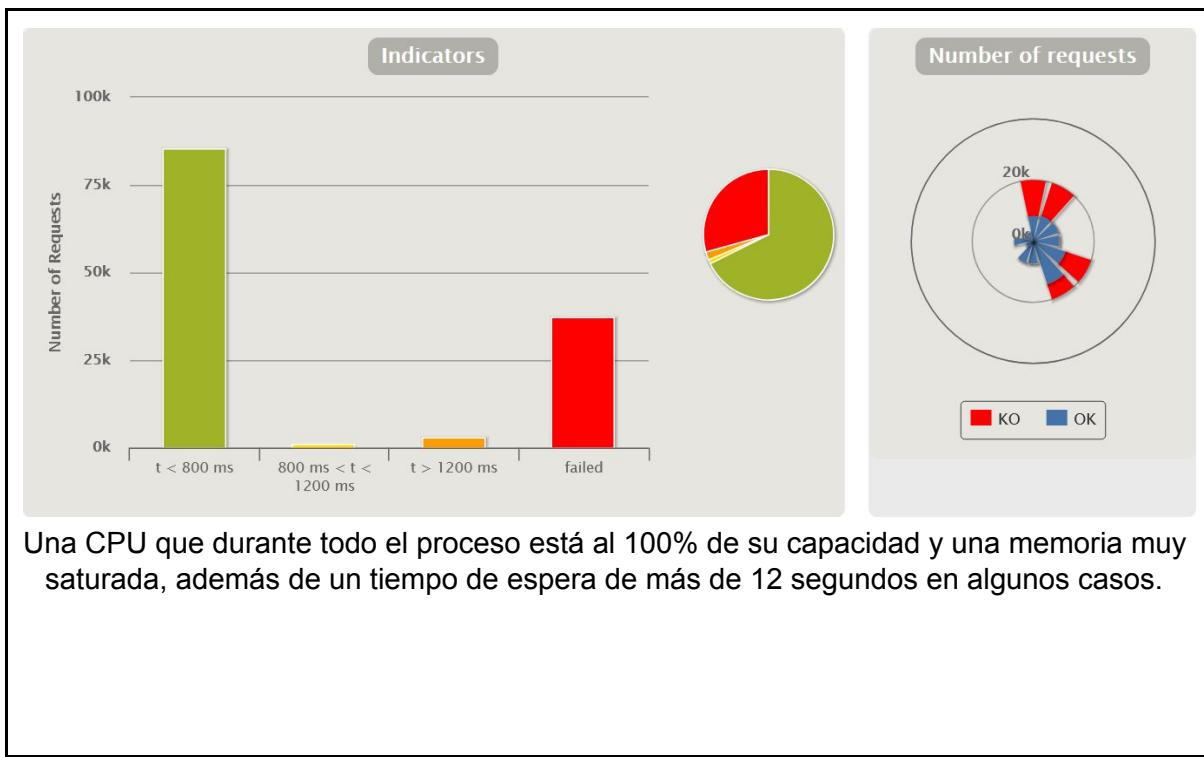
Nota respecto a la Historia de Usuario 9: A la hora de hacer las pruebas de rendimiento, nos hemos dado cuenta de que la Historia de Usuario 9 no se puede testear dado que no sería una historia de usuario como tal sino más bien una Regla de Negocio. Por ello, aunque nos hemos dado cuenta demasiado tarde, hemos decidido quitarla de nuestra lista de historias de usuario.

Historia de Usuario 10	Yo COMO admin QUIERO poder crear salas PARA que las mascotas tengan un lugar donde quedarse en caso de que el dueño no puede hacerse cargo de las mascotas por diferentes motivos.
Escenario 1	Un admin crear una sala válida.
Escenario 2	Un admin crea una sala ya existente.
Script	CreateRoomAdminTwoScenariosDiagnosis

Mínimo número de usuarios concurrentes que no es soportado por el sistema

Con un número de 10000 usuarios en 100 segundos los resultados son:





Una CPU que durante todo el proceso está al 100% de su capacidad y una memoria muy saturada, además de un tiempo de espera de más de 12 segundos en algunos casos.

STATISTICS													Expand all groups Collapse all groups			
Requests \downarrow	Executions					Response Time (ms)										
	Total \pm	OK \pm	KO \pm	% KO \pm	Cnt/s \pm	Min \pm	50th pct \pm	75th pct \pm	95th pct \pm	99th pct \pm	Max \pm	Mean \pm	Std Dev \pm			
Global Information	126881	89550	37331	29%	456.408	0	17	2018	2194	2441	60025	749	2000			
Home	20000	8192	11808	59%	71.942	1	2014	2029	2133	2339	60025	1500	4267			
FormLogin	20000	8604	11396	57%	71.942	0	2014	2031	2172	2388	10088	1222	1061			
LoggedAdmin	8604	8604	0	0%	30.95	2	6	10	27	144	384	12	26			
LoggedAd...direct 1	8604	8604	0	0%	30.95	1	7	11	25	81	359	11	18			
RoomsList	20000	11067	8933	45%	71.942	5	1402	2054	2339	2969	21737	1169	1414			
RoomCreateForm	20000	14806	5194	26%	71.942	1	31	2025	2296	2796	21950	818	1353			
ErrorCre...BackForm	7387	7387	0	0%	26.572	1	13	25	100	233	549	26	44			
RoomCreated	7419	7419	0	0%	26.687	1	13	25	96	228	573	26	43			
RoomsLis...direct 1	2463	2463	0	0%	8.86	0	22	40	76	171	346	29	33			
RoomCrea...direct 1	6202	6202	0	0%	22.309	0	14	32	74	136	358	23	31			
ErrorCre...direct 1	3090	3090	0	0%	11.115	0	4	7	23	51	94	7	9			
RoomCrea...direct 1	3112	3112	0	0%	11.194	0	4	7	22	52	127	6	9			

Se observa el porcentaje de KO del 29% y un tiempo máximo en respuestas de hasta 60 segundos en algunos casos aunque la media máxima es de 1,5 seg.

Si observamos el número de respuestas por segundo vemos que a partir de los casi 9000 usuarios (8998) empiezan los KO por lo tanto haremos una segunda prueba con 4000 Usuarios, para comparar.

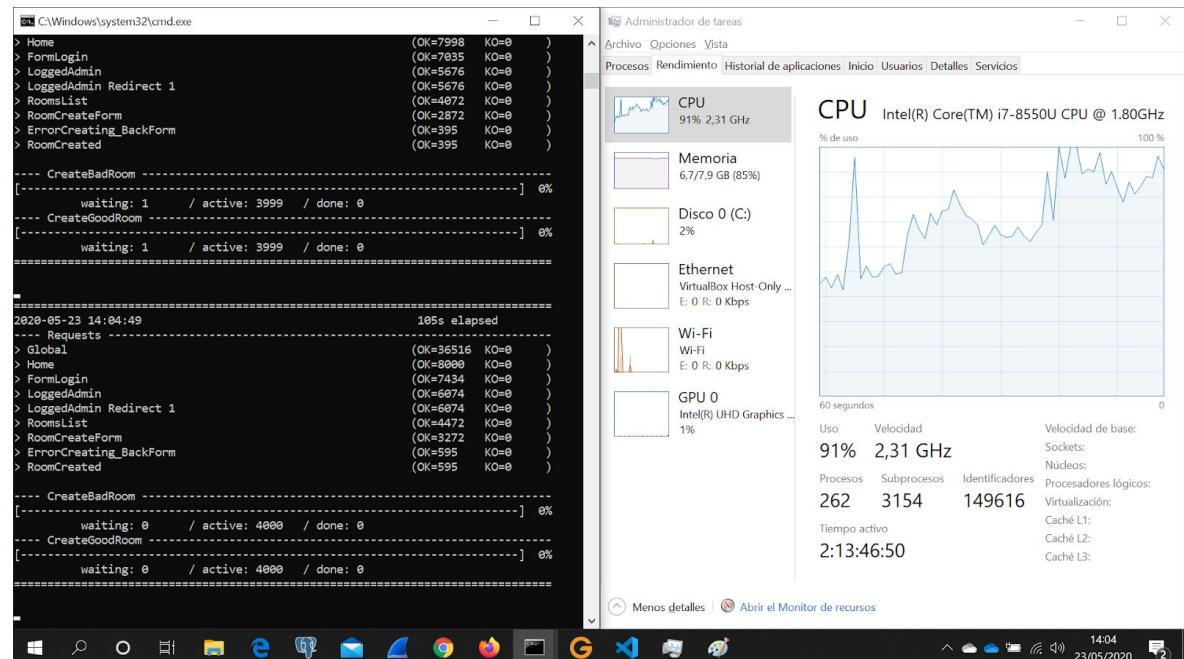
ASSERTIONS		Assertion \downarrow	Status \downarrow
Global: max of response time is less than 5000.0			KO
Global: mean of response time is less than 1000.0			OK
Global: percentage of successful events is greater than 95.0			KO

En cuanto a los Assert vemos que solo se cumplen 1 de los 3 establecidos.

Máximo número de usuarios que soporta teniendo un buen rendimiento

Con 4000 Usuarios concurrentes por cada escenario durante 100 segundos.

CPU + GATLIN



Cuando alcanza los 4000 usuarios concurrentes por escenario la CPU comienza a rondar el 90 por cien de su capacidad e incluso llega a picos del 100%.

> Global Information



Vemos como todas las request se han completado con éxito en menos de 800 milisegundos.

▶ 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

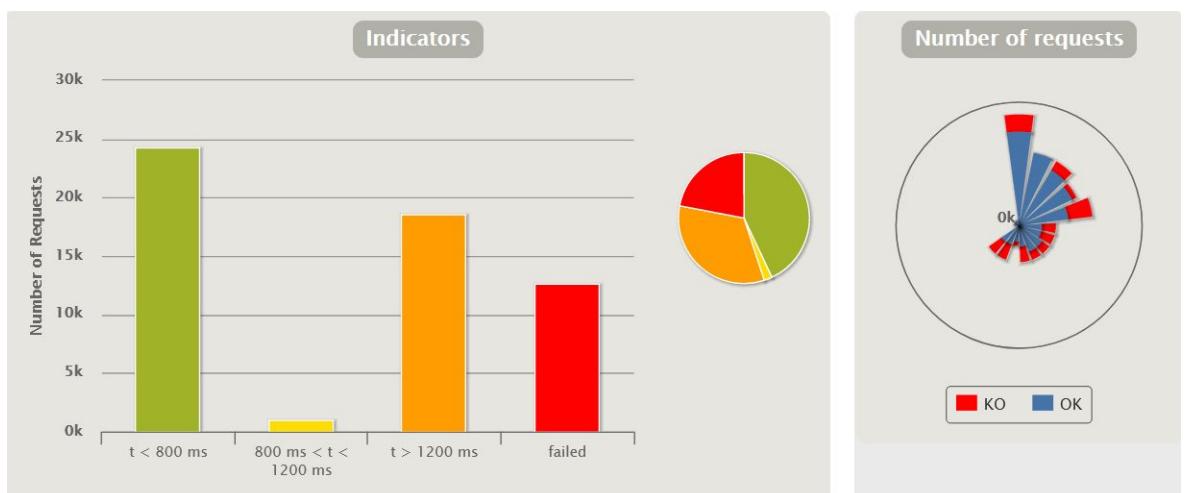
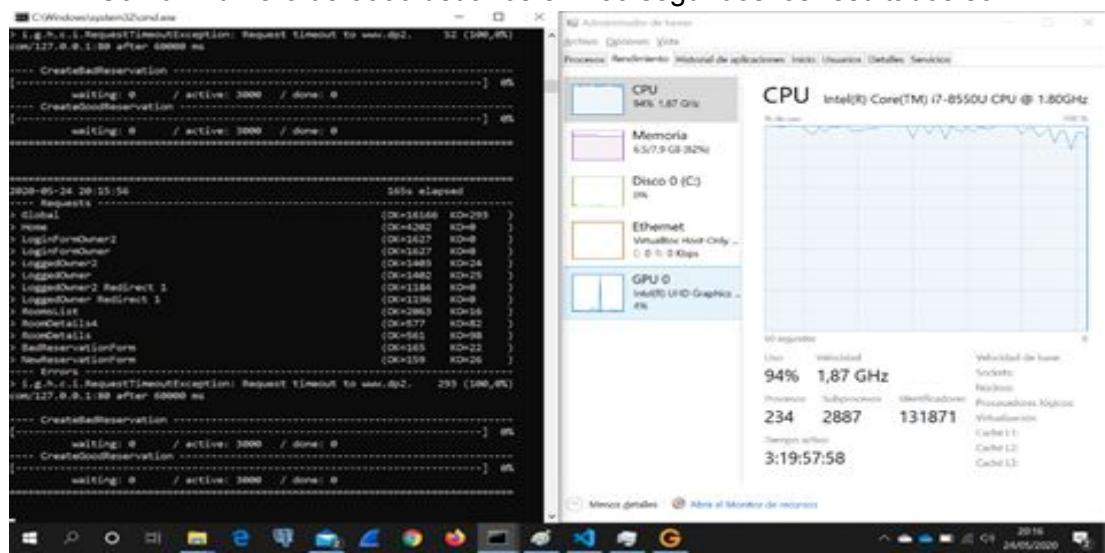
▶ STATISTICS													
Requests ↴	⌚ Executions					⌚ Response Time (ms)							
	Total ↴	OK ↴	KO ↴	% KO ↴	Cnt/s ↴	Min ↴	50th pct ↴	75th pct ↴	95th pct ↴	99th pct ↴	Max ↴	Mean ↴	Std Dev ↴
Global Information	56000	56000	0	0%	215.385	0	7	11	23	32	107	9	7
Home	8000	8000	0	0%	30.769	2	6	9	16	25	107	7	5
FormLogin	8000	8000	0	0%	30.769	0	3	4	9	15	45	4	3
LoggedAdmin	8000	8000	0	0%	30.769	1	5	7	13	20	55	6	4
LoggedAd...direct 1	8000	8000	0	0%	30.769	1	5	8	15	23	46	7	4
RoomsList	8000	8000	0	0%	30.769	3	12	18	28	37	61	14	7
RoomCreateForm	8000	8000	0	0%	30.769	3	9	13	23	30	55	11	6
ErrorCre...BackForm	4000	4000	0	0%	15.385	4	12	19	29	37	54	14	8
RoomCreated	4000	4000	0	0%	15.385	5	12	19	29	37	67	14	8

Todos los assert se han cumplido correctamente y el número de KO es del 0% por parte del sistema.

