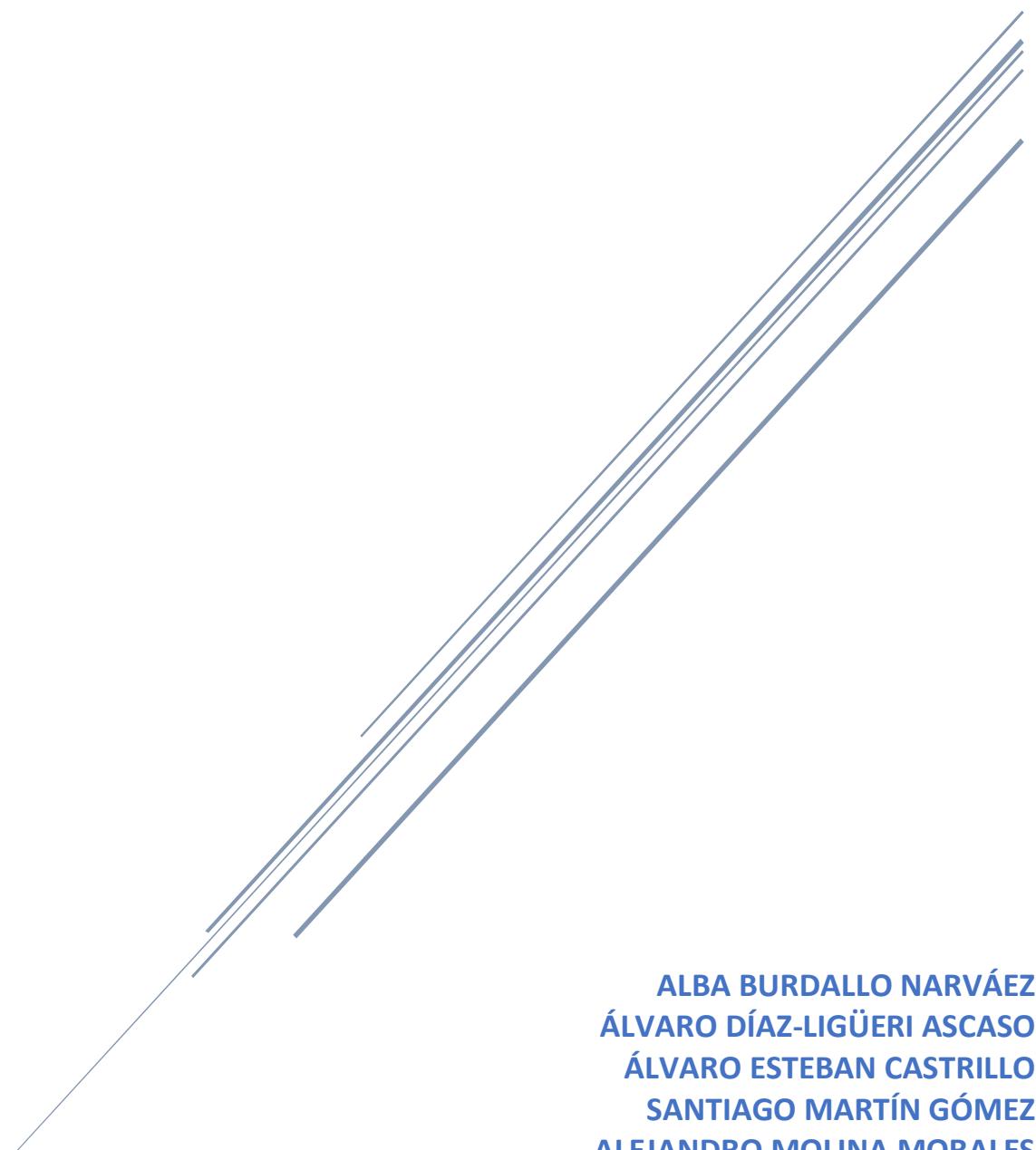


# PRUEBAS DE RENDIMIENTO

Diseño y Pruebas II



ALBA BURDALLO NARVÁEZ  
ÁLVARO DÍAZ-LIGÜERI ASCASO  
ÁLVARO ESTEBAN CASTRILLO  
SANTIAGO MARTÍN GÓMEZ  
ALEJANDRO MOLINA MORALES  
MIGUEL EMMANUEL SAAVEDRA GÓMEZ

## **PROCEDIMIENTO**

Por cada historia de usuario se han ido probando un número diferente de usuarios, analizando el rendimiento que ofrece el sistema y buscando el número máximo de usuarios para que la aplicación funcione. Una vez encontrados, hemos buscado el número mínimo de usuarios para que la aplicación no funcione.

## Álvaro Díaz-Ligüeri Ascaso

HU	Descripción
HU-001	Listado de mis viviendas
HU-004	Editar una vivienda
HU-007	Editar perfil de usuario (cliente o propietario)

## CARACTERÍSTICAS DEL SISTEMA

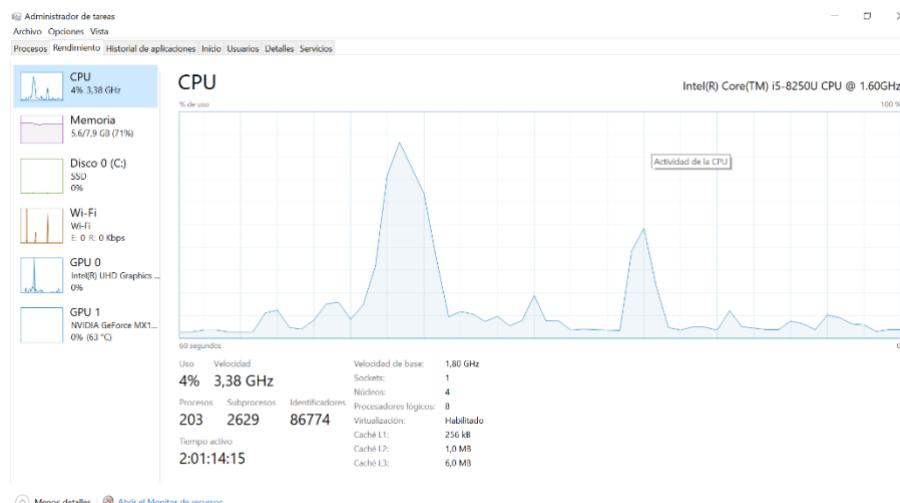
He utilizado un Xiaomi Mi Air 13, cuyas características son las siguientes.

<b>Processor(Model)/Frecuencia/Nucleos/HT</b>	Intel Core i5-8250U / 1.60GHz /4
<b>Cache level 1-D</b>	32KB *4, 8-WAY-asociative 64B line size
<b>Cache level 1-I</b>	32KB *4, 8-WAY-asociative 64B line size
<b>Cache level 2</b>	256KB *4, 4-WAY-asociative 64B line size
<b>Cache level 3</b>	6MB, 12-WAY-asociative 64B line size
<b>Memory principal</b>	DDR4 8GB
<b>Chipset</b>	Kaby Lake 08

## RESULTADOS

### HU-001

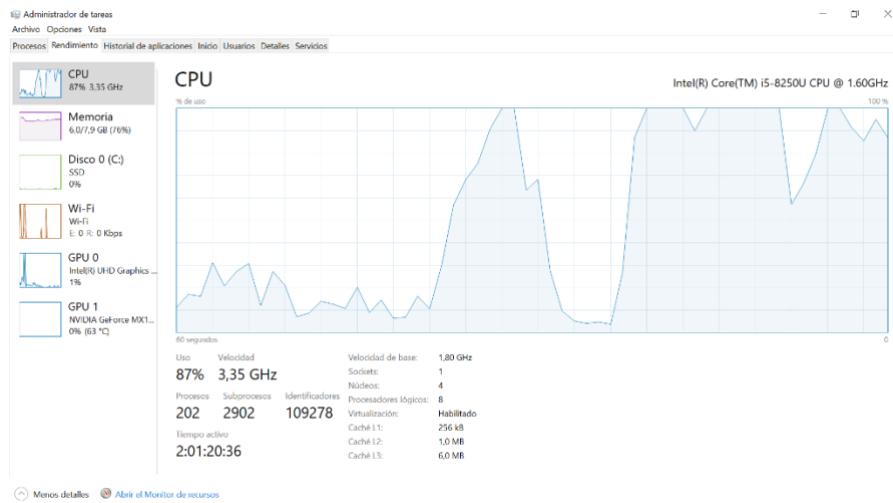
En primer lugar, he comenzado probando 1 solo usuario en 10 segundos, para comprobar el correcto funcionamiento de la prueba. El rendimiento obtenido es el siguiente:



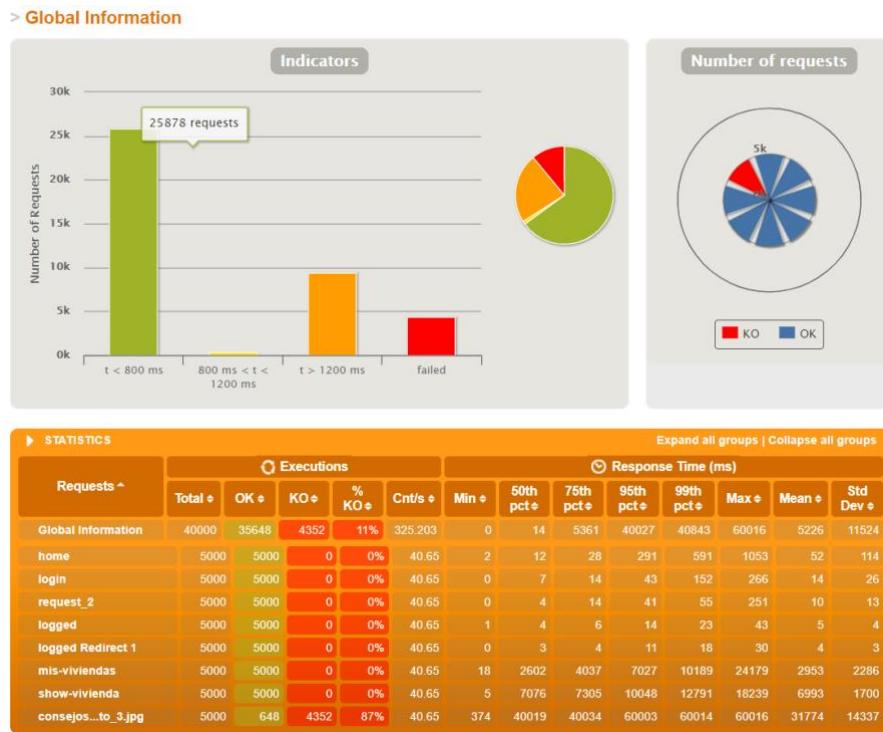


Mediante el informe que genera Gatling podemos observar que efectivamente, se realizan satisfactoriamente las peticiones.

### 5000 usuarios en 10 segundos

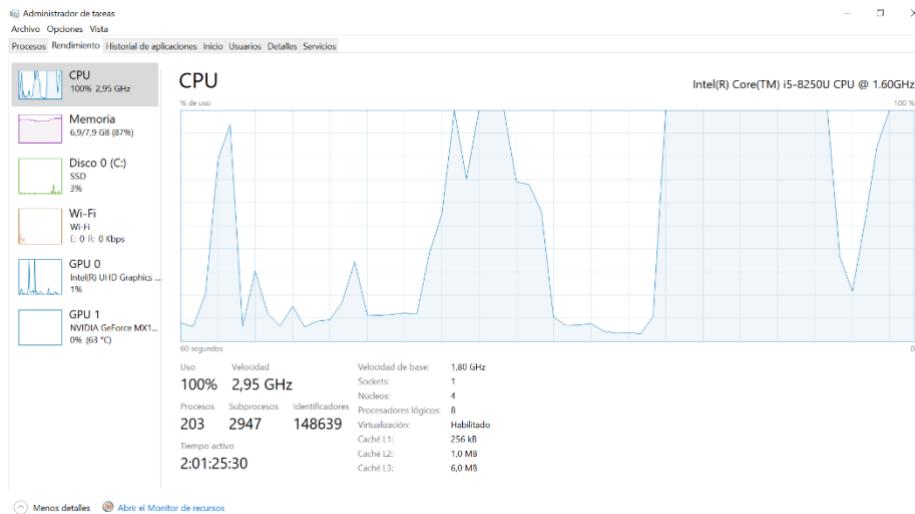


Produce un incremento de un 4% en la memoria, en cambio, el consumo de la CPU incrementa un 42%.



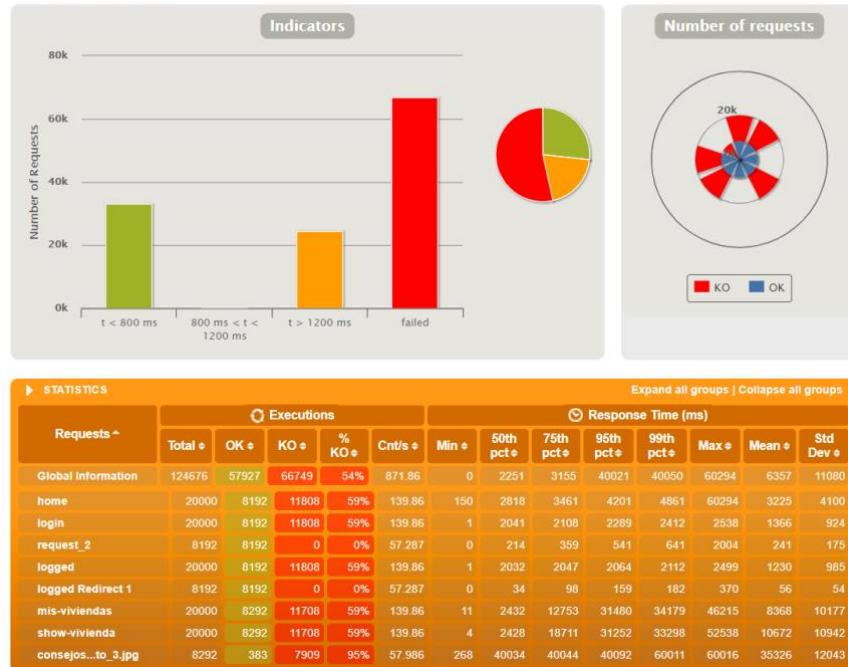
No se produce ningún fallo de petición.

## 20.000 usuarios en 10 segundos



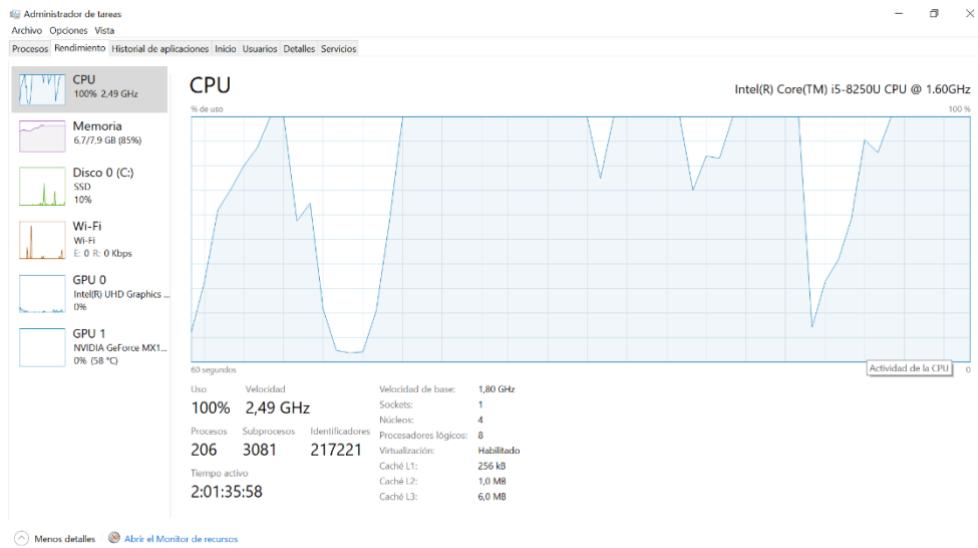
Podemos observar que consume un 100% de la CPU, pero con muchos picos.

### > Global Information



Comienza a producirse los primeros fallos en las peticiones.

### 50.000 usuarios en 10 segundos

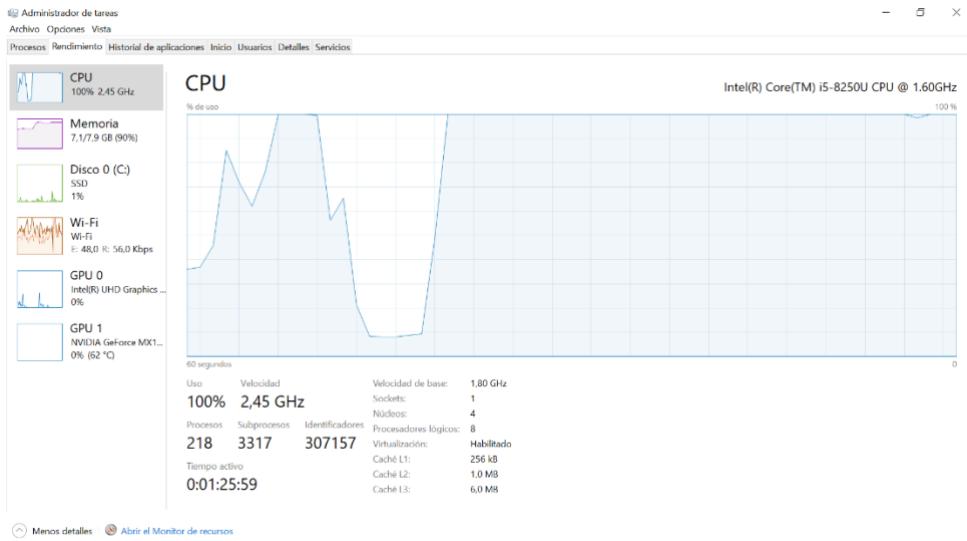


Podemos observar que consume un 100% de la CPU, pero con picos más altos que los anteriores.

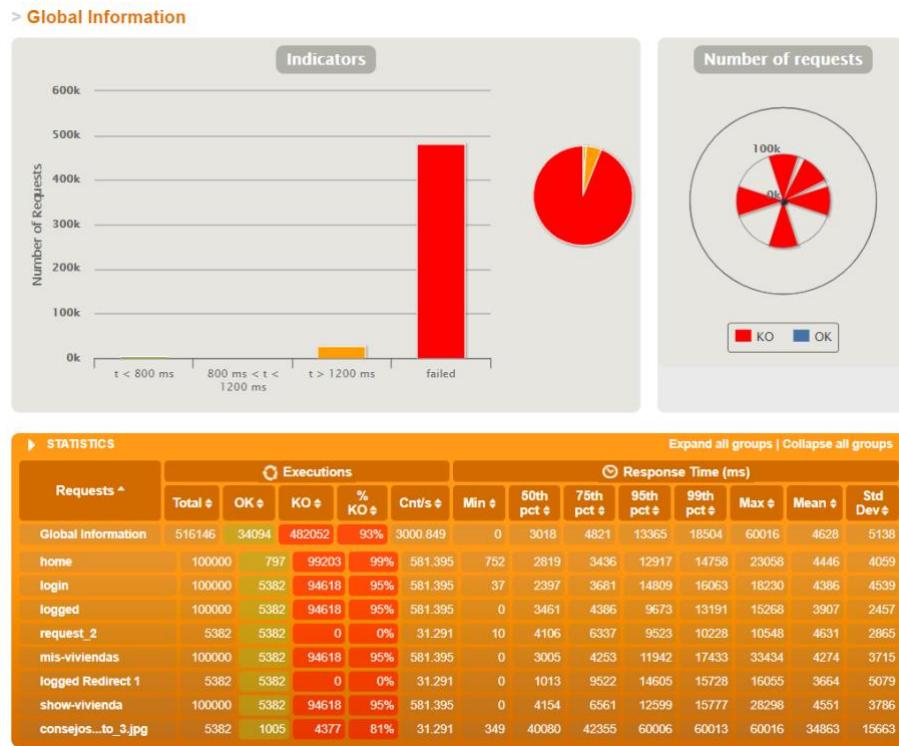


La mayoría de las peticiones producen KO. Por tanto, podemos afirmar que esta no es una buena prueba.

### 100.000 usuarios en 10 segundos



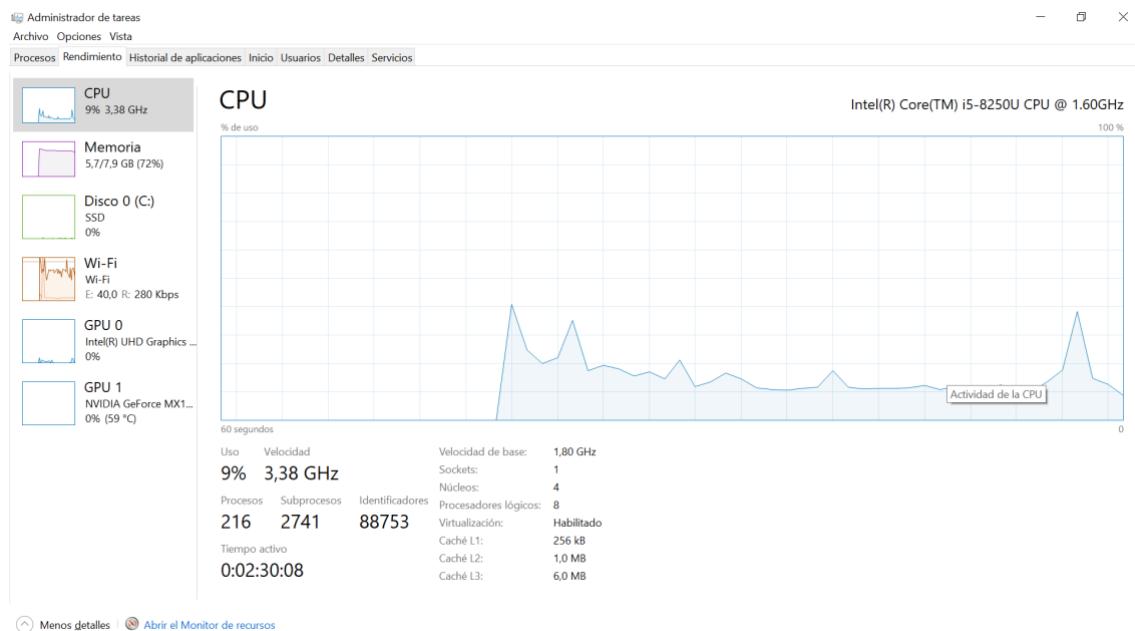
Tanto la memoria como la CPU están al máximo.



La mayoría de las ejecuciones producen fallo, también existen peticiones con un tiempo de respuesta de hasta 60 segundos, lo que indica que tampoco es una buena prueba.

## HU-004

En primer lugar, he comenzado probando 1 solo usuario en 10 segundos, para comprobar el correcto funcionamiento de la prueba. El rendimiento obtenido es el siguiente:



> Global Information

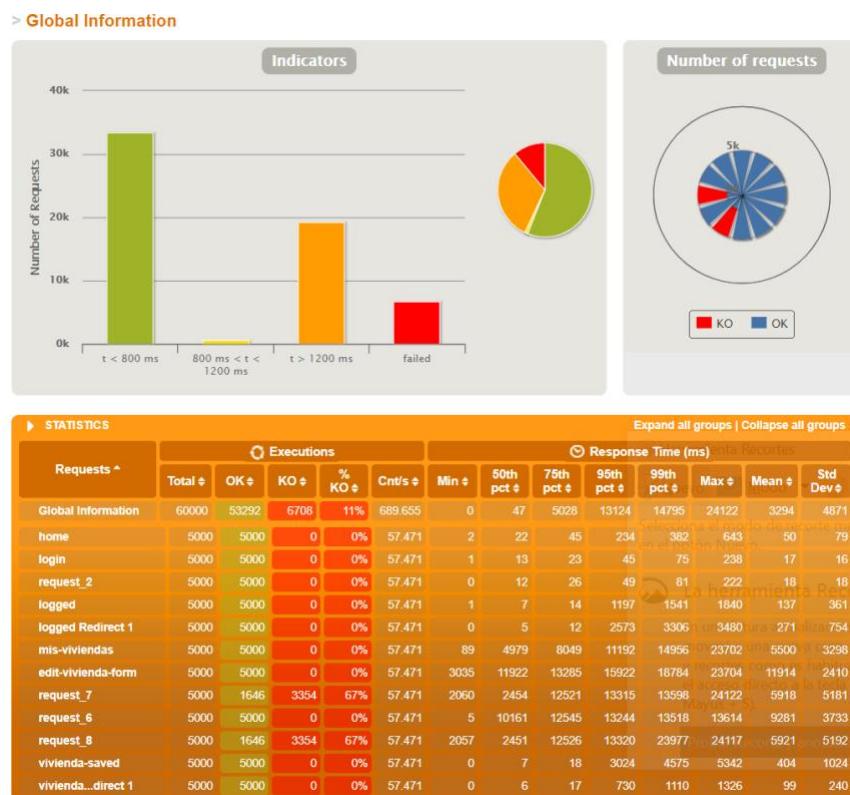


Mediante el informe que genera Gatling podemos observar que efectivamente, se realizan satisfactoriamente las peticiones.