Historia de Usuario 11	Yo COMO owner QUIERO que mis mascotas estén en una sala dedicada a dicho tipo de animal, es decir perro en una sala para perro, una sala para gatos, etc PARA evitar que los animales tengan problemas con otro tipo de animal.
Escenario 1	Un owner crea una reserva en una sala de Dogs para un Pet de tipo Dog.
Escenario 2	Un owner crea una reserva en una sala de Bird para un Pet Dog.
Script	SamePetTypeReservationTwoScenariosHU11Diagnosis.

Mínimo número de usuarios concurrentes que no es soportado por el sistema

Con un número de 3000 usuarios en 150 segundos los resultados son:



La CPU durante todo el proceso está rondando el 100% de su capacidad y la memoria está muy saturada, además de un tiempo de espera de más de 12 segundos en algunos casos y bastantes KOs.

Estadísticas con 3000 Usuarios concurrentes por escenario:

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	56576	43893	12683	22%	97.043	0	2690	51732	60013	60018	60038	20085	25149			
request_0	9000	7598	1402	16%	15.437	2	18	16526	60012	60016	60025	13535	22286			
LoginFormOwner	6000	6000	0	0%	10.292	0	5	899	50275	55069	55066	6834	15731			
LoggedOwner	6000	5171	829	14%	10.292	2	230	27253	60011	60017	60021	15574	22995			
LoggedOw...direct 1	5171	4771	400	8%	8.87	2	14	16468	60006	60015	60025	12630	21605			
RoomsList	6000	4012	1988	33%	10.292	7	1794	60004	60014	60018	60027	23862	27977			
RoomDetails2	3000	1847	1153	38%	5.146	13	10079	60006	60015	60019	60035	27383	28206			
Room4Details	3000	1853	1147	38%	5.146	13	9141	60005	60014	60018	60036	27248	28219			
BadReservationForm	3000	2234	766	26%	5.146	2	19757	60000	60014	60019	60027	26420	24979			
GoodReservationForm	3000	2255	745	25%	5.146	2	19363	59844	60014	60019	60038	26232	24957			
GoodRese...nCreated	3000	1707	1293	43%	5.146	1	22626	60002	60014	60019	60033	29640	25017			
GoodRese...direct 1	1703	1327	376	22%	2.921	0	23397	58431	60013	60017	60020	28035	24079			
ReservationFailed	3000	1676	1324	44%	5.146	2	27106	60002	60014	60018	60024	31963	23618			
ReturnRoomDetails	3000	1740	1260	42%	5.146	1	51821	60007	60015	60019	60032	37468	24279			
RoomDeta...direct 1	28	28	0	0%	0.048	3846	4002	4143	4161	4163	4163	4025	120			
Room4Det...direct 1	28	28	0	0%	0.048	3847	3997	4143	4162	4166	4167	4018	118			
BadReser...direct 1	405	405	0	0%	0.695	1	603	1560	3338	3835	4158	993	1181			
GoodRese...direct 1	403	403	0	0%	0.691	0	301	1547	3328	3822	4158	961	1169			
Reservat...direct 1	419	419	0	0%	0.719	0	8466	8861	11422	11944	11946	6937	3517			
ReturnRo...direct 1	419	419	0	0%	0.719	0	1080	2650	8284	10116	10696	2030	2714			

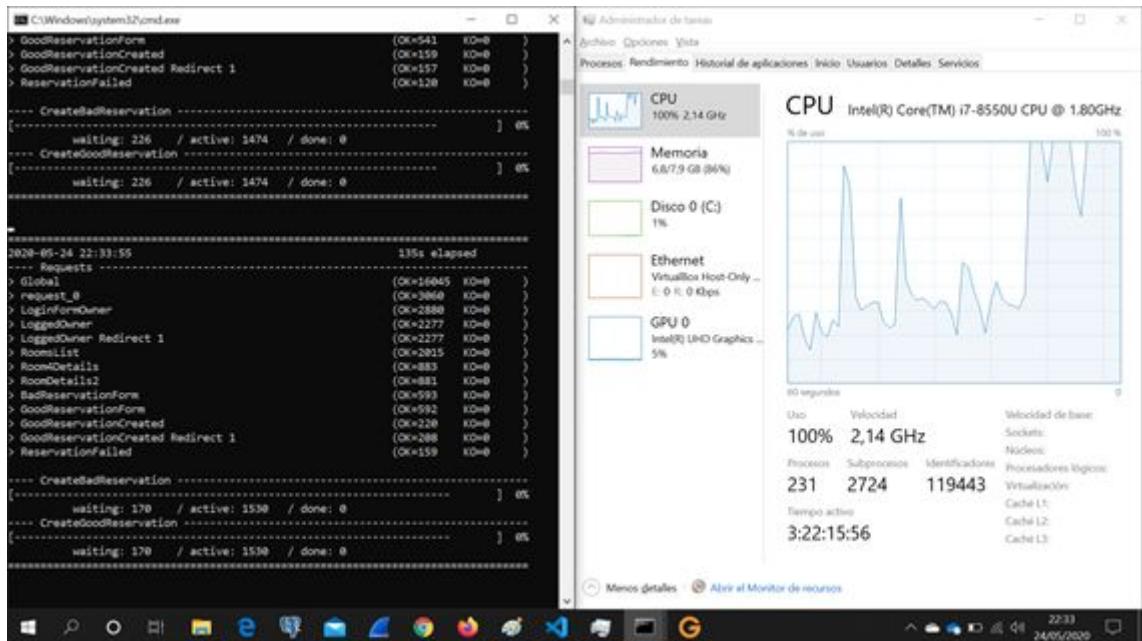
Se observa el porcentaje de KO del 22% y un tiempo máximo en respuestas de hasta 60 segundos en algunos casos y la media de respuestas es muy alta también casi siempre por encima de los 10 seg excepto algunos casos..

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

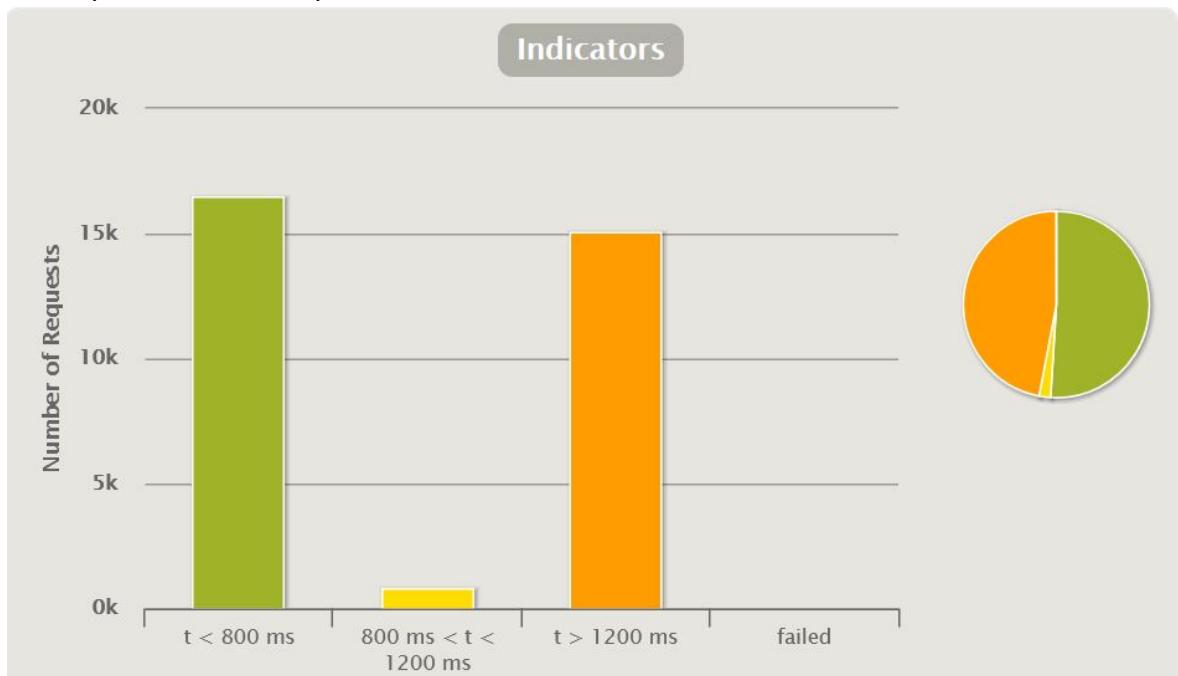
En cuanto a los Assert vemos que no se cumplen ninguno de los 3 establecidos.

Con 1000 Usuarios concurrentes por cada escenario durante 150 segundos.

CPU + GATLING



Cuando alcanza los 1700 usuarios concurrentes por escenario la CPU comienza a rondar el 100 por cien de su capacidad.



Vemos como todas las request se han completado con éxito en menos de 800 milisegundos, pero muchas de ellas han precisado más de 12 segundos.

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

Solo se ha cumplido 1 assert de los tres propuestos el que hace referencia a un 95% de eventos exitosos, pero en cuanto al tiempo tenemos un mal rendimiento.

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	32299	32298	1	0%	70.987	0	588	14114	34566	38752	60000	8018	11473		
request_0	5100	5100	0	0%	11.209	1	10	4010	28059	29933	30974	4257	8354		
LoginFormOwner	3400	3400	0	0%	7.473	0	3	5	3955	6674	8217	403	1318		
LoggedOwner	3400	3400	0	0%	7.473	1	9	3792	15103	20191	34962	2956	5306		
LoggedOw...direct 1	3400	3400	0	0%	7.473	1	8	3507	16560	20658	27777	2904	5453		
RoomsList	3400	3400	0	0%	7.473	6	269	9640	29865	35582	51263	6233	10023		
RoomDetails2	1700	1700	0	0%	3.736	13	1143	18052	37197	42952	51447	9627	13376		
Room4Details	1700	1700	0	0%	3.736	12	881	16345	35976	41790	51813	9105	12951		
BadReservationForm	1700	1700	0	0%	3.736	16	8036	22713	36431	42944	58609	12707	13405		
GoodReservationForm	1700	1700	0	0%	3.736	16	7947	23787	36356	43690	51726	12830	13471		
GoodRese...nCreated	1700	1700	0	0%	3.736	25	15384	24772	36086	41242	50986	16292	11933		
GoodRese...direct 1	1699	1698	1	0%	3.734	41	18172	27707	38110	43988	60000	18113	12590		
ReservationFailed	1700	1700	0	0%	3.736	23	16502	25770	36165	42235	59486	16926	11939		
ReturnRoomDetails	1700	1700	0	0%	3.736	20	18506	28884	36852	42679	50167	18993	11533		

En las estadísticas vemos un porcentaje de KO del 1% y que los tiempos máximos y las medias son muy altas, en algunos casos.

SI BAJAMOS EL NÚMERO DE USUARIOS A 1000 EL SISTEMA CUMPLE LOS TRES ASSERTS.

ASSERTIONS											Expand all groups Collapse all groups			
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		

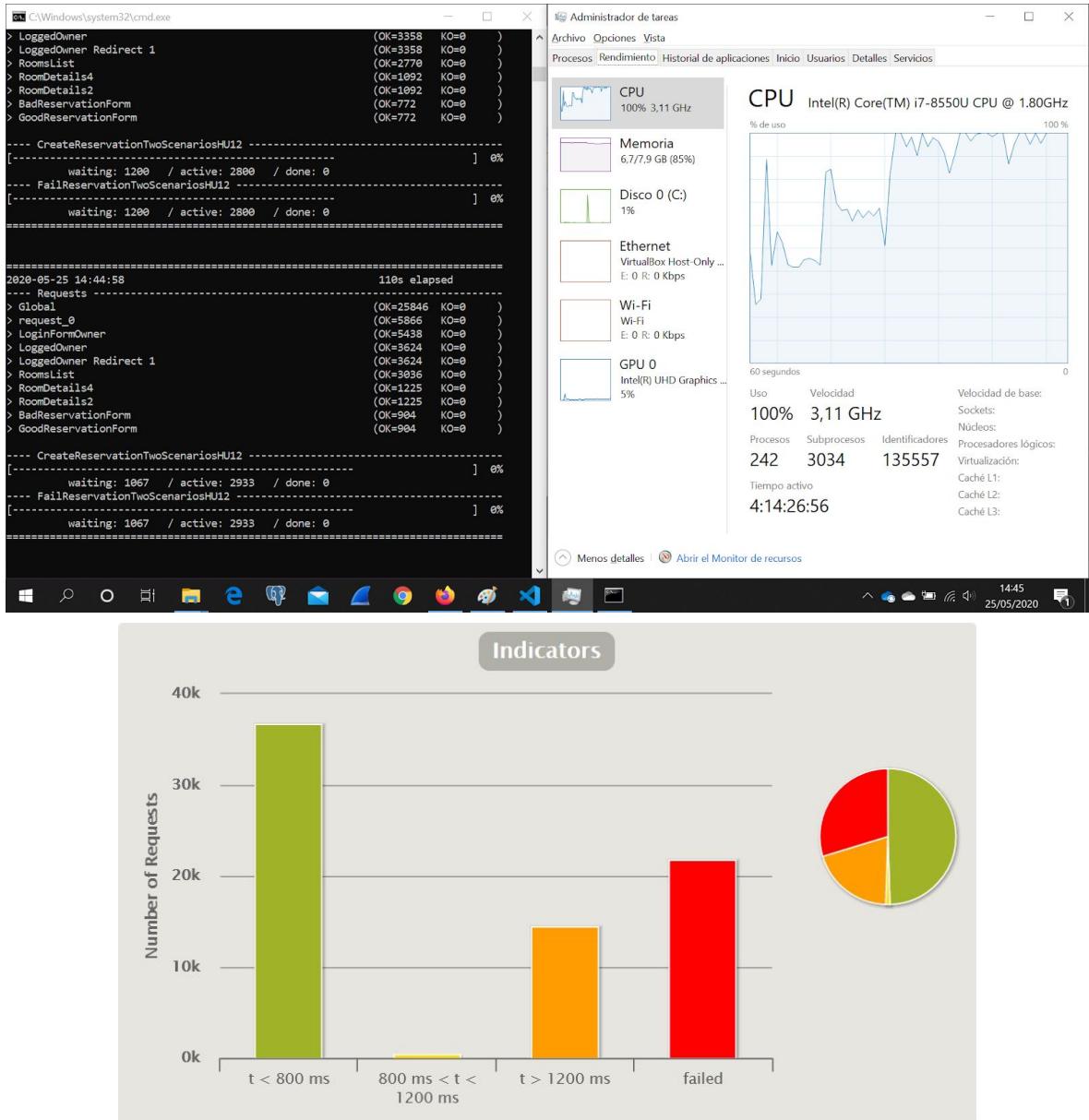
STATISTICS											Expand all groups Collapse all groups				
Requests ^	Executions					Response Time (ms)									
	Total ↴	OK ↴	KO ↴	% KO ↴	Cnt/s ↴	Min ↴	50th pct ↴	75th pct ↴	95th pct ↴	99th pct ↴	Max ↴	Mean ↴	Std Dev ↴		
Global Information	19000	19000	0	0%	64.407	0	14	36	689	1276	3713	98	264		
request_0	3000	3000	0	0%	10.169	1	6	9	16	1595	3713	39	285		
LoginFormOwner	2000	2000	0	0%	6.78	0	3	3	7	82	120	5	12		
LoggedOwner	2000	2000	0	0%	6.78	1	5	6	65	866	1686	28	134		
LoggedOw...direct 1	2000	2000	0	0%	6.78	2	5	6	13	54	182	7	16		
RoomsList	2000	2000	0	0%	6.78	6	18	22	598	1036	1627	72	218		
RoomDetails2	1000	1000	0	0%	3.39	15	177	421	1010	1652	2213	310	372		
Room4Details	1000	1000	0	0%	3.39	12	35	43	894	1103	1953	115	255		
BadReservationForm	1000	1000	0	0%	3.39	15	30	58	1028	1357	1827	148	320		
GoodReservationForm	1000	1000	0	0%	3.39	17	30	60	1106	1657	2108	153	335		
GoodRese...nCreated	1000	1000	0	0%	3.39	20	32	47	392	1019	1962	92	197		
GoodRese...direct 1	1000	1000	0	0%	3.39	27	555	740	1061	1696	2477	539	346		
ReservationFailed	1000	1000	0	0%	3.39	21	33	47	471	962	1172	90	178		
ReturnRoomDetails	1000	1000	0	0%	3.39	10	18	33	377	967	1916	74	190		