## 5000 usuarios en 10 segundos

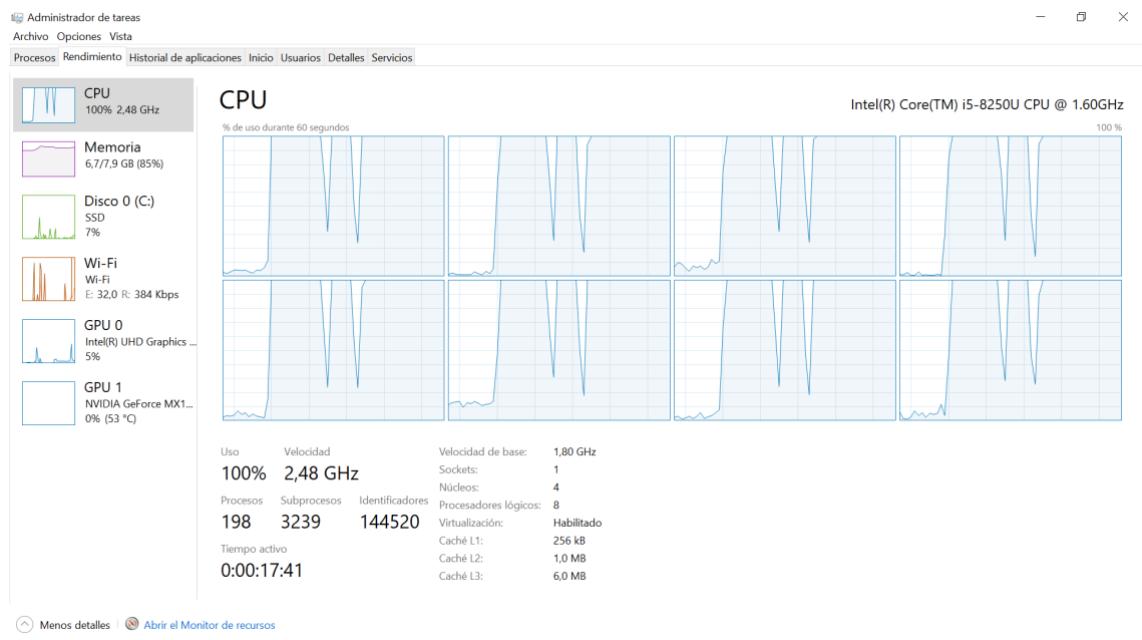


Produce un incremento de un 4% en la memoria, en cambio, el consumo de la CPU incrementa un 91%.

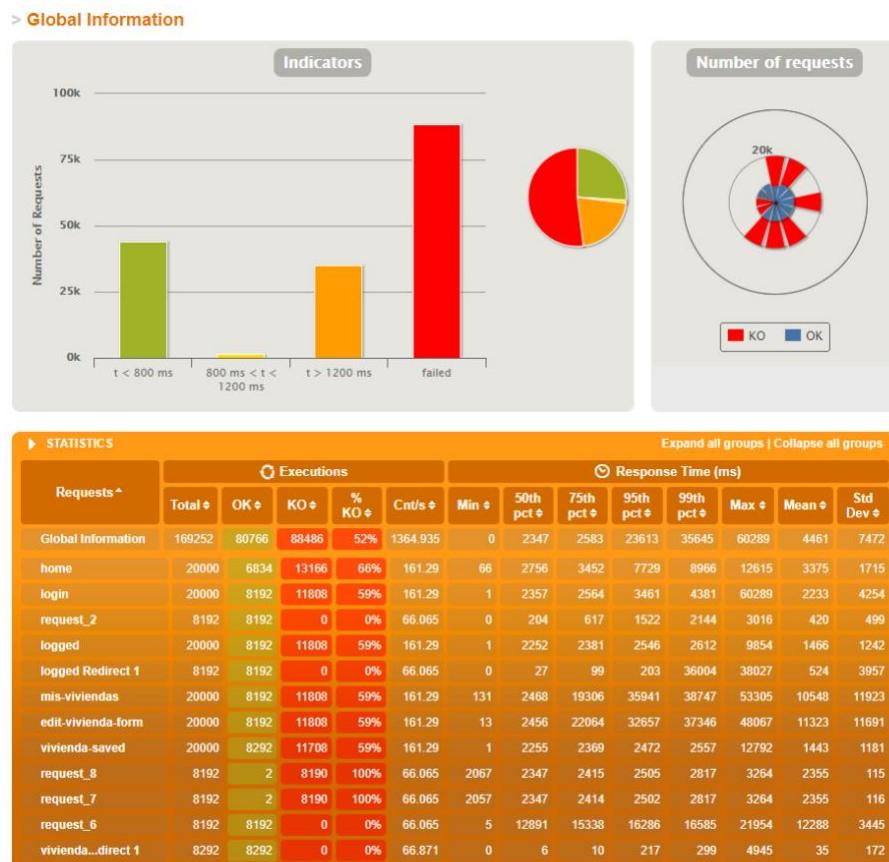


Todas las ejecuciones producen éxito, además, el tiempo de respuesta máximo es de 25 segundos. Lo que podemos considerar un buen rendimiento.

## 20.000 usuarios en 10 segundos

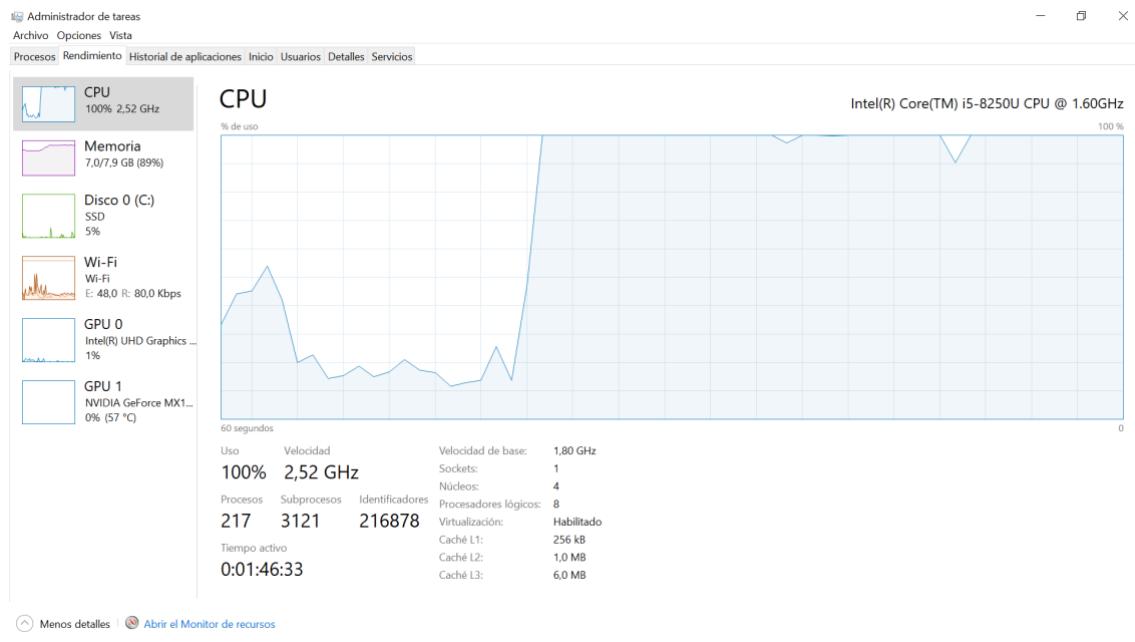


Podemos observar un gran aumento en la memoria, mientras que en la CPU consume el 100%.



Para la gran mayoría de los usuarios, las pruebas producen fallo, por tanto no es una buena prueba.

### 50.000 usuarios en 10 segundos



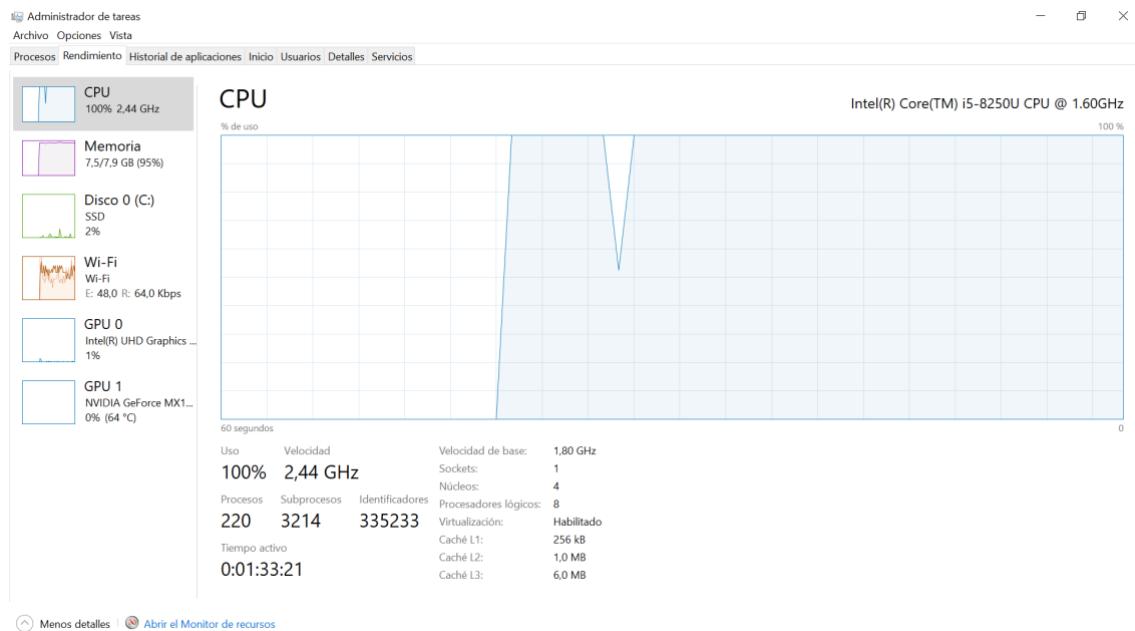
La CPU sigue rindiendo al 100%, mientras que el consumo de la memoria tiene un pequeño ascenso.

> Global Information



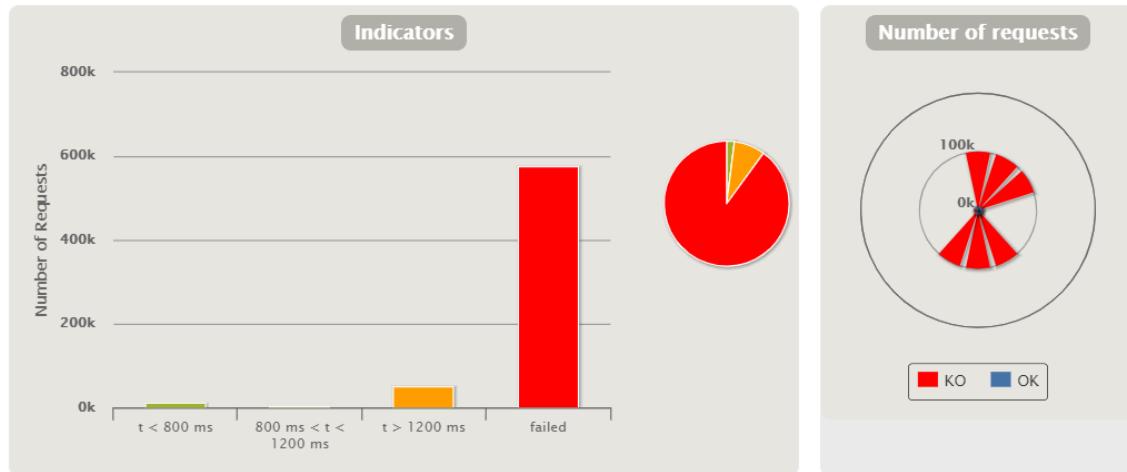
Los tiempos de respuesta ascienden hasta los 61 segundos, tenemos un 76% de KO. Por tanto, tampoco es una buena prueba.

### 100.000 usuarios en 10 segundos



Produce el rendimiento esperado, pues es el máximo número de usuarios que puede soportar el ordenador.

#### > Global Information

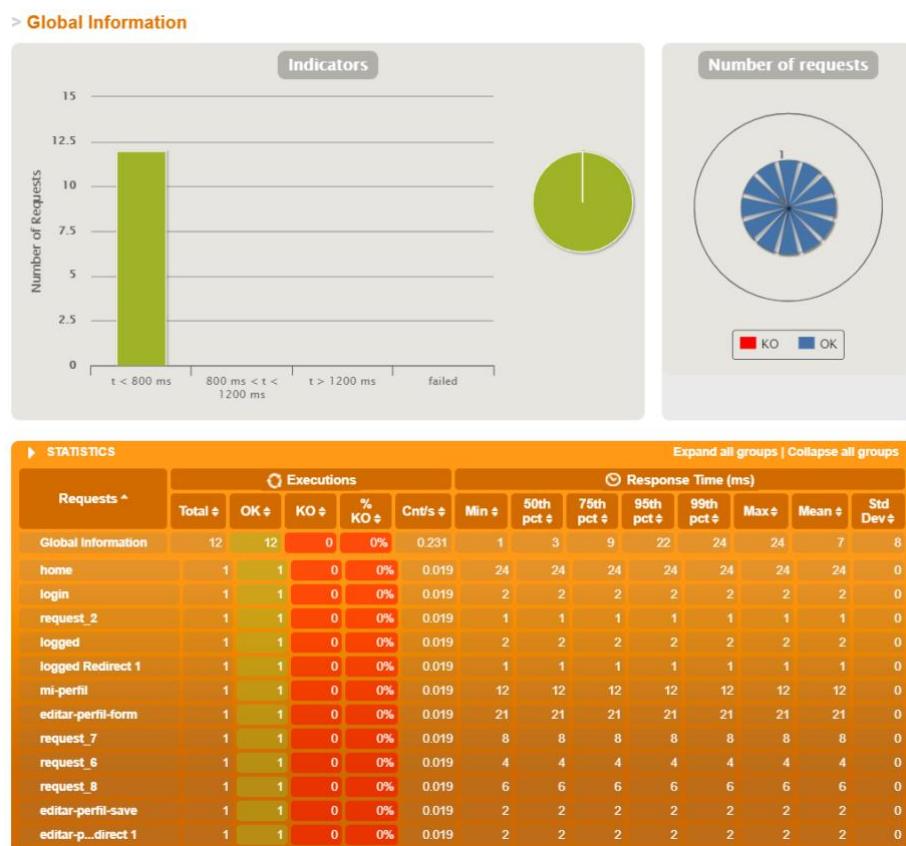
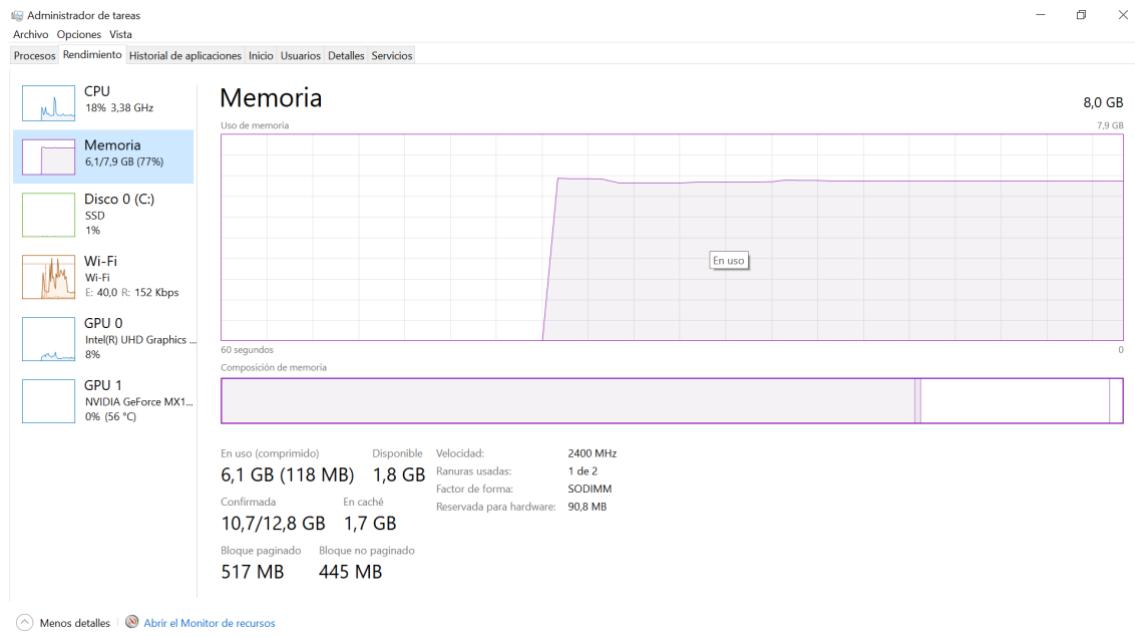


STATISTICS													<a href="#">Expand all groups</a>		<a href="#">Collapse all groups</a>	
Requests ^	Executions					Response Time (ms)								<a href="#">Max ^</a>	<a href="#">Mean ^</a>	<a href="#">Std Dev ^</a>
	Total ♦	OK ♦	KO ♦	% KO ♦	Cnt/s ♦	Min ♦	50th pct ♦	75th pct ♦	95th pct ♦	99th pct ♦	Max ♦					
Global Information	642695	65692	577003	90%	4284.633	0	3495	5322	13969	22434	42187	4622	4482			
home	100000	1649	98351	98%	666.667	75	4613	6659	14959	18598	27514	6627	4443			
login	100000	7110	92890	93%	666.667	156	2949	4325	13845	19265	27517	4158	3867			
logged	100000	7117	92883	93%	666.667	5	5058	6111	12159	15507	26135	5573	3267			
request_2	7110	7110	0	0%	47.4	927	4165	9650	12332	13165	13687	5711	4148			
logged Redirect 1	7117	7117	0	0%	47.447	3	2346	13962	26466	27714	27995	8303	8977			
mis-viviendas	100000	7117	92883	93%	666.667	0	1319	3255	14532	27050	42187	3395	5413			
edit-vivienda-form	100000	7117	92883	93%	666.667	0	3499	5250	13965	23874	40510	4935	4705			
vivienda-saved	100000	7117	92883	93%	666.667	0	1872	3221	10300	12929	15551	2762	2953			
request_8	7117	2	7115	100%	47.447	2	3690	6282	9605	10775	12448	3986	3037			
request_7	7117	2	7115	100%	47.447	2	3690	6267	9604	10772	12448	3983	3035			
request_6	7117	7117	0	0%	47.447	5	10943	12363	14439	16421	25806	9064	4378			
vivienda...direct 1	7117	7117	0	0%	47.447	0	14	1099	3355	3995	12807	620	1049			

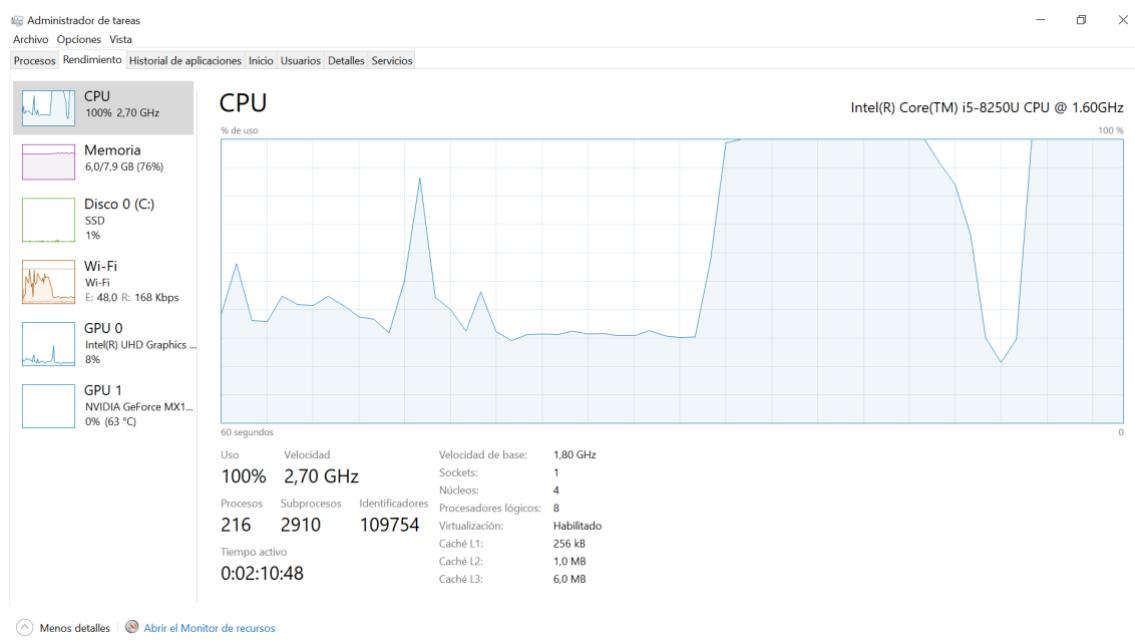
Para la gran mayoría de usuarios el número de solicitudes ha fallado y para los otros, ha tenido un tiempo de más de 15 segundos. Hemos tenido tiempos máximos de hasta 42 segundos lo que no nos indica un buen rendimiento.

HU-007

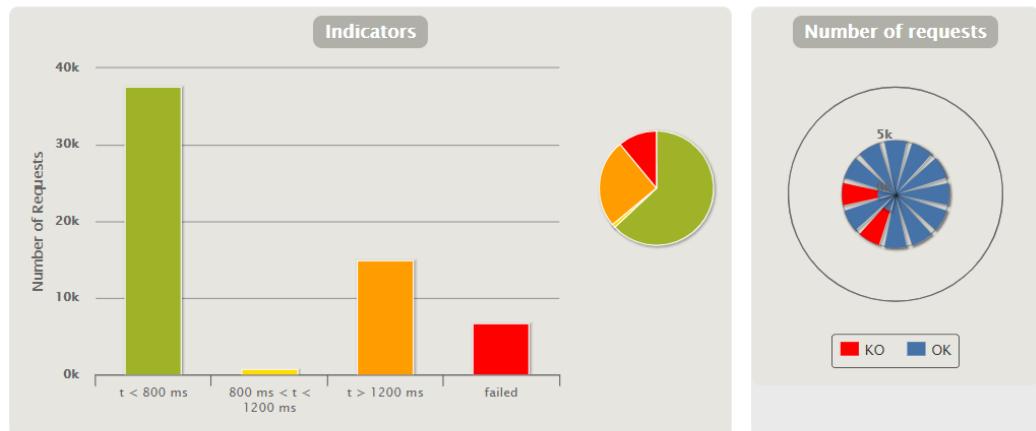
En primer lugar, he comenzado probando 1 solo usuario en 10 segundos, para comprobar el correcto funcionamiento de la prueba. El rendimiento obtenido es el siguiente:



## 5000 usuarios en 10 segundos



### > Global Information



▶ STATISTICS

Expand all groups | Collapse all groups

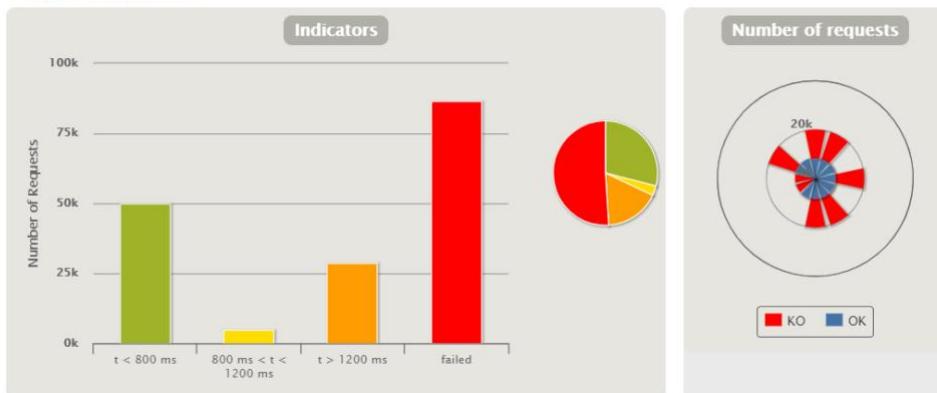
Requests ▲	Executions					Response Time (ms)								
	Total ▲	OK ▲	KO ▲	% KO ▲	Cnt/s ▲	Min ▲	50th pct ▲	75th pct ▲	95th pct ▲	99th pct ▲	Max ▲	Mean ▲	Std Dev ▲	
Global Information	60000	53292	6708	11%	674.157	0	52	2426	12832	14102	26743	2468	4248	
home	5000	5000	0	0%	56.18	3	186	803	1893	2141	2306	495	631	
login	5000	5000	0	0%	56.18	1	42	131	254	378	508	81	88	
request_2	5000	5000	0	0%	56.18	0	27	140	277	329	721	79	92	
logged	5000	5000	0	0%	56.18	1	8	15	34	68	97	12	12	
logged Redirect 1	5000	5000	0	0%	56.18	0	6	14	30	54	84	10	11	
mi-perfil	5000	5000	0	0%	56.18	1	9	20	3160	5433	7255	328	1192	
editar-perfil-form	5000	5000	0	0%	56.18	465	9731	12777	14467	15296	17614	8942	4377	
request_7	5000	1646	3354	67%	56.18	2020	2432	7570	13809	15171	26687	5060	4593	
request_6	5000	5000	0	0%	56.18	5	9194	11979	13798	14224	15019	8921	3552	
request_8	5000	1646	3354	67%	56.18	2030	2429	7570	13795	15030	26743	5056	4587	
editar-perfil-save	5000	5000	0	0%	56.18	1	6	10	5262	7696	8173	535	1569	
editar-p...direct 1	5000	5000	0	0%	56.18	0	5	8	831	1852	1957	98	305	

Comienzan a producirse los primeros fallos, pero el rendimiento sigue siendo bastante bueno. Los tiempos de respuesta no superan los 2600ms.