y se ve como los tiempos de respuesta se reducen considerablemente.

Historia de Usuario 12	Yo COMO owner QUIERO poder realizar reservas de salas PARA dejar a mi mascota durante mi tiempo de vacaciones.
Escenario 1	Un owner crea un reserva correcta sin fallos.
Escenario 2	Un owner crea un reserva con fechas en pasado.
Script	CreateReservationTwoScenariosHU12Diagnosis

Mínimo número de usuarios concurrentes que no es soportado por el sistema

Con un número de 4000 usuarios en 150 segundos los resultados son:



Una CPU que durante todo el proceso está rondando el 100% de su capacidad y una memoria muy saturada, además muchas request precisan más de 12 seg y se producen bastantes KO.

Estadísticas con 4000 Usuarios concurrentes por escenario:

▶ 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	73342	51568	21774	30%	123.264	0	567	60000	60015	60019	60077	20444	26267
request_0	12000	9936	2064	17%	20.168	1	5	11716	60013	60018	60038	12120	22130
LoginFormOwner	8000	7540	460	6%	13.445	0	2	12	60002	60016	60025	7716	17662
LoggedOwner	7540	6428	1112	15%	12.672	1	8	32392	60013	60019	60036	16213	23139
LoggedOw...direct 1	6428	5462	966	15%	10.803	1	5	22933	60013	60018	60031	14043	23333
RoomsList	8000	4699	3301	41%	13.445	4	3103	60006	60017	60020	60036	26020	28653
RoomDetails4	4000	2166	1834	46%	6.723	7	2371	60005	60017	60019	60057	24238	27714
RoomDetails2	4000	2168	1832	46%	6.723	7	2376	60005	60017	60020	60077	24390	27730
BadReservationForm	4000	2199	1801	45%	6.723	9	27084	60006	60017	60020	60025	30049	27705
GoodReservationForm	4000	2189	1811	45%	6.723	9	29763	60006	60016	60020	60024	30408	27736
BadReservation	3849	1967	1882	49%	6.469	1	31911	60004	60016	60019	60038	31170	25508
GoodRese...nCreated	3849	1861	1988	52%	6.469	0	37532	60005	60016	60019	60038	33630	24834
GoodRese...direct 1	1861	1235	626	34%	3.128	0	23782	60005	60017	60019	60027	28531	26853
ReturnRoomDetails	4000	1903	2097	52%	6.723	0	60000	60009	60018	60020	60038	32937	29027
BadReser...direct 1	252	252	0	0%	0.424	1	1570	1754	1912	1935	1942	1514	330
GoodRese...direct 1	249	249	0	0%	0.418	1	1606	1765	1900	1935	1942	1518	330
BadReser...direct 1	529	529	0	0%	0.889	0	3	570	1513	1871	1942	332	564
ReturnRo...direct 1	785	785	0	0%	1.319	0	1	2	315	840	935	38	150

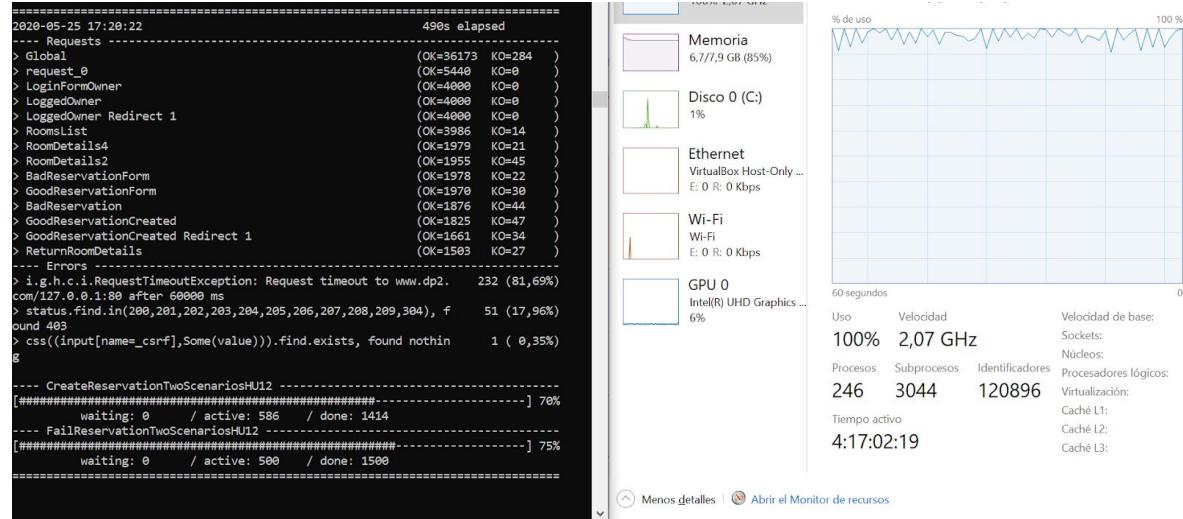
Se observa el porcentaje de KO del 30% y un tiempo máximo en respuestas de hasta 60 segundos en algunos casos y la media está por encima de los 10 segundos en casi todas las requests.

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

En cuanto a los Assert vemos que no se cumplen ninguno de los 3 establecidos.

Con 2000 Usuarios concurrentes por cada escenario durante 150 segundos.

CPU + GATLING



Cuando alcanza los 2000 usuarios concurrentes por escenario la CPU comienza a rondar el 100 por cien de su capacidad y se producen una serie KOs.



Vemos como todas las request se han completado con éxito en menos de 800 milisegundos, pero que muchas más han necesitado más de 12 segundos.

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

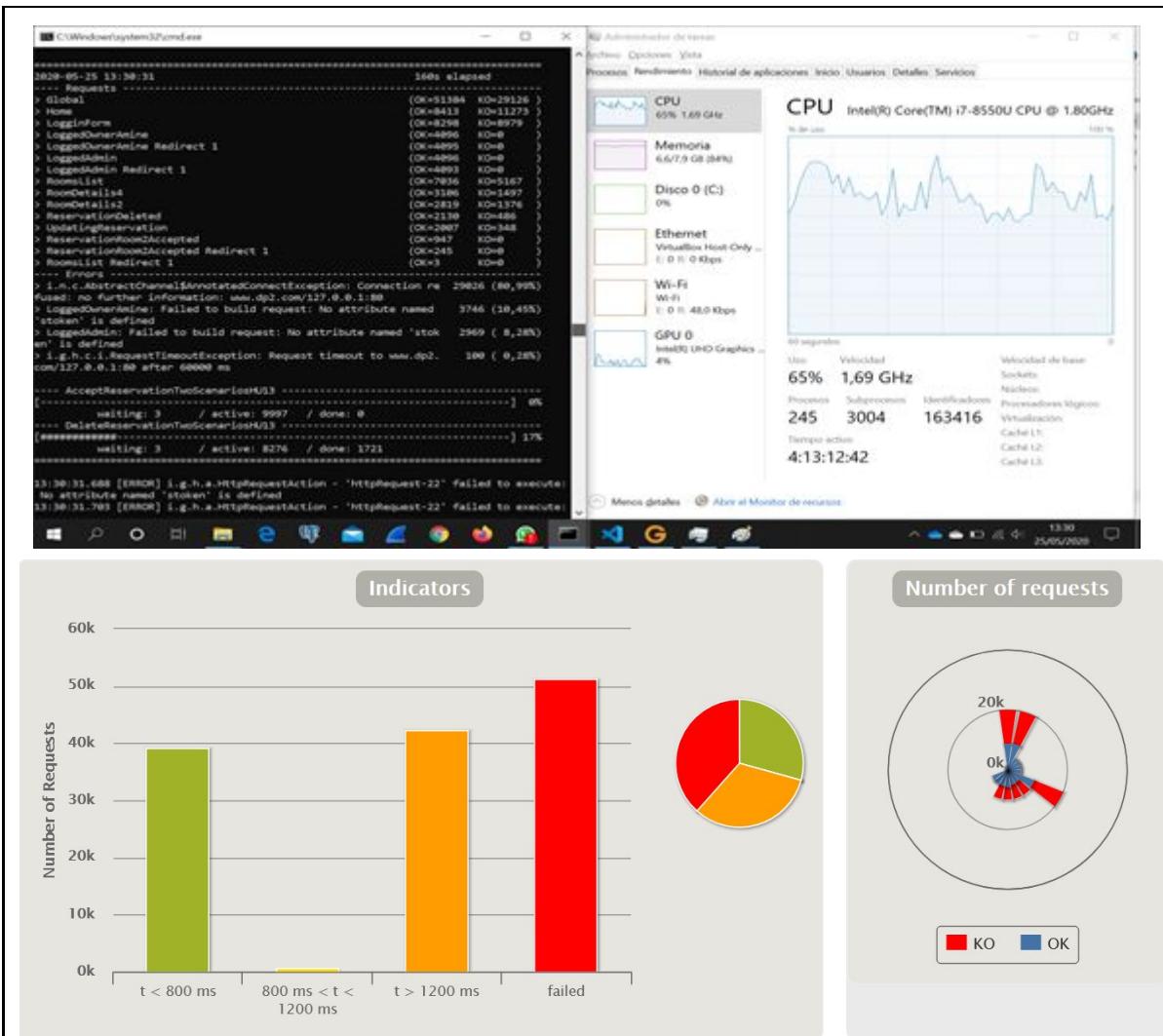
Solo se han cumplido uno de los 3 asserts y es el que indica que se han realizado correctamente más del 95% de las request.

▶ 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	37951	37666	285	1%	63.252	0	1779	26430	51435	58966	60033	13832	18216		
request_0	6000	6000	0	0%	10	1	12	12229	43814	49765	52458	8356	14788		
LoginFormOwner	4000	4000	0	0%	6.667	0	3	16	5741	9908	11693	686	1956		
LoggedOwner	4000	4000	0	0%	6.667	1	74	8372	27942	35921	47791	6037	9776		
LoggedOw...direct 1	4000	4000	0	0%	6.667	1	17	10203	35828	39726	40551	7205	12068		
RoomsList	4000	3986	14	0%	6.667	6	1520	25730	47673	54806	60014	12827	17881		
RoomDetails4	2000	1979	21	1%	3.333	10	3189	41090	56946	60000	60014	17496	21481		
RoomDetails2	2000	1955	45	2%	3.333	10	3671	42025	58361	60009	60021	18209	22118		
BadReservationForm	2000	1978	22	1%	3.333	16	8748	40647	57122	60004	60025	20056	21262		
GoodReservationForm	2000	1970	30	2%	3.333	15	9504	41197	56958	60004	60015	20246	21324		
BadReservation	2000	1956	44	2%	3.333	22	22612	38222	56447	60000	60014	23488	18185		
GoodRese...nCreated	2000	1952	48	2%	3.333	38	23920	39592	56379	59329	60018	25017	17336		
GoodRese...direct 1	1951	1917	34	2%	3.252	24	27979	41429	58082	60003	60015	27889	17271		
ReturnRoomDetails	2000	1973	27	1%	3.333	13	29956	44416	56972	60002	60033	32175	14973		

En las estadísticas vemos que hay un 1% de KO pero los tiempos de respuesta medios son muy altos.

Por lo tanto podemos afirmar que nuestro sistema dará una mejor respuesta si el número de usuarios concurrentes es menor que los 2000 por escenario.