## 20.000 usuarios en 10 segundos



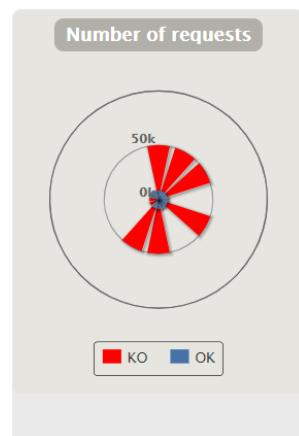
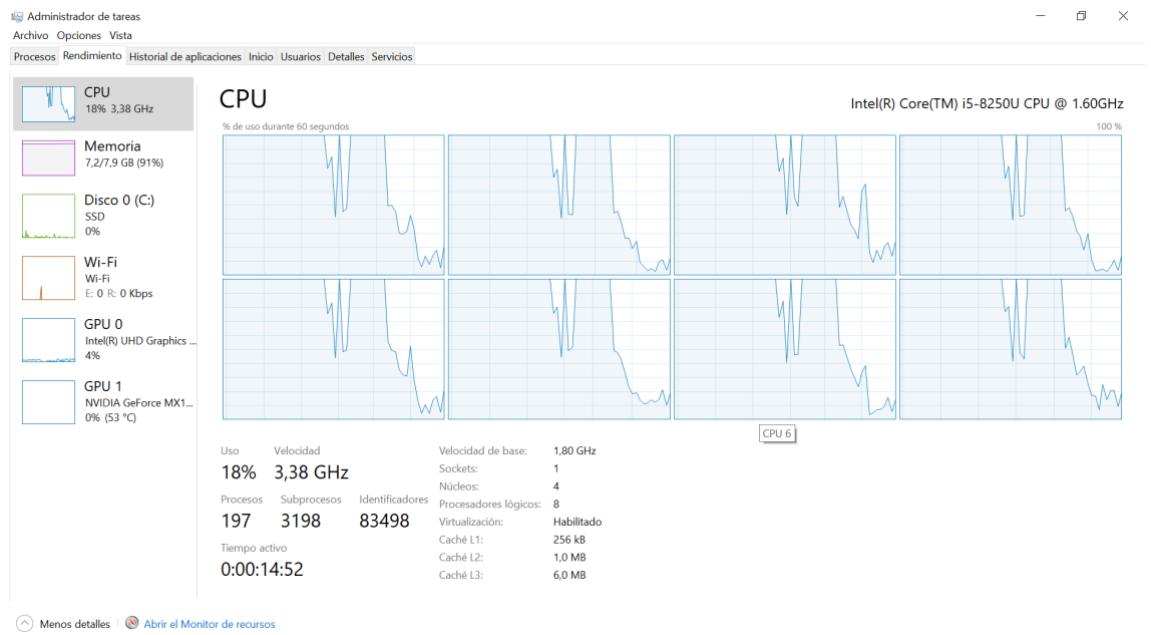
### > Global Information



Requests	STATISTICS										Expand all groups   Collapse all groups			
	Executions					Response Time (ms)								
Requests	Total	OK	KO	% KO	Cnt/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev	
Global Information	169946	83408	86538	51%	1465.052	0	2051	2423	9487	15190	60965	2313	3274	
home	20000	8192	11808	59%	172.414	2	1611	2667	4598	7159	60965	2266	4338	
login	20000	8292	11708	59%	172.414	1	1049	2256	3195	3716	4916	1403	995	
request_2	8292	8292	0	0%	71.483	0	601	1298	2491	3998	4673	838	805	
logged	20000	8292	11708	59%	172.414	1	2033	2054	2176	2275	2487	1258	971	
logged Redirect 1	8292	8292	0	0%	71.483	0	67	217	347	357	426	119	123	
mi-perfil	20000	8292	11708	59%	172.414	0	2092	2321	2477	2591	3137	1457	1004	
editar-perfil-form	20000	8292	11708	59%	172.414	3	2490	6118	14900	16363	17446	4778	4294	
request_6	8292	8292	0	0%	71.483	2	10475	13651	15550	16006	16099	10090	4311	
request_8	8292	100	8192	99%	71.483	3	2365	2454	2581	2677	2983	2336	288	
request_7	8292	100	8192	99%	71.483	3	2367	2454	2582	2676	2981	2337	288	
editar-perfil-save	20000	8486	11514	58%	172.414	0	2225	2395	6453	9760	10184	1879	1968	
editar-p...direct 1	8486	8486	0	0%	73.155	0	6	15	2497	3025	3334	231	711	

Podemos observar que tenemos un 59% de pruebas que se ejecutan incorrectamente y los tiempos de respuesta son inferiores a 800ms para las que han dado éxito.

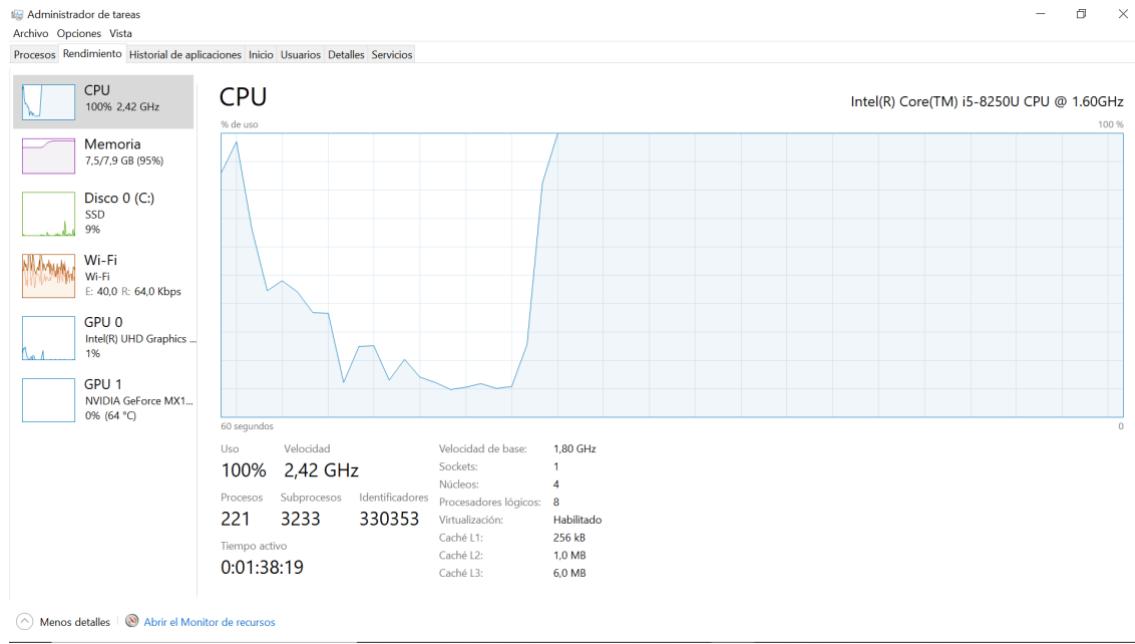
## 50.000 usuarios en 10 segundos



Requests ^	STATISTICS										Expand all groups   Collapse all groups			
	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	645399	68740	576659	89%	4745.581	0	2581	4040	13016	16341	21155	3812	3820	
home	100000	813	99187	99%	735.294	107	3413	4169	13732	14738	20951	5066	3975	
login	100000	7302	92698	93%	735.294	11	2200	3078	11793	13345	17919	3467	3225	
request_2	7302	7302	0	0%	53.691	16	2007	3155	4739	5088	18247	2358	1344	
logged	100000	7609	92391	92%	735.294	0	1377	2252	13129	16762	19724	2875	3903	
logged Redirect 1	7609	7609	0	0%	55.949	4	1169	2393	10611	10717	18703	2445	2753	
mi-perfil	100000	7617	92383	92%	735.294	0	1796	3198	12600	15012	19908	3352	3894	
editar-perfil-form	100000	7622	92378	92%	735.294	0	2608	3760	13071	17288	21155	3744	3711	
request_8	7622	0	7622	100%	56.044	0	3776	5366	8765	10515	12475	3908	2568	
request_7	7622	0	7622	100%	56.044	0	3778	5364	8761	10514	12672	3905	2567	
request_6	7622	7622	0	0%	56.044	6	9774	12985	14074	14762	16696	9271	3874	
editar-perfil-save	100000	7622	92378	92%	735.294	0	3627	5137	13176	17693	19918	4413	3773	
editar-p...direct 1	7622	7622	0	0%	56.044	0	8	26	3272	4381	10284	357	962	

La gran mayoría de las peticiones han fallado, aunque los tiempos de respuesta siguen siendo bastante buenos. En cuanto al rendimiento, sigue siendo bastante elevado, dado la gran cantidad de usuarios.

## 100.000 usuarios en 10 segundos



Tan solo un 11% de las peticiones han llegado a éxito. Los tiempos de respuesta van desde 4ms hasta 16342ms. Tanto la CPU como la memoria rinden al máximo posible.

## CONCLUSIÓN

Dadas las características de mi ordenador, puedo afirmar que tanto 100.000 como 50.000 peticiones a la vez son demasiadas. No es que son buenas pruebas debido a que el % de peticiones KO supera a las de éxito.

Al ejecutar 20.000 usuarios se produce porcentaje de fallos un poco mayor que el de aciertos, por tanto, sigue sin ser unas buenas pruebas. Pero podemos obtener información importante acerca del rendimiento que ofrece el ordenador.

En base a los resultados obtenidos, el número máximo de usuarios para que no se produzcan fallos podría ser 15.000.

## Alba Burdallo Narváez

HU	Descripción
HU-008	Filtrar viviendas
HU-020	Borrar una vivienda
HU-023	Borrar un usuario por completo

## CARACTERÍSTICAS DEL SISTEMA

He utilizado un Asus R510V, cuyas características son las siguientes.