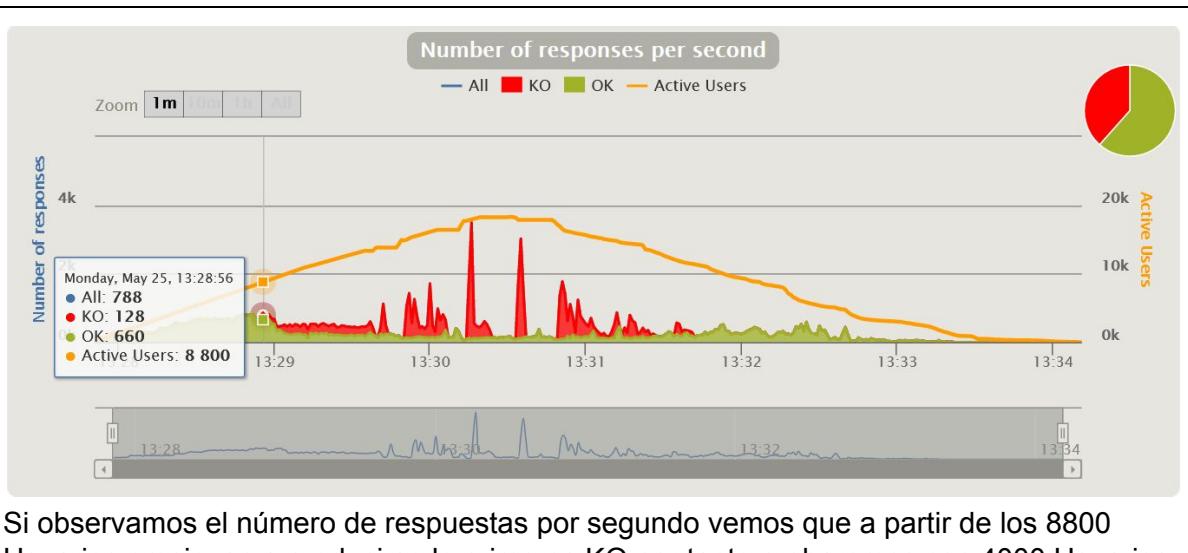
Historia 13	Yo COMO admin QUIERO poder aceptar o rechazar las reservas de salas PARA poder llevar una gestión precisa de qué reservas pueden ser aceptadas y cuáles rechazadas.
Escenario 1	Un admin aceptar una reserva que está pendiente
Escenario 2	Un admin eliminar una reserva que no está aceptada
Script	AcceptReservationTwoScenariosHU13Diagnosis
Mínimo número de usuarios concurrentes que no es soportado por el sistema	
Con un número de 10000 usuarios en 150 segundos los resultados son:	



Vemos como muchas de las request se han completado con éxito en menos de 800 milisegundos, pero que una cantidad mayor ha necesitado más de 12 seg y los fallos corresponden al mayor número de request.

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

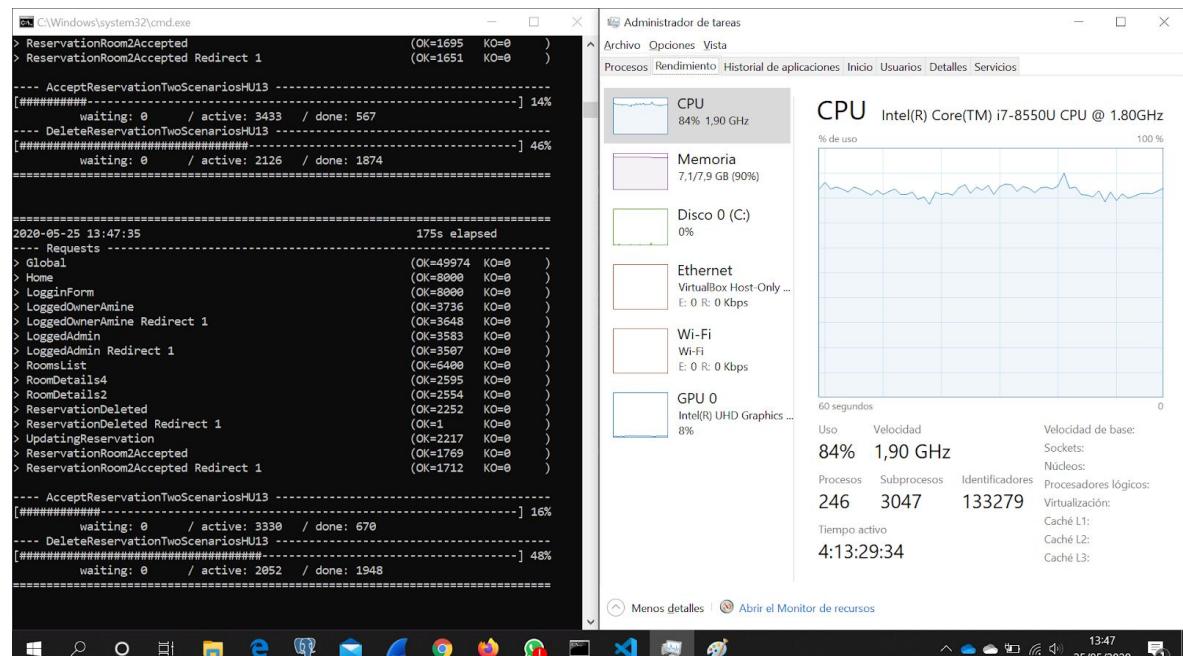
Observamos que no se cumplen ninguno de los tres Asserts, el sistema está totalmente sobrepasado.



Si observamos el número de respuestas por segundo vemos que a partir de los 8800 Usuarios empiezan a producirse los primeros KO por tanto probaremos con 4000 Usuarios en el siguiente análisis.

Con 4000 Usuarios concurrentes por cada escenario durante 100 segundos.

CPU + GATLING



Cuando alcanza los 4000 usuarios concurrentes por escenario la CPU comienza a rondar el 85 por cien de su capacidad e incluso llega a picos del 100%.



Vemos como bastantes request se han completado con éxito en menos de 800 milisegundos, pero que la gran mayoría han necesitado más de 12 segundos.

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

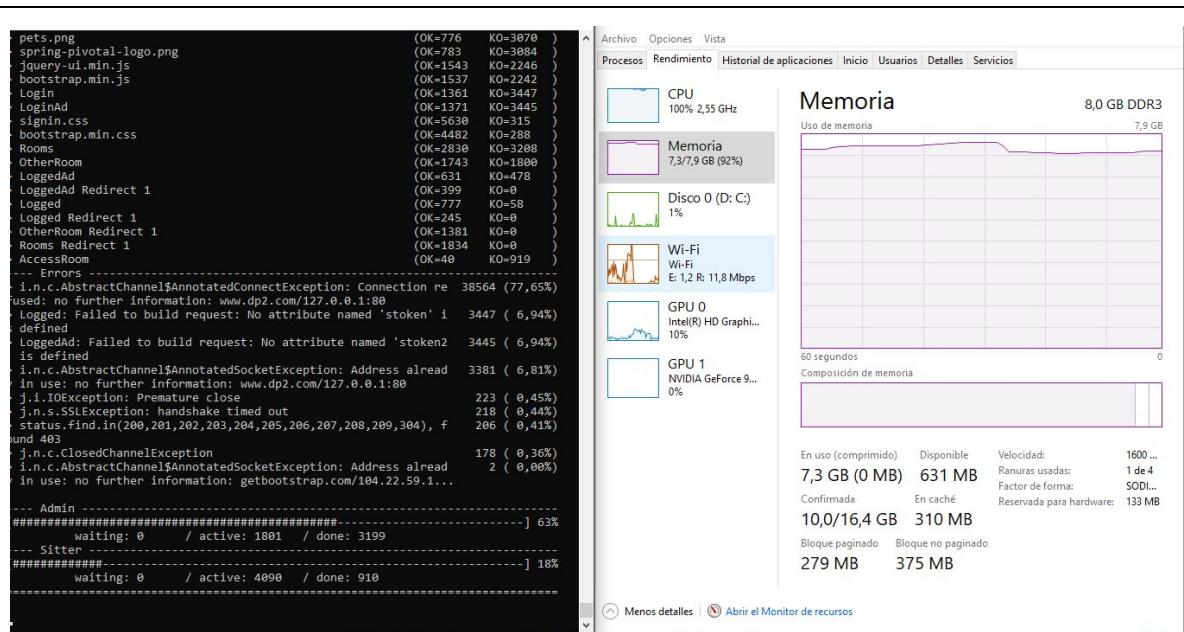
Se ha cumplido un assert correctamente, el que indica que el 95% de los eventos se han llevado a cabo con éxito, por otro lado, el número de KO es del 0% por parte del sistema.

▶ 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		64001	64001	0	0%	196.926	0	3122	4129	6572	9496	23704	2670	2344
Home		8000	8000	0	0%	24.615	1	15	3390	4339	4518	4661	1395	1755
LoginForm		8000	8000	0	0%	24.615	0	23	3360	4295	4473	4646	1471	1736
LoggedOwnerAmine		4000	4000	0	0%	12.308	1	2906	3881	6113	9113	16495	2347	2302
LoggedOw...direct 1		4000	4000	0	0%	12.308	1	2869	3380	4237	4496	4661	1863	1700
LoggedAdmin		4000	4000	0	0%	12.308	1	2993	3942	6218	8731	17328	2465	2238
LoggedAd...direct 1		4000	4000	0	0%	12.308	1	2945	3401	4268	4507	4661	1977	1705
RoomsList		8000	8000	0	0%	24.615	4	3140	4284	6794	9311	22400	2852	2358
RoomDetails4		4000	4000	0	0%	12.308	8	3747	4696	7815	10709	19405	3599	2387
RoomDetails2		4000	4000	0	0%	12.308	7	3791	4829	8037	11037	23704	3719	2501
ReservationDeleted		4000	4000	0	0%	12.308	17	3996	5035	8160	11216	22220	4037	2277
Reservat...direct 1		1	1	0	0%	0.003	1255	1255	1255	1255	1255	1255	0	
UpdatingReservation		4000	4000	0	0%	12.308	14	3966	5026	8103	11136	15834	4015	2252
Reservat...Accepted		4000	4000	0	0%	12.308	15	3738	4782	7693	10463	20407	3678	2377
Reservat...direct 1		4000	4000	0	0%	12.308	6	3664	4774	7502	10623	19281	3577	2390

Los tiempos máximos de respuesta están en 23 seg de tope lo cual sigue siendo en rendimiento nada aceptable, pero mejor que con los 10000 Usuarios anteriores. Por lo

tanto podemos afirmar que si el número de Usuarios concurrentes por escenario es menor que 4000, nuestro sistema dará una respuesta aceptable (Cuanto menor sea este número, mejor funcionará)

Historia 14	Yo COMO cuidador QUIERO tener acceso a aquellas salas que tengan animales de los que estoy al cargo PARA saber dónde están los animales y donde debe dejarlos al final de la jornada laboral.
Escenario 1	Un cuidador accede a una sala
Escenario 2	Un admin accede a la misma sala
Mínimo número de usuarios concurrentes que no es soportado por el sistema	
Con un número de 5000 usuarios en 10 segundos los resultados son:	



> Global Information



con un tiempo medio de espera de 11 segundos. Como vemos la CPU siempre esta al 100% y la memoria está casi al máximo.

Máximo número de usuarios que soporta teniendo un buen rendimiento

Con un número de 600 usuarios en 120 segundos:



Historia de Usuario 15	Yo COMO admin QUIERO poder gestionar las solicitudes de los usuarios que quieran ser cuidadores PARA poder tener un control de la plantilla y aumentarla en caso de que fuera necesario.
Escenario 1	Un admin acepta una solicitud
Escenario 2	Un admin deniega una solicitud
Mínimo número de usuarios concurrentes que no es soportado por el sistema	
Con un número de 5000 usuarios en 10 segundos los resultados son:	

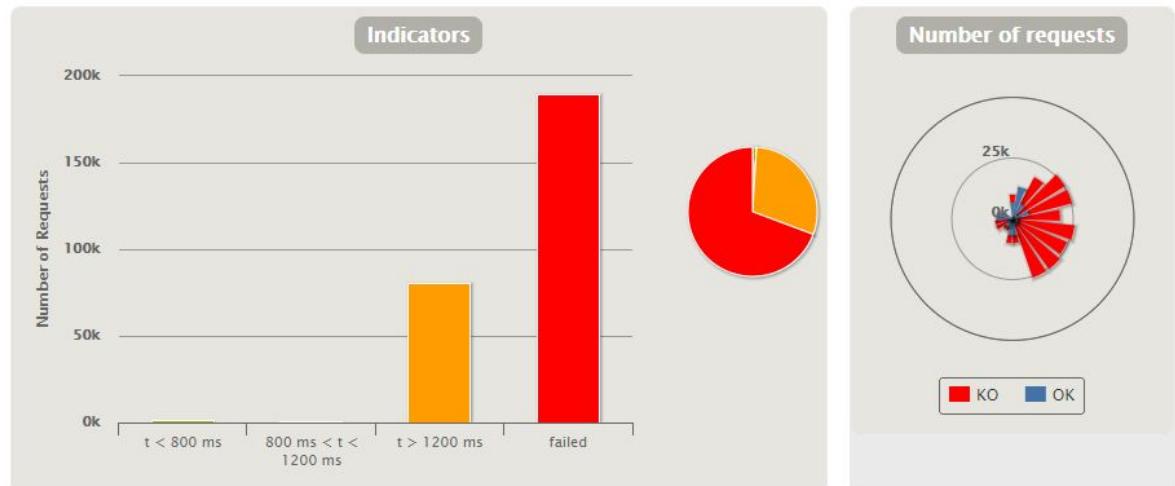
```

G:\WINDOWS\system32\cmd.exe
> bootstrap.min.css          (OK=99   KO=2116 )
> signin.css                 (OK=97    KO=1669 )
> LoggedAd                   (OK=485   KO=0  )
> LoggedAd Redirect 1        (OK=82    KO=0  )
> Reject Redirect 1          (OK=21    KO=0  )
> Accept Redirect 1          (OK=21    KO=0  )
> Request Redirect 1         (OK=11    KO=0  )
> Error: [REDACTED]
> i.n.c.AbstractChannel$AnnotatedConnectException: Connection re... 36736 (52,52%)
e used: further information: www.dp2.com/127.0.0.1:80
> i.n.c.AbstractChannel$AnnotatedSocketException: No buffer spac... 25477 (36,42%)
e available (maximum connections reached): connect: www.dp2.c...
e LoggedAd: Failed to build request: No attribute named 'stoken'...
is defined
> i.n.c.AbstractChannel$AnnotatedSocketException: No buffer spac... 3380 ( 4,83%)
e available (maximum connections reached): connect: maxcdn.bo...
> j.n.s.SSLEngineException: handshake timed out                1396 ( 2,00%)
> i.n.c.AbstractChannel$AnnotatedSocketException: No buffer spac... 499 ( 0,71%)
> i.n.c.ConnectTimeoutException: connection timed out: www.dp2.c... 471 ( 0,67%)
om/127.0.0.1:80 after 60000 ms
> i.n.c.AbstractChannel$AnnotatedSocketException: No buffer spac... 471 ( 0,67%)
e available (maximum connections reached): connect: getbootstrap...
> i.g.h.c.i.RequestTimeoutException: Request timeout to www.dp2... 99 ( 0,14%)
com/127.0.0.1:80 after 60000 ms
> i.n.c.AbstractChannel$AnnotatedSocketException: Address already... 2 ( 0,00%)
y in use: no further information: www.dp2.com/127.0.0.1:80

--- Reject ---
[#####] waiting: 0 / active: 4039 / done: 961
--- Accept ---
[#####] waiting: 0 / active: 4023 / done: 977
[#####] waiting: 0 / active: 4039 / done: 961
--- Accept ---
[#####] waiting: 0 / active: 4023 / done: 977

```

> Global Information



Como vemos la CPU siempre esta al 100% y solo el 40% aproximado a accedido correctamente a la página aunque con un tiempo de respuesta medio de 12 segundos.