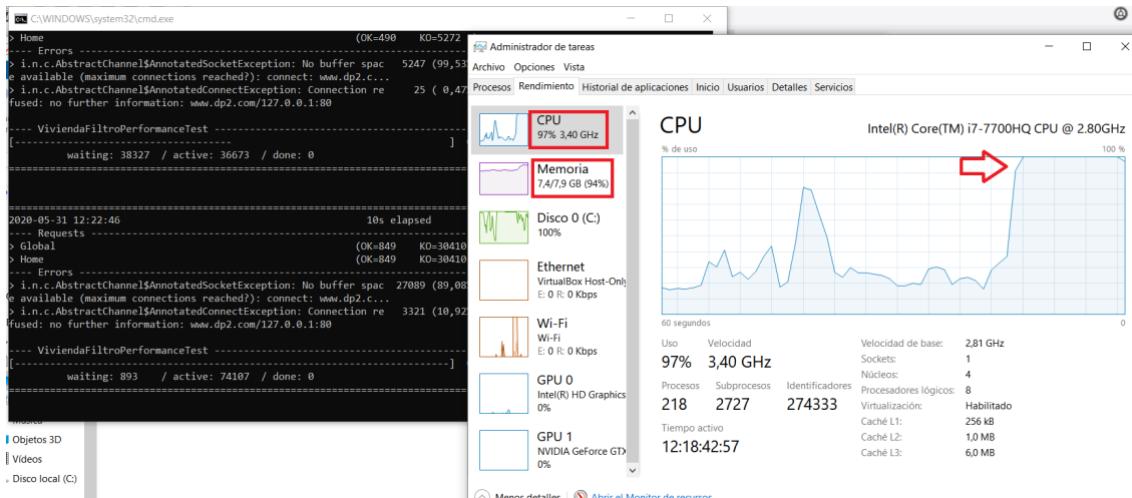
<b>Processor(Model)/Frecuency/Nucleos</b>	Intel Core i7-7700HQ / 2.80GHz
<b>Memory principal</b>	8 Gb

## RESULTADOS

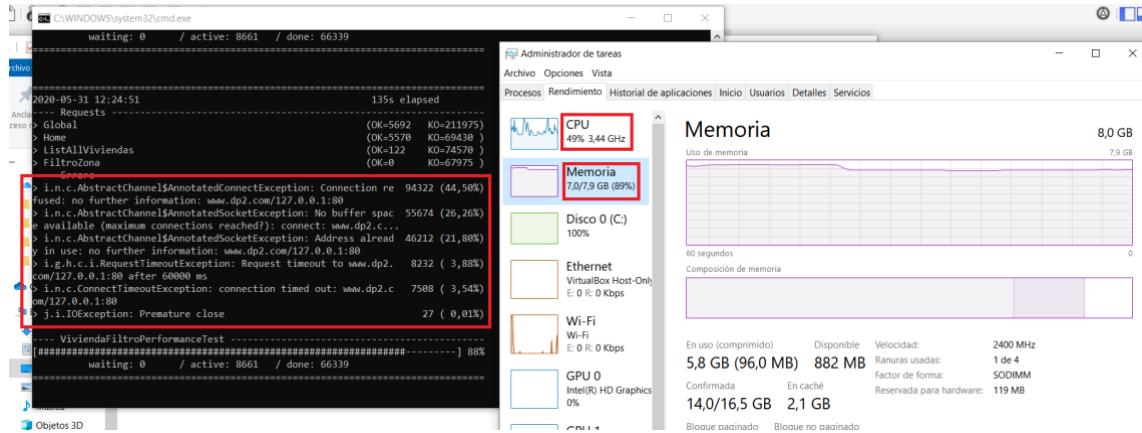
### HU-008: Filtrado de viviendas

75000 usuarios en 10 segundos

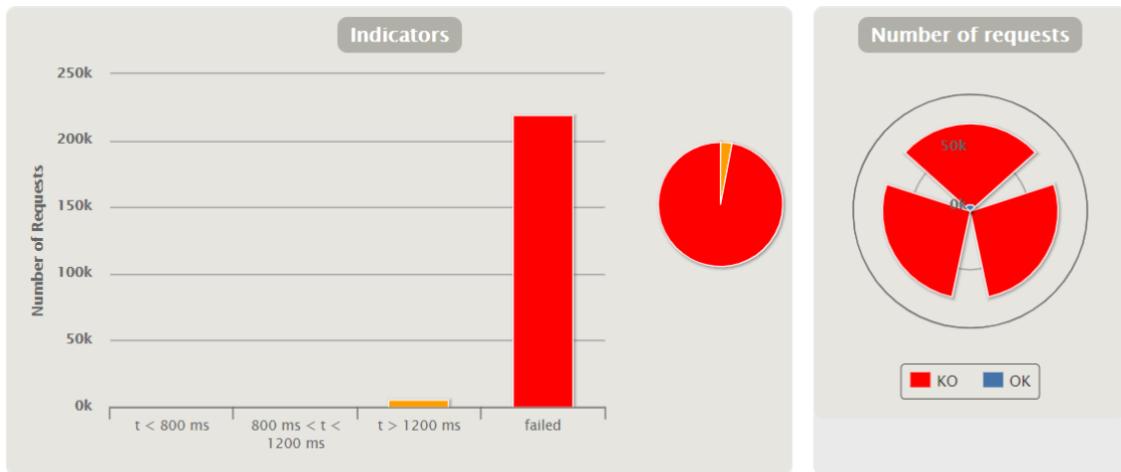
Podemos ver que nada más empezar la ejecución, la CPU y la memoria se ponen al máximo.



Aunque la CPU se va relajando a medida que se va avanzando, podemos ver que la memoria no y, además, tenemos muchos errores.



En las estadísticas de Gatling podemos observar lo siguiente:

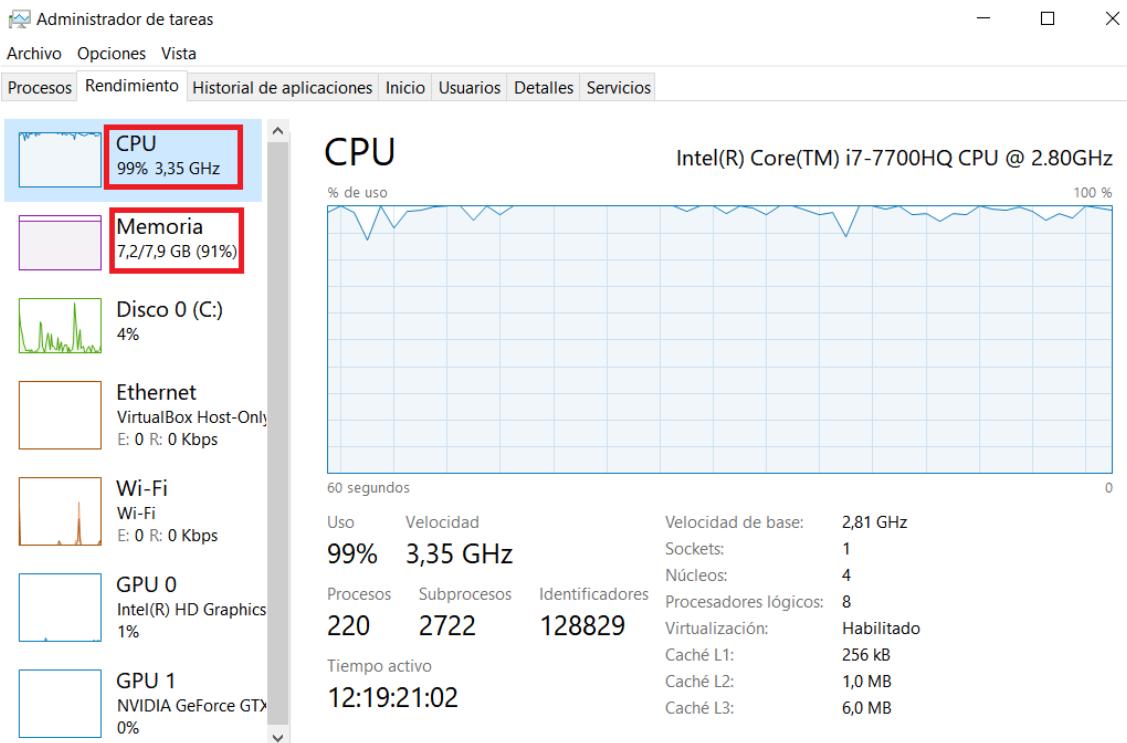


STATISTICS													<a href="#">Expand all groups</a>   <a href="#">Collapse all groups</a>			
Requests ^	Executions					Response Time (ms)										
	Total ^	OK ^	KO ^	% KO ^	Cnt/s ^	Min ^	50th pct ^	75th pct ^	95th pct ^	99th pct ^	Max ^	Mean ^	Std Dev ^			
Global Information	225000	5692	219308	97%	839.552	0	2050	3029	59533	60207	64628	5563	12999			
Home	75000	5570	69430	93%	279.851	68	3946	6976	13506	61564	64628	6761	10312			
ListAllViviendas	75000	122	74878	100%	279.851	0	2044	2194	60007	60071	60413	6126	16028			
FiltroZona	75000	0	75000	100%	279.851	0	2026	2048	2274	60014	60169	3802	11785			

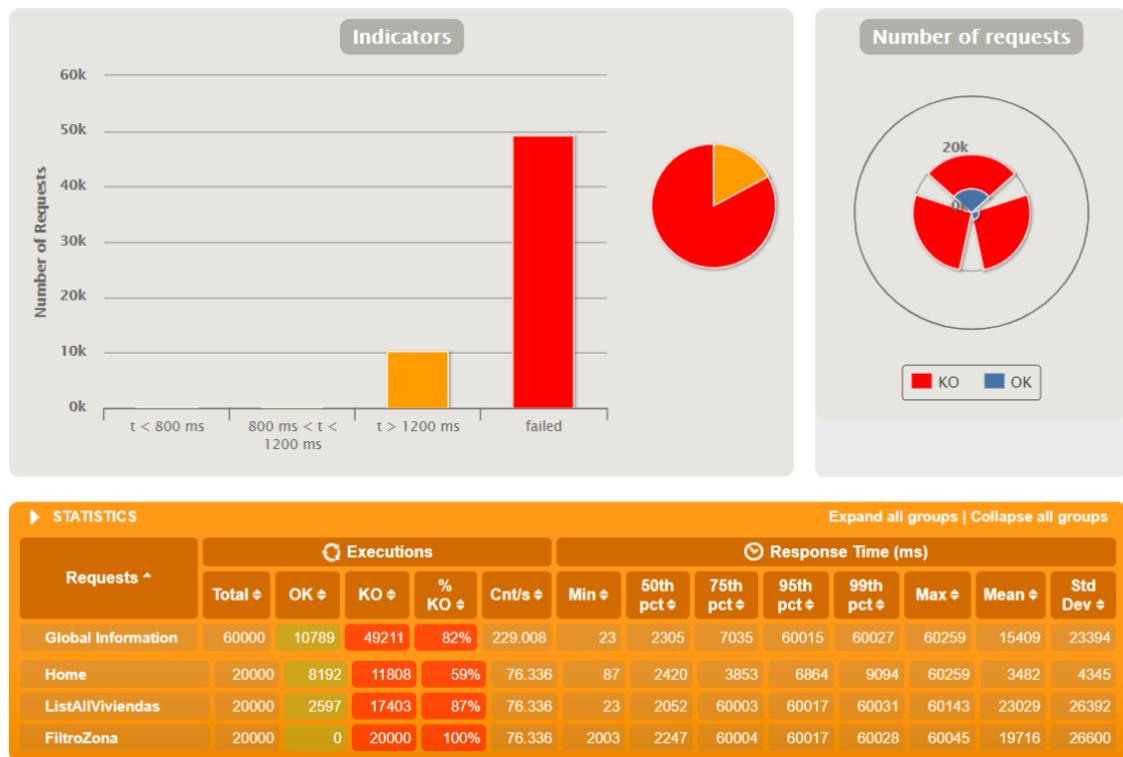
Para la gran mayoría de usuarios el número de solicitudes ha fallado y para los otros, ha tenido un tiempo de más de 1200ms. Hemos tenido tiempos máximos de hasta 64 segundos lo que no nos indica un buen rendimiento.

### 20000 usuarios en 10 segundos

Al iniciarla, la CPU y la memoria se ponen al máximo.

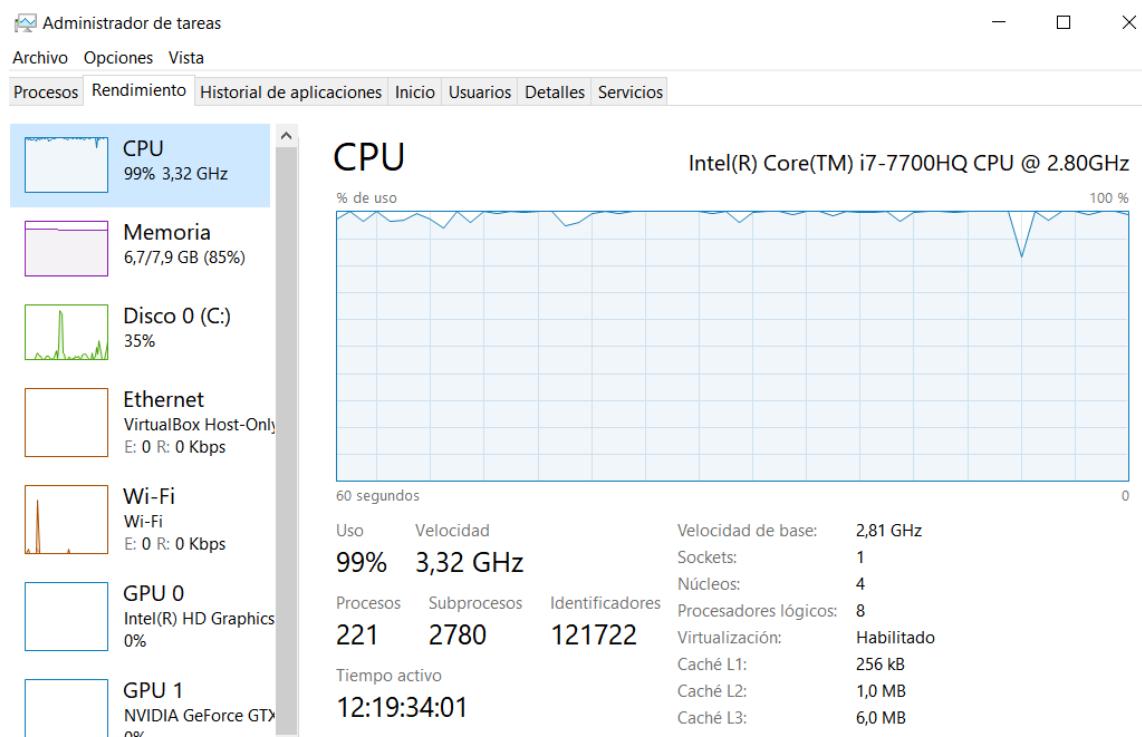


En las estadísticas de Gatling podemos ver que esta prueba tampoco sería válida.



## 5000 usuarios en 10 segundos

Para este número de usuarios nos encontramos que la CPU sigue estando al máximo, pero esta vez de una forma más estable.

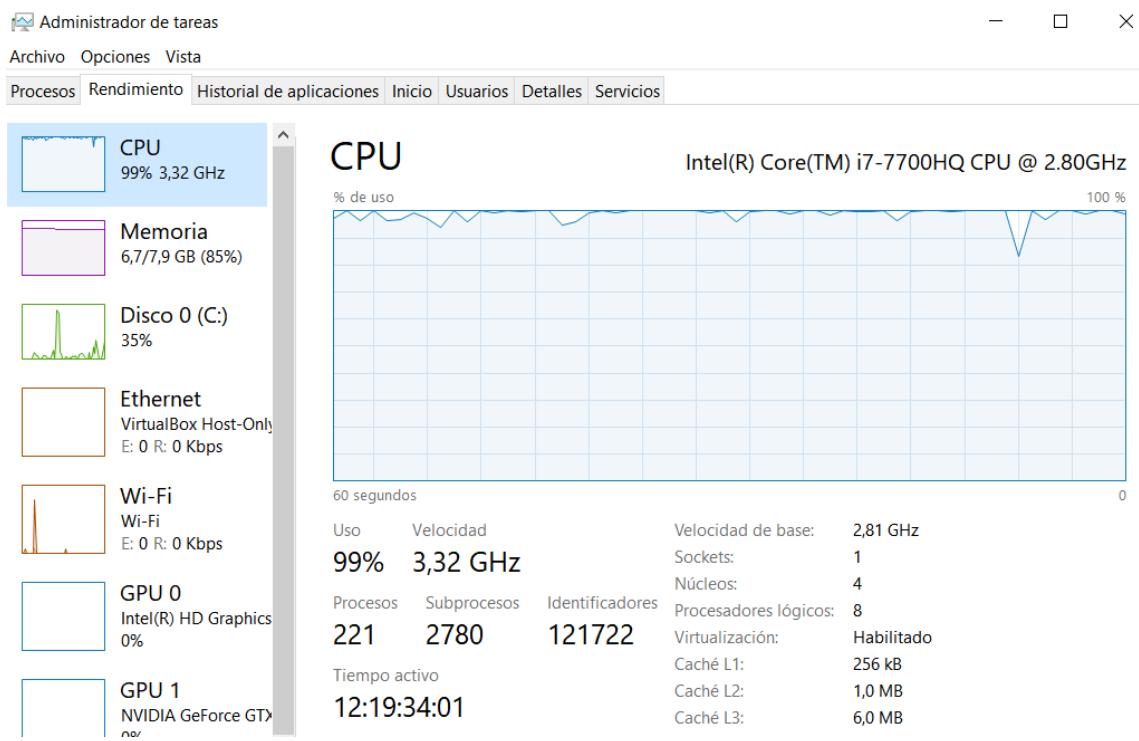


En las estadísticas de Gatling nos encontramos que la mayoría de usuarios consiguen entrar.

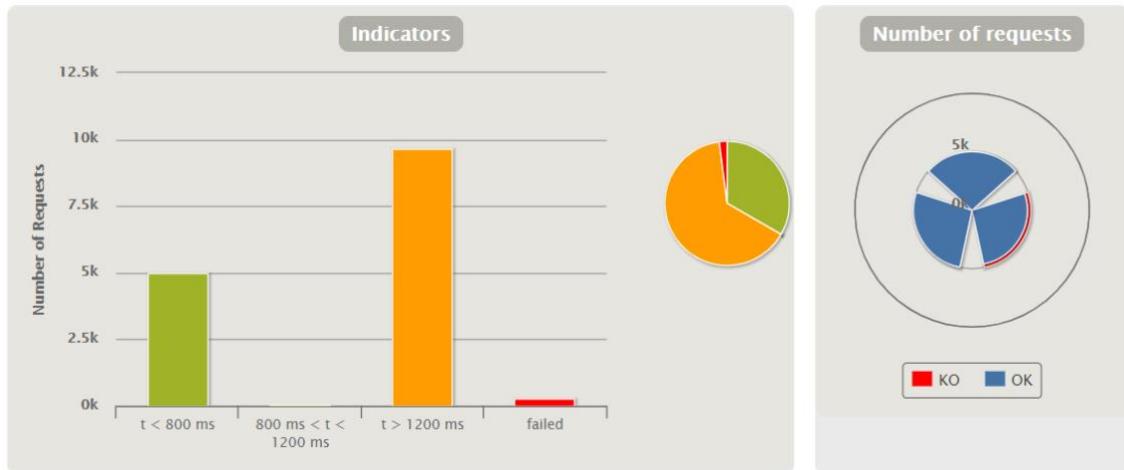


En esta gráfica podemos ver que solo un número muy reducido de peticiones han fallado en comparación con las que sí han obtenido respuesta.

Para este número de usuarios nos encontramos que la CPU sigue estando al máximo, pero esta vez de una forma más estable.



En las estadísticas de Gatling nos encontramos que la mayoría de los usuarios consiguen entrar.



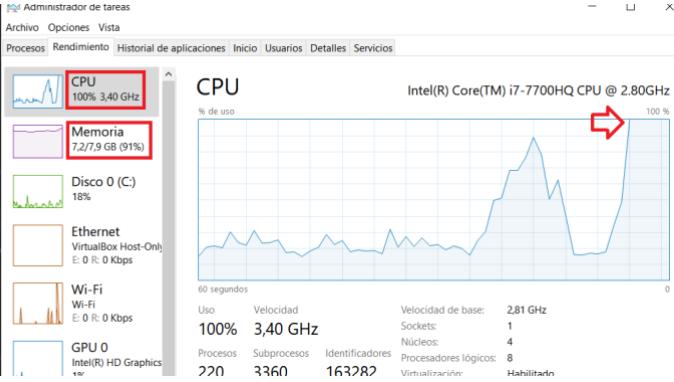
En esta gráfica podemos ver que solo un número muy reducido de peticiones han fallado en comparación con las que sí han obtenido respuesta.

## HU-020: Borrar viviendas

## 20000 usuarios en 10 segundos

En esta historia de usuario, al ser más compleja que la anterior, en 20000 ya podemos observar que la CPU y la memoria se ponen al máximo al empezar y se mantienen durante toda la ejecución.

```
[2] computerdatabase.advanced.AdvancedSimulationStep02
[3] computerdatabase.advanced.AdvancedSimulationStep03
[4] computerdatabase.advanced.AdvancedSimulationStep04
[5] computerdatabase.advanced.AdvancedSimulationStep05
[6] computerdatabase.advanced.AdvancedSimulationStep06
[7] dp2.BorrarViviendaPerformanceTest
[8] dp2.ViviendaFiltroPerformanceTest
/
Select run description (optional)
Simulation dp2.BorrarViviendaPerformanceTest started...
=====
2020-05-31 13:25:05      5s elapsed
--- Requests ---
> Global          (OK=496   KO=670
> Home           (OK=496   KO=670
--- Errors ---
> i.m.c.AbstractChannel$AnnotatedSocketException: No buffer space available (maximum connections reached): connect: www.dp2.c...
> i.m.c.AbstractChannel$AnnotatedConnectException: Connection re used: no further information: www.dp2.com/127.0.0.1:80
--- BorrarViviendaPerformanceTest ---
{
    waiting: 0 / active: 20000 / done: 0
}
=====
```



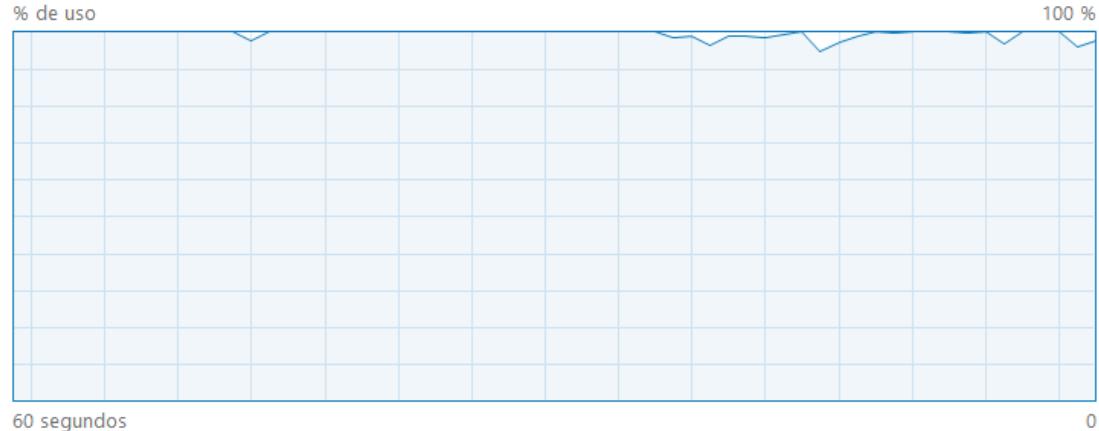
## Memoria

8,0 GB

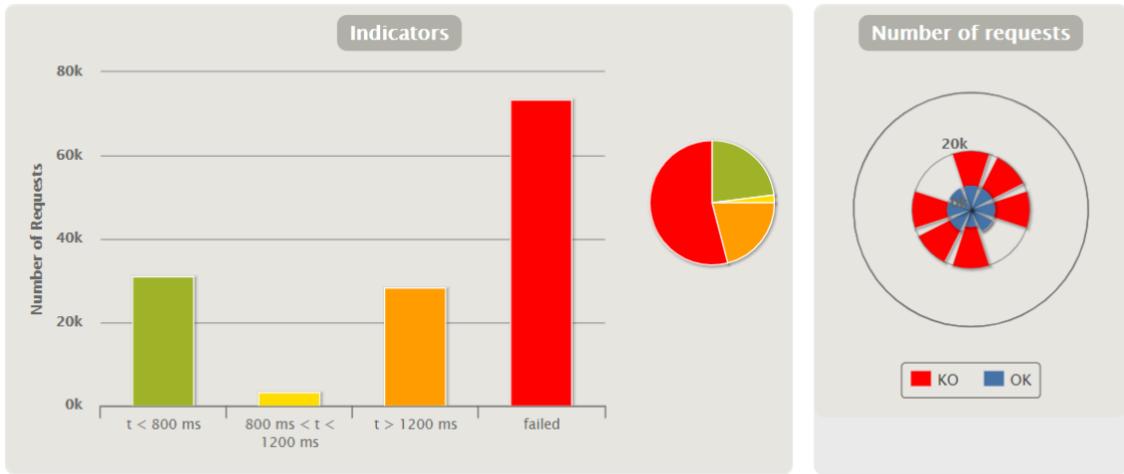


## CPU

Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz



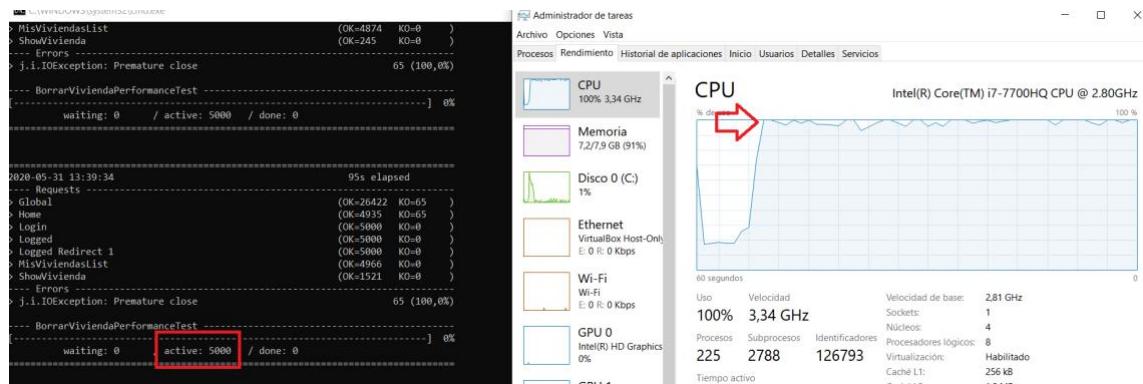
Podemos observar entonces, que, aunque los tiempos de respuesta medios no son tan altos, la mayoría de las peticiones fallan por lo que no es una buena prueba.