Máximo número de usuarios que soporta teniendo un buen rendimiento

Con un número de 650 usuarios en 120 segundos:

Administrator de tareas

Procesos Rendimiento Historial de aplicaciones Inicio Usuarios Detalles Servicios

CPU

Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz

% de uso 100% 2,59 GHz

Memoria 6,8/7,9 GB (86%)

Disco 0 (D: C:) 12%

Wi-Fi Wi-Fi E: 56,0 R: 32,0 Kbps

GPU 0 Intel(R) HD Graphi... 14%

GPU 1 NVIDIA GeForce 9... 0%

60 segundos

Uso Velocidad Velocidad de base: 2,40 G...

100% 2,59 GHz 100% 2,59 GHz 100% 2,59 GHz

Sockets: 1 Núcleos: 2 Procesadores lógicos: 4

Procesos Subprocesos Identificadores Virtualización Habit...

168 1774 74026 Caché L1: 128 kB

Tiempo activo 0:03:25:38 Caché L2: 512 kB

Caché L3: 3,0 MB

Global Information

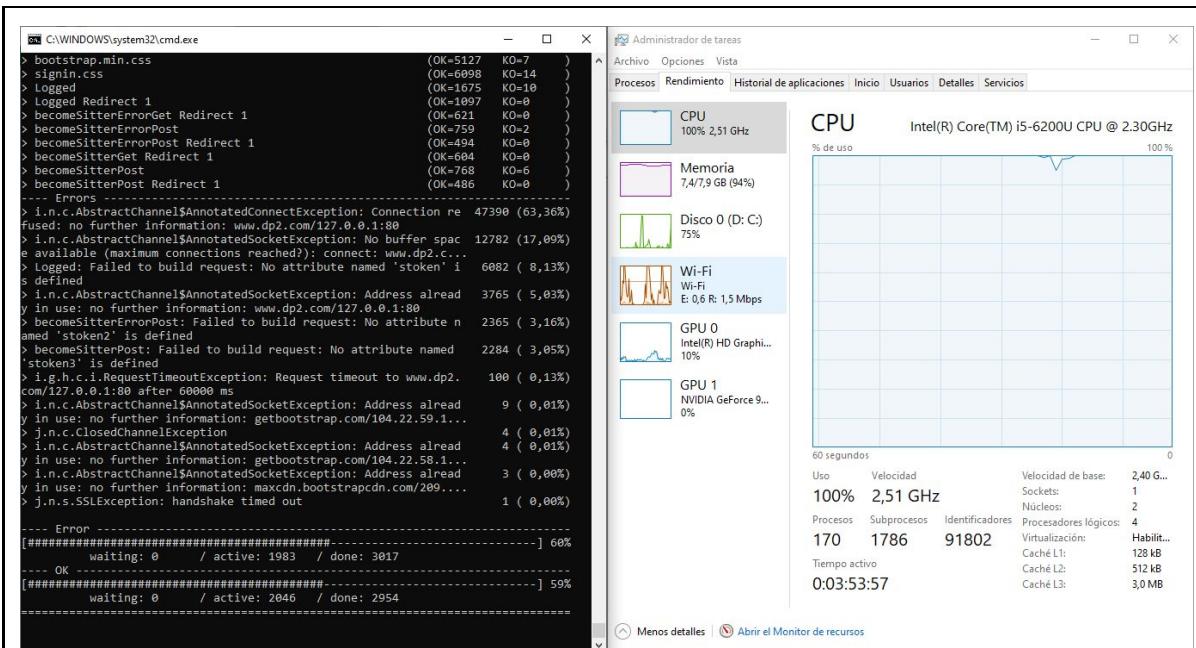
Indicators

Number of Requests

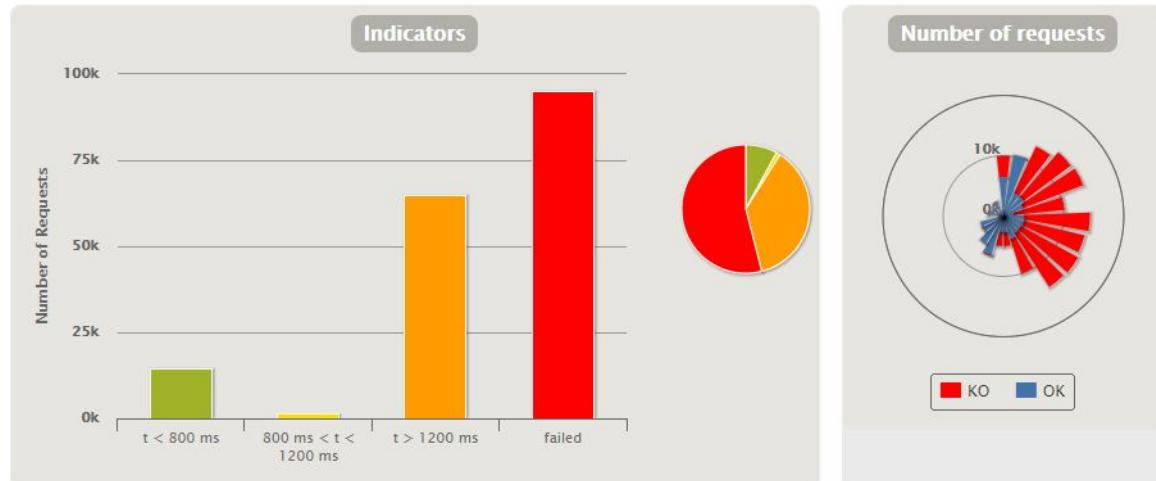
Number of requests

Como vemos a partir de los 650 usuarios no podríamos garantizar un buen rendimiento de la web, el tiempo medio es de 0,4 segundos y con un tiempo máximo puntual de 8 segundos.

Historia 16	Yo COMO usuario registrado QUIERO tener acceso a un solicitud que me permita entregar una solicitud PARA convertirme en un cuidador
Escenario 1	Un usuario rellena una solicitud para convertirse en cuidador
Escenario 2	Un usuario intenta enviar un formulario erróneo para convertirse en cuidador
Mínimo número de usuarios concurrentes que no es soportado por el sistema	
Con un número de 5000 usuarios en 10 segundos los resultados son:	



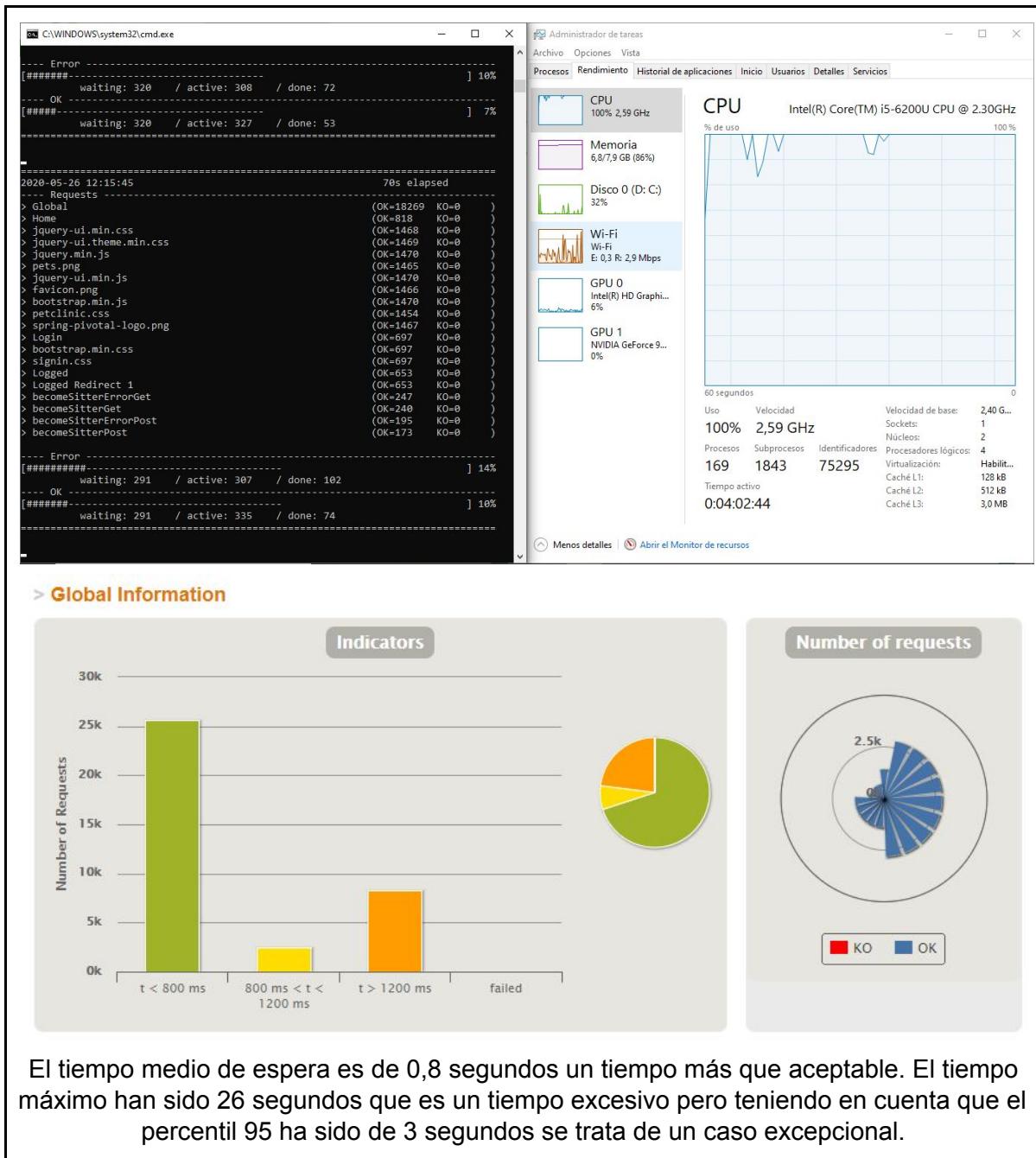
> Global Information



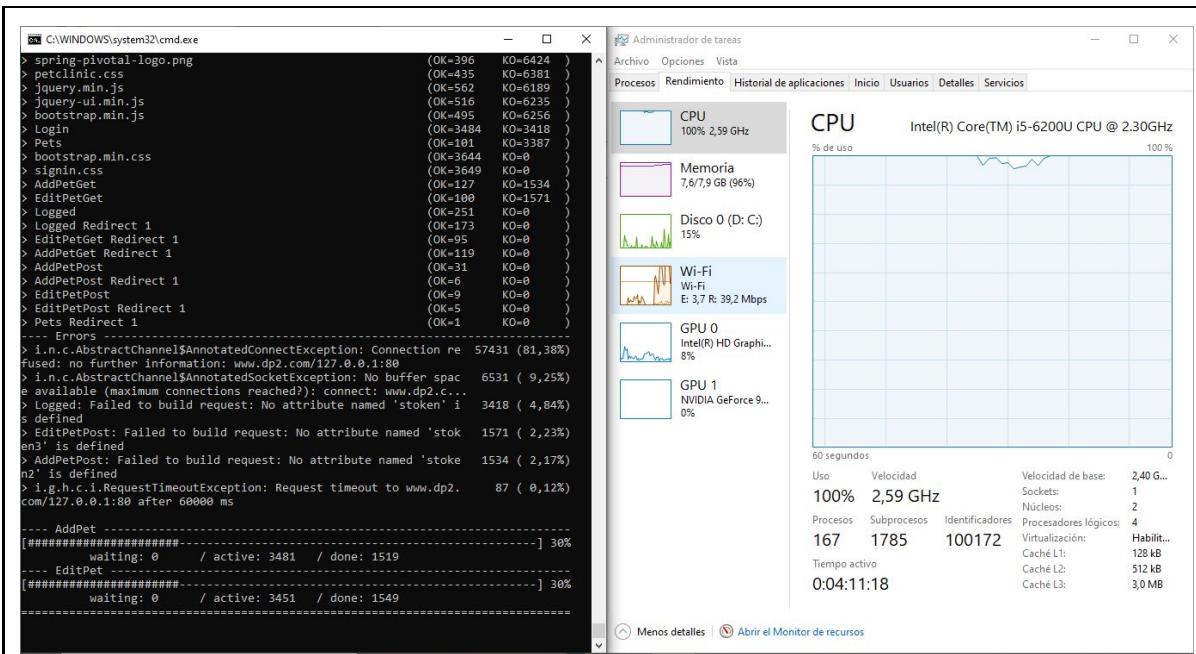
Como vemos más de la mitad de los usuarios no ha logrado acceder al sistema y los que lo han hecho la mayoría a tardado una media de 10 segundos

Máximo número de usuarios que soporta teniendo un buen rendimiento

Con una cantidad de 700 usuarios en 120 segundos los resultados son:



Historia 17	Yo COMO owner QUIERO poder registrar una mascota a mi nombre PARA poder tener un registro de las mismas.
Escenario 1	Un owner registra una mascota
Escenario 2	Un owner edita su mascota
Mínimo número de usuarios concurrentes que no es soportado por el sistema	
Con un número de 5000 usuarios en 10 segundos los resultados son:	



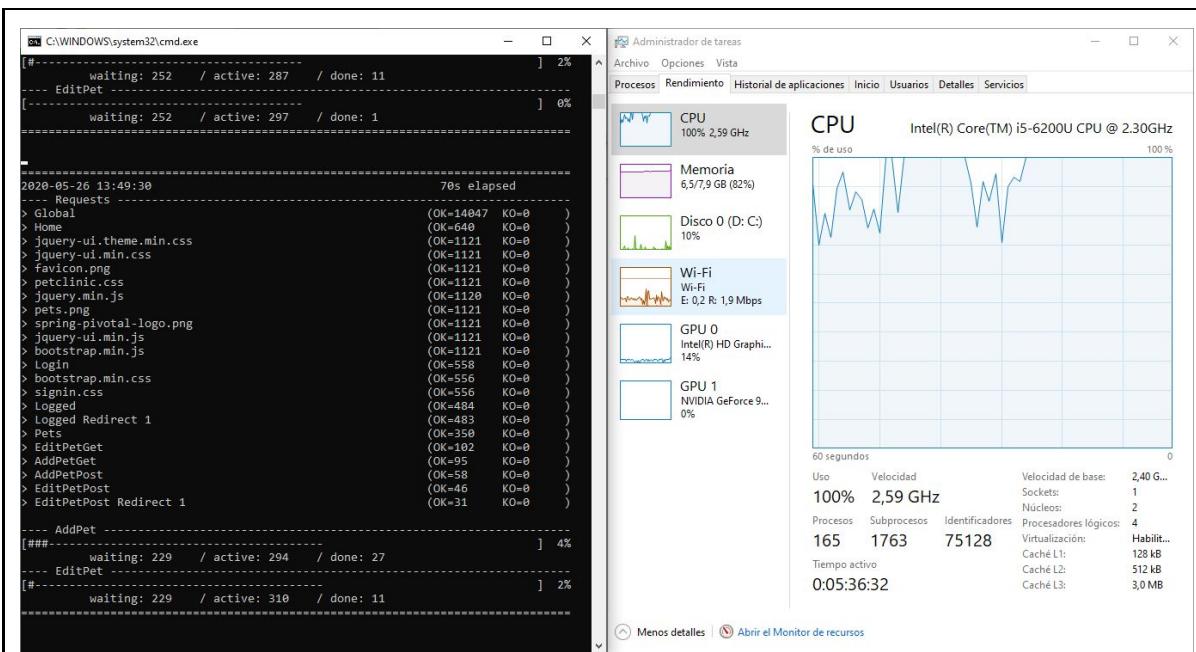
> Global Information



Con un tiempo medio de 12 segundos y un tiempo máximo de 87 segundos

Máximo número de usuarios que soporta teniendo un buen rendimiento

Para un número de 550 usuarios en 120 segundos los resultados son:

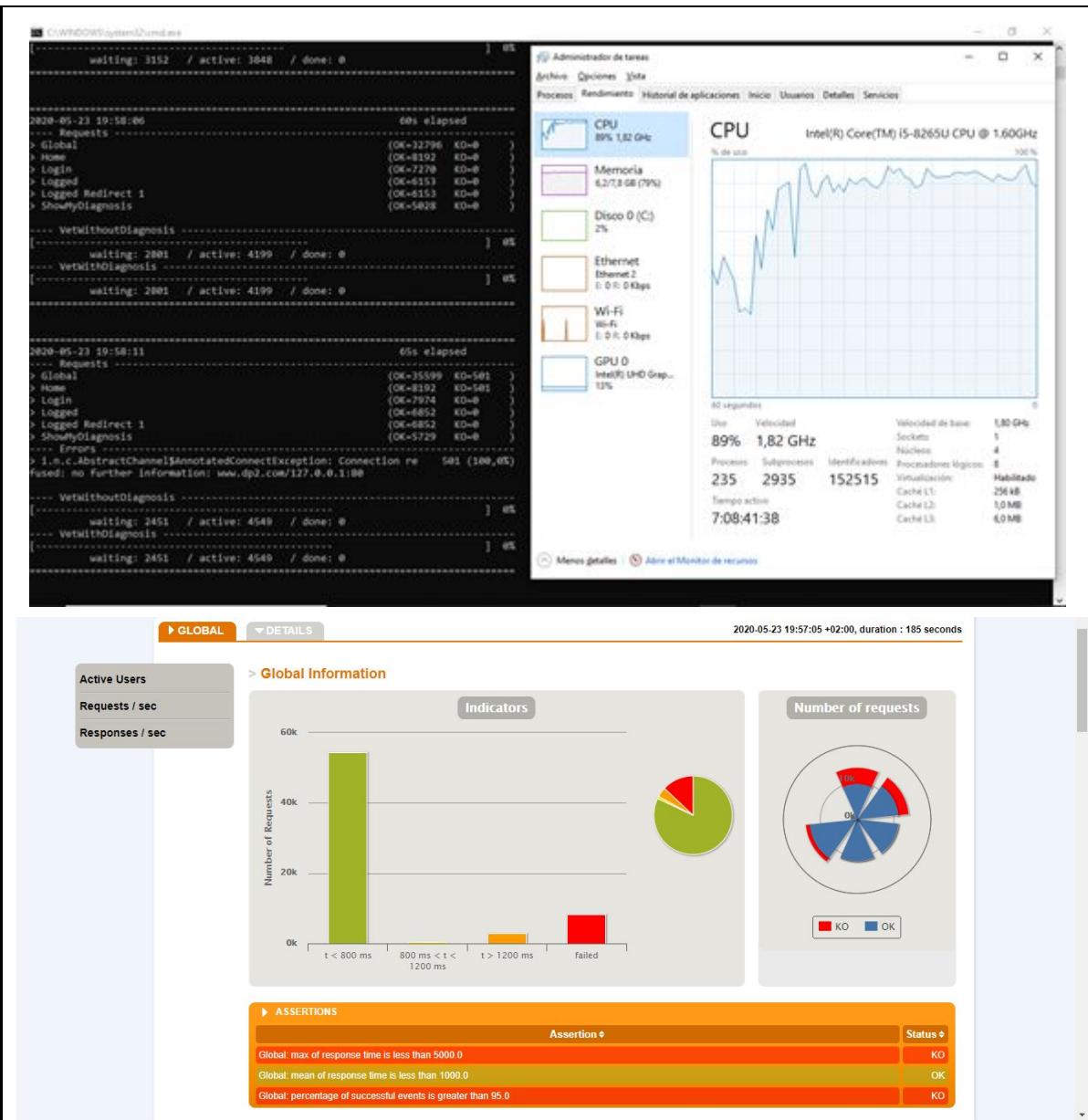


> Global Information



Que aunque la media sea de 1,6 segundos consideramos que es un tiempo aceptable para afirmar que este seria el maximo que aguanta la aplicación con un buen rendimiento.

Historia 18	Yo COMO veterinario QUIERO tener acceso al diagnóstico de las pets que he tratado PARA poder llevar un seguimiento de sus diferentes patologías.
Escenario 1	Un Vet tiene diagnóstico de una pet tratada
Escenario 2	Un Vet tiene no diagnóstico de una pet, ya que no la ha tratado
Mínimo número de usuarios concurrentes que no es soportado por el sistema	
Con un número de 7000 usuarios en 100 segundos:	



Hemos conseguido un tiempo medio de respuesta de 0,4 segundos, pero aun así la CPU casi siempre esta al 90%. Cuando el sistema tiene en activo 4500 usuarios, empieza a cometer fallos en las peticiones y no obtenemos un buen rendimiento

Máximo número de usuarios que soporta teniendo un buen rendimiento

Con un número de 5000 usuarios en 100 segundos:

```

C:\WINDOWS\system32\cmd.exe
[##### waiting: 0 / active: 2054 / done: 2946] 58%
... VetWithDiagnosis ...
[##### waiting: 0 / active: 2055 / done: 2945] 58%
=====

2020-05-26 19:24:14 145s elapsed
--- Requests -----
> Global (OK=50000 KO=0 )
> Home (OK=10000 KO=0 )
> Login (OK=10000 KO=0 )
> Logged (OK=10000 KO=0 )
> Logged Redirect 1 (OK=10000 KO=0 )
> ShowMyDiagnosis (OK=10000 KO=0 )

--- VetWithoutDiagnosis -----
[##### waiting: 0 / active: 1815 / done: 3185] 63%
--- VetWithDiagnosis -----
[##### waiting: 0 / active: 1816 / done: 3184] 63%
=====

2020-05-26 19:24:19 158s elapsed
--- Requests -----
> Global (OK=50000 KO=0 )
> Home (OK=10000 KO=0 )
> Login (OK=10000 KO=0 )
> Logged (OK=10000 KO=0 )
> Logged Redirect 1 (OK=10000 KO=0 )
> ShowMyDiagnosis (OK=10000 KO=0 )

--- VetWithoutDiagnosis -----
[##### waiting: 0 / active: 1554 / done: 3446] 68%
--- VetWithDiagnosis -----
[##### waiting: 0 / active: 1554 / done: 3446] 68%
=====

Administrator de tareas
Archivo Opciones Vista
Procesos Rendimiento Historial de aplicaciones Inicio Usuarios Detalles Servicios
CPU 11% 1,45 GHz
Memoria 6,1/7,8 GB (78%)
Disco 0 (C) 0%
Ethernet Ethernet 2 E: 0 R: 0 Kbps
Wi-Fi Wi-Fi E: 0 R: 0 Kbps
GPU 0 Intel(R) UHD Grap... 9%
8,0 GB
7,8 GB
Uso de memoria
60 segundos Composición de memoria
En uso (comprimido) Disponible Velocidad: 2133 ...
6,0 GB (222 MB) 1,8 GB Ranuras usadas: 2 de 2
Confirmada En caché Factor de forma: Row ...
9,8/15,0 GB 1,6 GB Reservada para hardware: 185 MB
Bloque paginado Bloque no paginado
489 MB 513 MB
Menos detalles | Abrir el Monitor de recursos

```



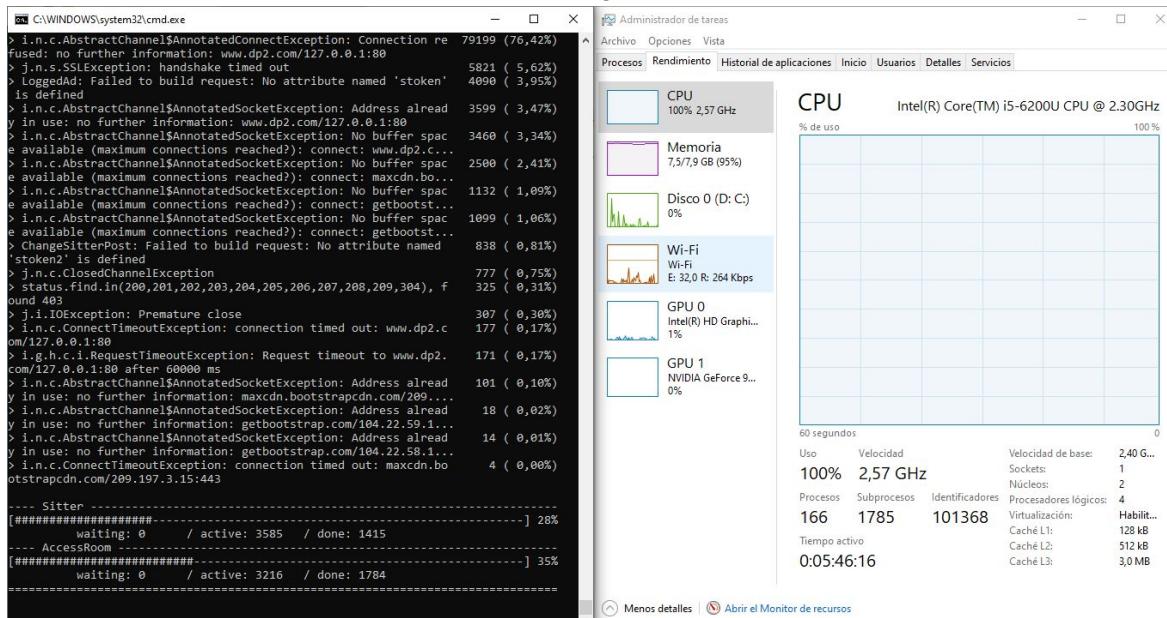
Para este caso, obtenemos un magnífico rendimiento ya que se cumplen todas las peticiones, y los asserts están en OK. Además el tiempo de respuesta media es de 0.06 segundos.

Como conclusión de la Prueba de Rendimiento de la historia 18, podemos ver que de 0-5000 usuarios el sistema responde de forma muy correcta, con tiempos de respuesta muy rápidos y respondiendo a todas las peticiones del sistema. Pero en el momento que se encuentran activas 4500 peticiones a la vez, o más, el sistema empieza a cometer fallos y los tiempos de respuesta aumentan considerablemente.

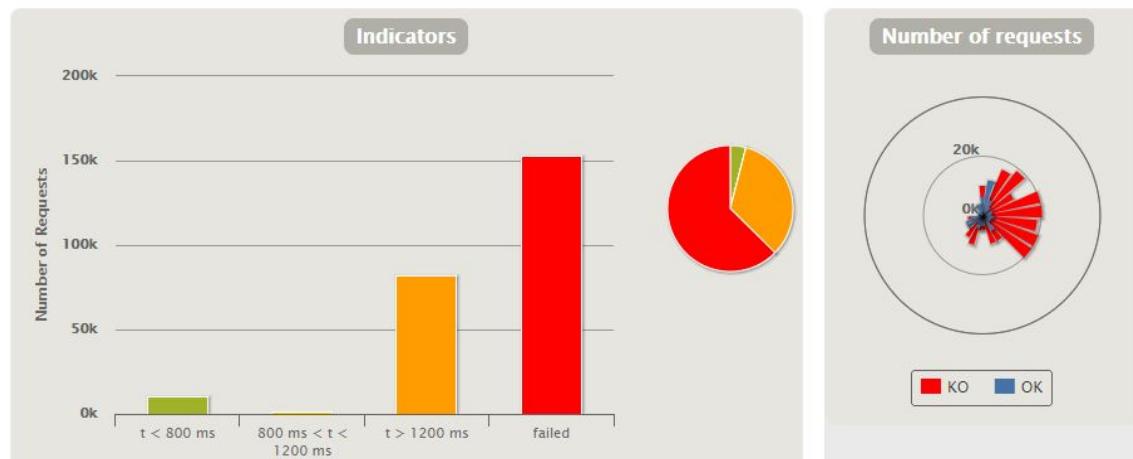
Historia 19	Yo COMO admin QUIERO poder tener un conocimiento de las salas que actualmente están en la clínica PARA poder asignarlas a los cuidadores.
Escenario 1	Un admin accede a una sala
Escenario 2	Un admin accede otra sala y cambia el sitter asignado

Mínimo número de usuarios concurrentes que no es soportado por el sistema

Con 5000 usuarios en 10 segundos los resultados son:



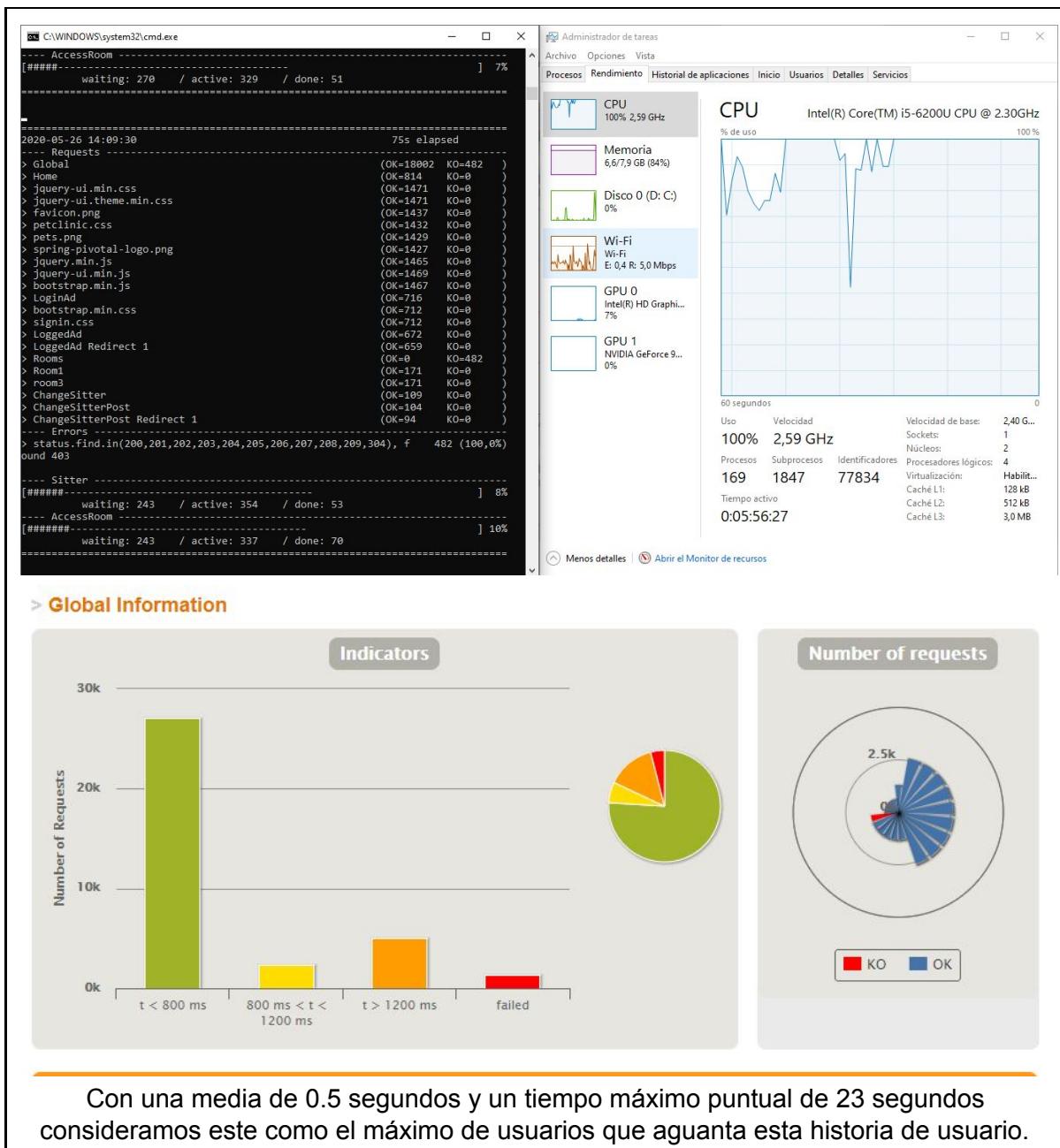
Global Information



Con una media de 12 segundos consideramos que este es el límite del sistema para esta historia de usuario.

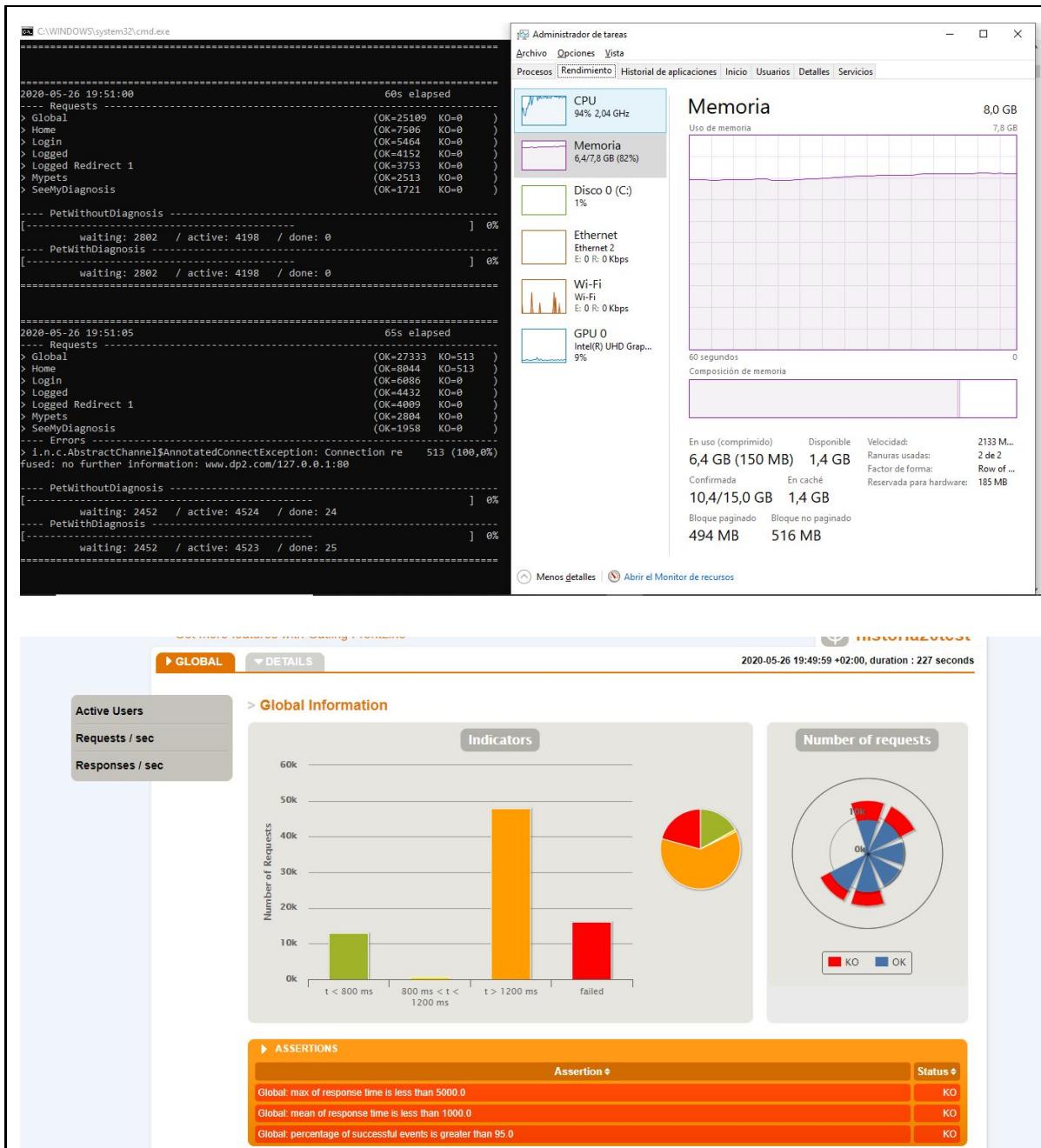
Máximo número de usuarios que soporta teniendo un buen rendimiento

Con un número de 650 usuarios en 120 segundos los resultados son:



Con una media de 0.5 segundos y un tiempo máximo puntual de 23 segundos consideramos este como el máximo de usuarios que aguanta esta historia de usuario.

Historia 20	Yo COMO owner QUIERO poder ver los diagnósticos de mi mascota PARA saber qué seguimiento se le está haciendo.
Escenario 1	Owner ve el diagnóstico de su mascota.
Escenario 2	Owner intenta ver el diagnóstico de su mascota que no ha pasado por el veterinario, por lo tanto aparece vacía.
Mínimo número de usuarios concurrentes que no es soportado por el sistema	
Con un número de 7000 usuarios en 100 segundos:	



El tiempo de respuesta es muy alto, además la CPU y Memoria casi siempre están al 90%. Cuando el sistema tiene en activo 4500 usuarios, empieza a cometer fallos en las peticiones y no obtenemos un buen rendimiento

Máximo número de usuarios que soporta teniendo un buen rendimiento

Con un número de 4500 usuarios en 150 segundos:

```

C:\WINDOWS\system32\cmd.exe
... PetWithDiagnosis
[## waiting: 2551 / active: 1831 / done: 118 ] 2%
-----
2020-05-26 19:43:35 78s elapsed
-- Requests --
> Global (OK=19718 KO=0 )
> Home (OK=4198 KO=0 )
> Login (OK=3778 KO=0 )
> Logged (OK=3296 KO=0 )
> Logged Redirect 1 (OK=3296 KO=0 )
> Mypets (OK=2816 KO=0 )
> SeeMyDiagnosis (OK=2334 KO=0 )

-- PetWithoutDiagnosis --
[### waiting: 2401 / active: 1831 / done: 268 ] 5%
-- PetWithDiagnosis --
[### waiting: 2401 / active: 1833 / done: 266 ] 5%
-----
2020-05-26 19:43:40 75s elapsed
-- Requests --
> Global (OK=21513 KO=0 )
> Home (OK=4498 KO=0 )
> Login (OK=4878 KO=0 )
> Logged (OK=3596 KO=0 )
> Logged Redirect 1 (OK=3595 KO=0 )
> Mypets (OK=3114 KO=0 )
> SeeMyDiagnosis (OK=2632 KO=0 )

-- PetWithoutDiagnosis --
[#### waiting: 2251 / active: 1831 / done: 418 ] 9%
-- PetWithDiagnosis --
[#### waiting: 2251 / active: 1832 / done: 417 ] 9%

```



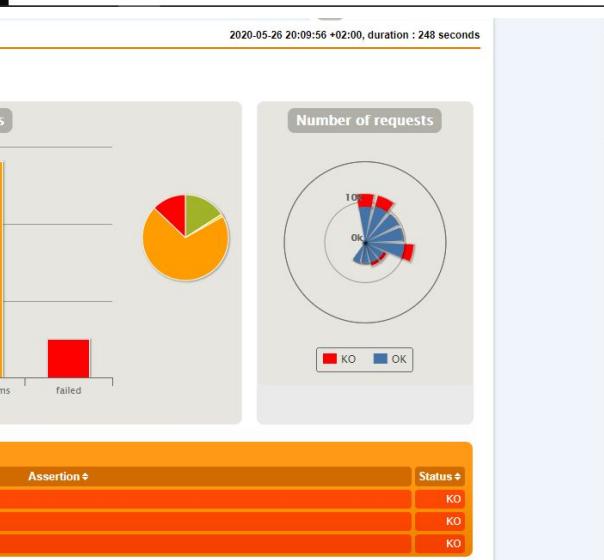
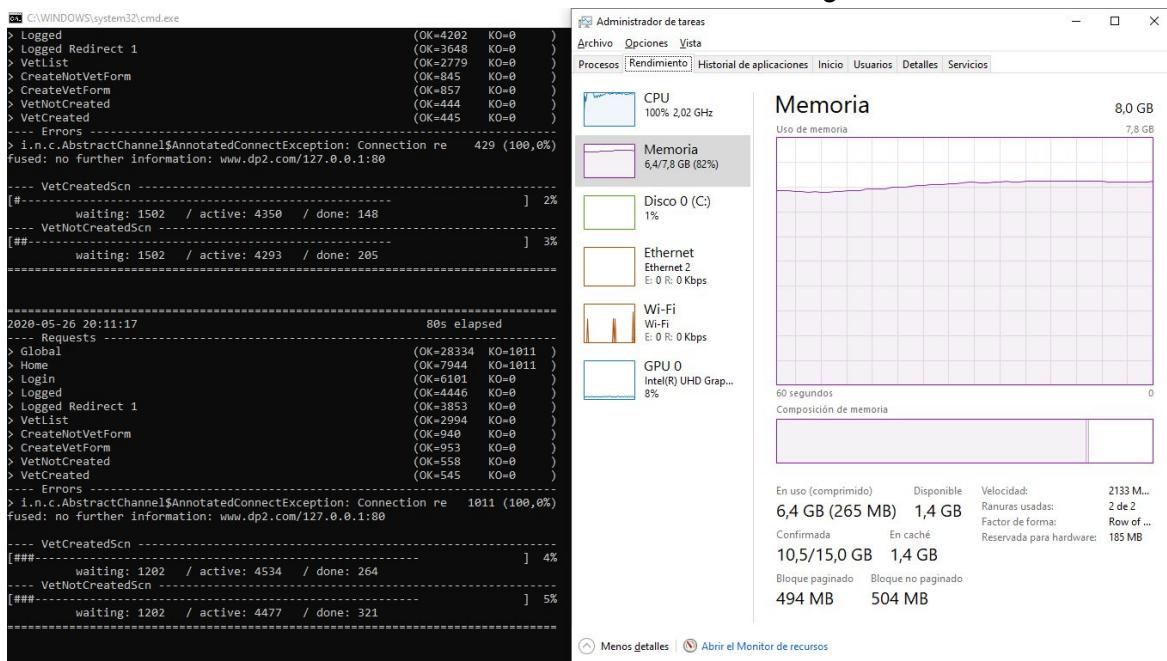
Para este caso, obtenemos un magnífico rendimiento ya que se cumplen todas las peticiones, y los asserts están en OK. Además el tiempo de respuesta media es de 0.06 segundos.

Como conclusión de la Prueba de Rendimiento de la historia 20, podemos ver que en esta prueba, es muy importante el número de segundos que pongamos para realizar las solicitudes, además del número de usuarios. Si tenemos 4500 usuarios en 150 segundos, el sistema tendrá un buen rendimiento, con respuestas muy rápidas y efectivas al 100%. Pero si metemos 7000 usuarios en 100 segundos, el sistema comenzará a fallar cuando estén 4500 de estos en activo (como en la prueba anterior), pero además, tendrá una media de respuestas muy alta, que hará que el rendimiento del sistema sea pésimo.

Historia 21	Yo COMO admin QUIERO poder crear nuevos veterinarios PARA aumentar la plantilla en caso de ser necesario.
Escenario 1	Un admin crea un veterinario.
Escenario 2	Un admin intenta crear un Vet dejando vacío el campo de username, pero sufre un error y no puede crearlo.

Mínimo número de usuarios concurrentes que no es soportado por el sistema

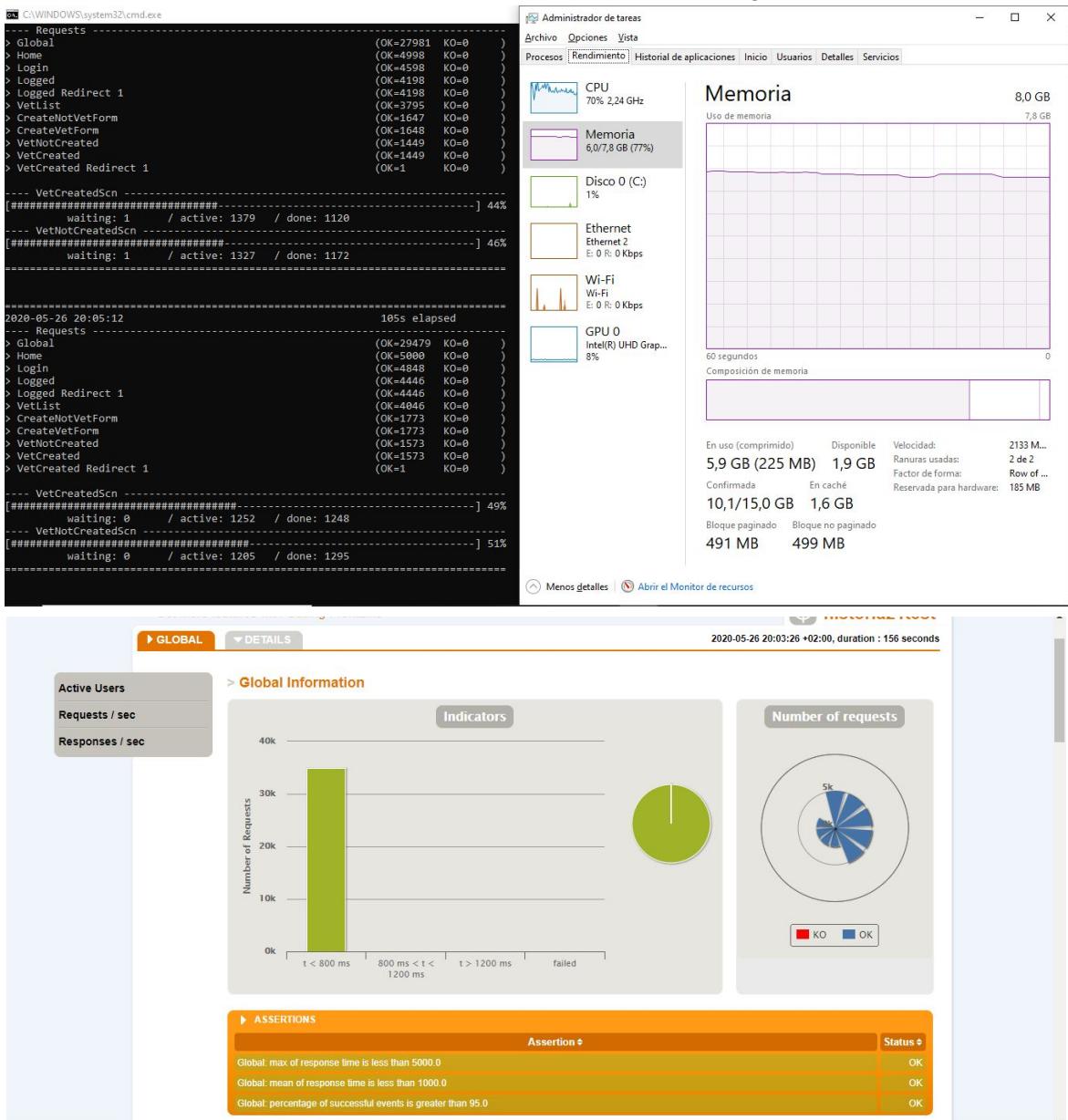
Con un número de 6000 usuarios en 100 segundos:



El tiempo de respuesta es muy alto, además la CPU y Memoria casi siempre están al 100%. Cuando el sistema tiene en activo 4500 usuarios, empieza a cometer fallos en las peticiones y no obtenemos un buen rendimiento

Máximo número de usuarios que soporta teniendo un buen rendimiento

Con un número de 2500 usuarios en 100 segundos:



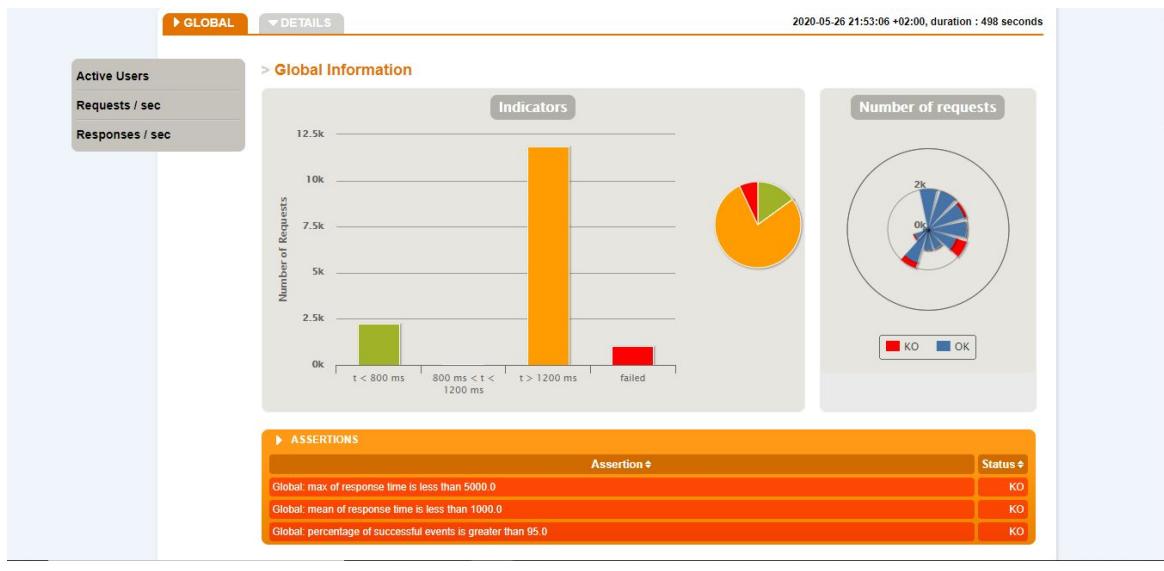
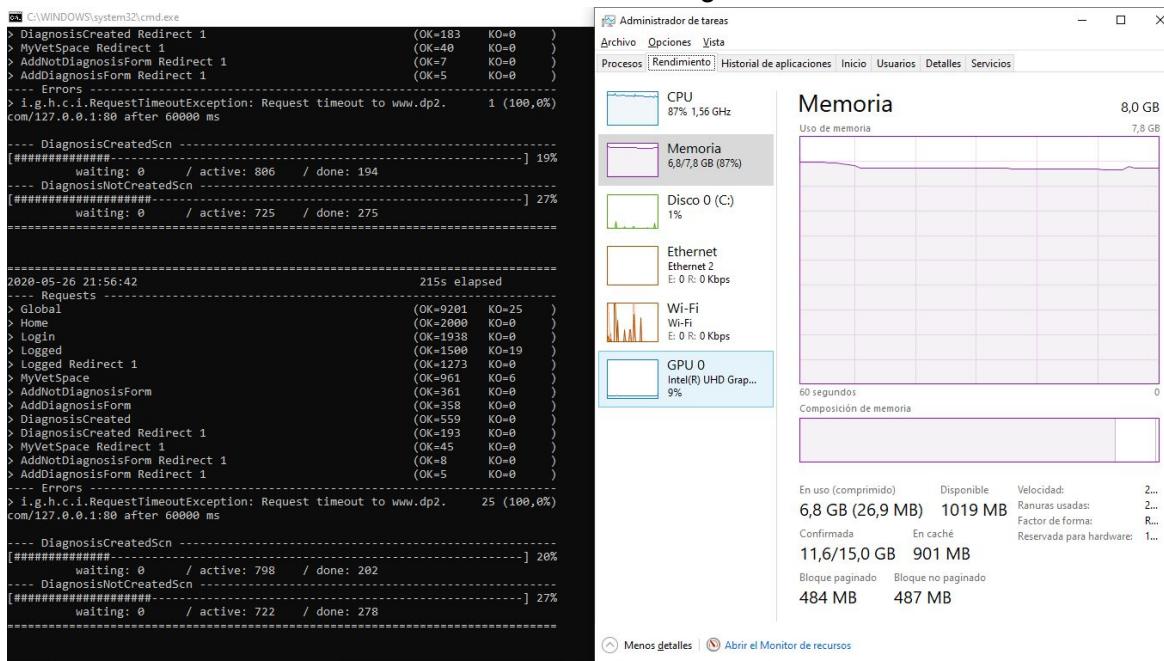
Para este caso, obtenemos un magnífico rendimiento ya que se cumplen todas las peticiones, y los asserts están en OK. Además el tiempo de respuesta media es de 0.18 segundos.

Como conclusión de la Prueba de Rendimiento de la historia 21, podemos ver que en un rango de 0-2500 usuarios (más o menos), nuestro sistema no dará ningún problema y realizará las peticiones en unos tiempos de respuesta muy bajos. Por otra parte, si en algún momento están en activo 4500 usuarios o más, el sistema empezará a dar fallos en las peticiones y su rendimiento no será nada bueno.

Historia 22	Yo COMO veterinario QUIERO crear diagnósticos de las pets que he tratado PARA que el owner pueda obtener esta información y que quede constancia de la consulta.
Escenario 1	Un Vet crea el diagnóstico de una pet tratada.
Escenario 2	Un Vet intenta crear diagnóstico de una pet pero pone como fecha en la que escribió el diagnóstico, una anterior a la fecha actual, por lo tanto no se creará y mostrará un mensaje de error.

Mínimo número de usuarios concurrentes que no es soportado por el sistema

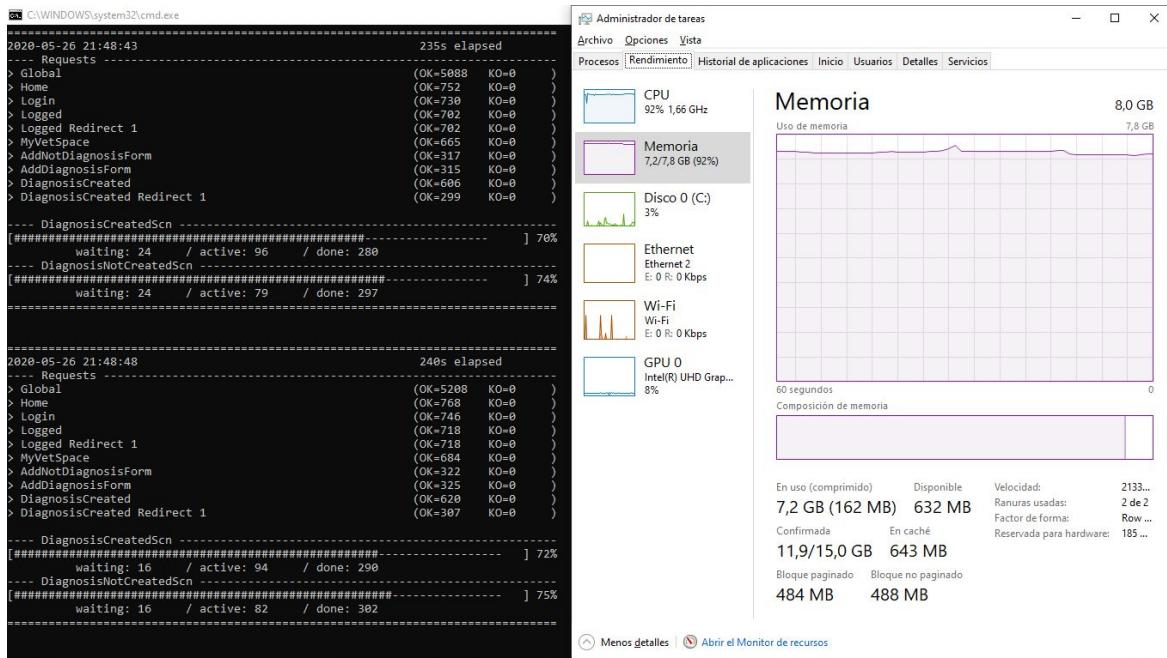
Con un número de 1000 usuarios en 150 segundos los resultados son:



Con un tiempo medio de espera de segundos. Como vemos la CPU y la memoria están casi al máximo. En el momento que se tienen en activo en torno a 800 usuarios, el sistema comienza a ir a más lento, además de unas respuesta más tardías, ofreciendo un rendimiento muy malo.

Máximo número de usuarios que soporta teniendo un buen rendimiento

Con un número de 300 usuarios en 250 segundos:



Hemos conseguido un tiempo medio de respuesta de 0,5 segundos, aunque la CPU y memoria casi siempre esta al 100%, el sistema ofrece un rendimiento bastante aceptable aunque tarda un poco más de 1 segundo en realizar algunas peticiones debido a su dificultad.

Como conclusión de la Prueba de Rendimiento de la historia 22, podemos observar que en este caso no podemos meter muchos usuarios y poco tiempo, si queremos que el sistema responda correctamente, debido a las peticiones que debe hacer, ya que si

superamos los 800 usuarios activos, el sistema comienza a tener errores en las peticiones.

Conclusiones

Luego de ejecutar todas las pruebas de rendimiento podemos saber cual es el límite operacional de nuestro sistema, que sería el mínimo número de usuarios del máximo número de usuarios concurrentes por historia de usuario , en nuestro caso serian 300 usuarios en 250 segundos que nos da unos 2 usuarios aproximados por segundo, esto viene de la historia 22.

Para poder mejorar este numero necesitamos tanto una mejor CPU como una cantidad de RAM mayor.