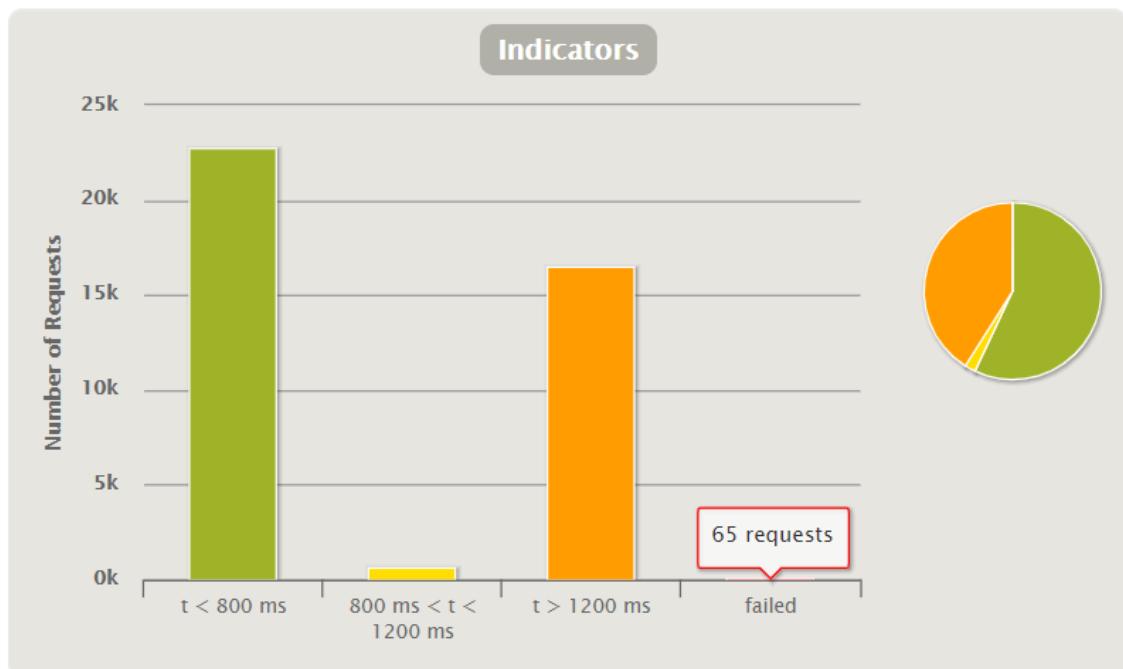
▶ STATISTICS		Expand all groups   Collapse all groups													
Requests ▲		🕒 Executions					🕒 Response Time (ms)								
		Total ▲	OK ▲	KO ▲	% KO ▲	Cnt/s ▲	Min ▲	50th pct ▲	75th pct ▲	95th pct ▲	99th pct ▲	Max ▲	Mean ▲	Std Dev ▲	
Global information		136306	62825	73481	54%	725.032	0	2240	3601	44091	60006	61298	6798	13361	
Home		20000	8192	11808	59%	106.383	824	4360	5717	6910	7305	61298	4818	5519	
Login		20000	8192	11808	59%	106.383	1	2051	2159	2441	2616	2751	1364	992	
Logged		20000	8192	11808	59%	106.383	1	2053	2175	2359	2518	3031	1376	973	
Logged Redirect 1		8192	8192	0	0%	43.574	0	123	274	508	568	806	180	155	
MisViviendasList		20000	5828	14172	71%	106.383	602	2334	31624	60005	60021	60057	17037	21862	
ShowVivienda		20000	8001	11999	60%	106.383	556	2388	29946	55210	60005	60033	16972	19205	
ViviendaDeleted		20000	8114	11886	59%	106.383	1	2304	3059	17450	20996	24085	4443	5204	
Vivienda...direct 1		8114	8114	0	0%	43.16	0	697	1022	1289	1362	1426	598	491	

## 5000 usuarios en 10 segundos

La CPU sufre altibajos al empezar mientras que la memoria se mantiene estable. Es cuando todos los usuarios están activos cuando la CPU se pone al 100%.



Sin embargo, podemos observar que la mayoría de los usuarios consiguen entrar y que los tiempos de respuesta medios son buenos. Solo 65 peticiones han fallado.



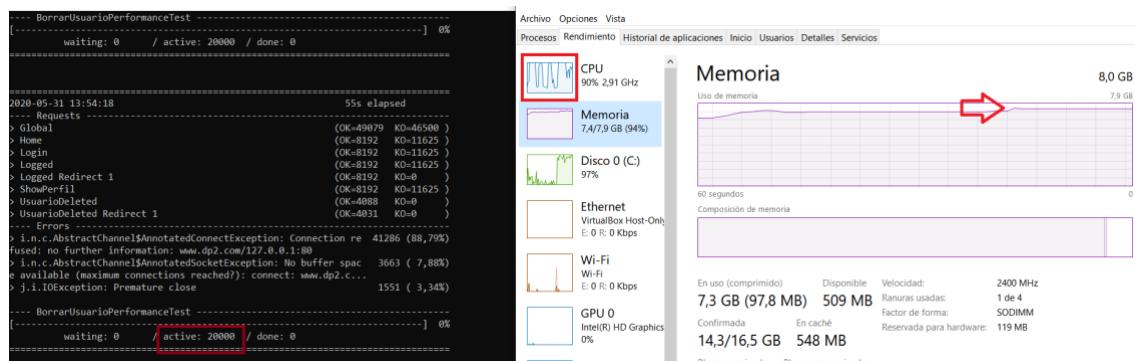
**STATISTICS** Expand all groups | Collapse all groups

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	40000	39935	65	0%	268.456	0	179	3283	24673	37368	58087	4334	8435		
Home	5000	4935	65	1%	33.557	1092	3045	3287	3567	3733	3859	2812	627		
Login	5000	5000	0	0%	33.557	1	39	139	206	235	280	77	69		
Logged	5000	5000	0	0%	33.557	1	40	100	213	240	275	69	68		
Logged Redirect 1	5000	5000	0	0%	33.557	0	43	75	202	222	242	59	57		
MisViviendasList	5000	5000	0	0%	33.557	7	19392	30090	38687	44291	58087	20004	12073		
ShowVivienda	5000	5000	0	0%	33.557	4	10192	16680	21650	26426	38064	10412	7351		
ViviendaDeleted	5000	5000	0	0%	33.557	1	17	1521	3516	4187	4282	830	1221		
Vivienda...direct 1	5000	5000	0	0%	33.557	0	9	780	1639	2140	2175	411	632		

## HU-023: Borrar usuarios

20000 usuarios en 10 segundos

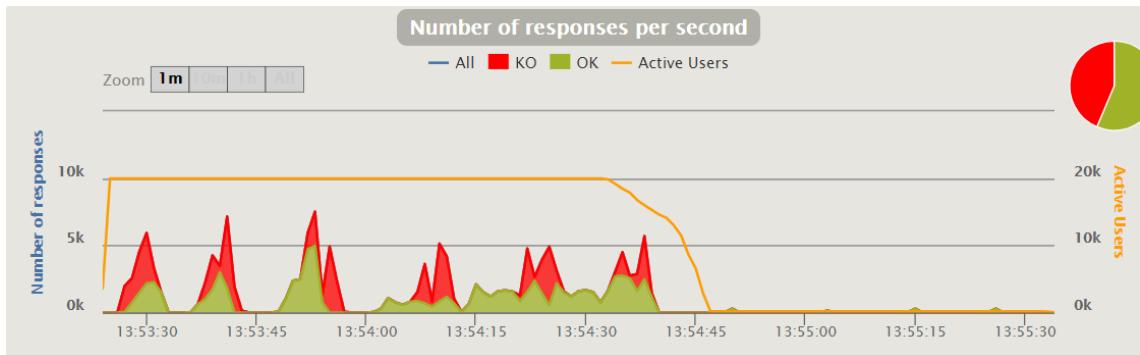
En esta historia de usuario observamos como la memoria aumenta y la CPU tiene altibajos cuando están todos los usuarios activos.



En las estadísticas de Gatling observamos lo siguiente:



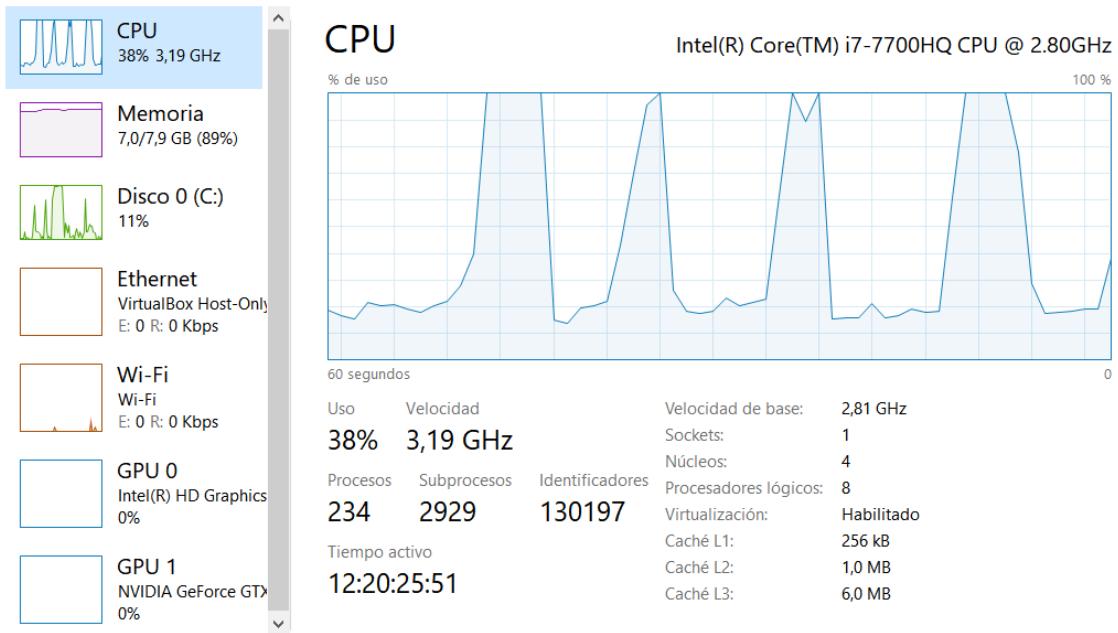
La mitad de las peticiones son buenas pero la otra mitad fallan.



De hecho, cuando los usuarios están activos al mismo tiempo, hay más respuestas fallidas por segundo que acertadas, por lo que no podemos considerar esta prueba como buena.

#### 5000 usuarios en 10 segundos

La CPU sufre altibajos al iniciar la prueba.



En las estadísticas de Gatling observamos que todas las peticiones son acertadas y los tiempos de ejecución son de 4 segundos máximo lo que es una muy buena respuesta.



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	45000	45000	0	0%	625	0	13	55	3488	3750	4591	438	1045				
Home	5000	5000	0	0%	69.444	546	3448	3623	3907	4356	4591	3259	577				
Login	5000	5000	0	0%	69.444	1	30	71	124	146	151	46	41				
Logged	5000	5000	0	0%	69.444	1	18	34	61	76	170	24	18				
Logged Redirect 1	5000	5000	0	0%	69.444	0	19	40	80	101	111	26	24				
ShowPerfil	5000	5000	0	0%	69.444	1	419	1006	1578	1678	1712	554	532				
UsuarioDeleted	5000	5000	0	0%	69.444	1	7	11	17	23	31	8	5				
UsuarioD...direct 1	5000	5000	0	0%	69.444	0	6	11	22	29	48	8	6				
LoogedOut	5000	5000	0	0%	69.444	1	6	10	24	45	57	8	8				
LoogedOu...direct 1	5000	5000	0	0%	69.444	0	5	9	23	36	55	7	7				

## CONCLUSIÓN

Puedo concluir que, para las características de mi ordenador, 75000 respuestas es demasiado y que la aplicación no funcionaría para ninguna historia de usuario.

Con 20000 usuarios, aunque en algunos casos hemos obtenido hasta la mitad de las peticiones correctas, no es buen rendimiento que la otra mitad siga fallando.

La mejor situación para probar la aplicación entera es, en mi caso con 5000 usuarios ya que, aunque en algunos casos de uso hemos obtenido respuestas fallidas, han sido la minoría y en otras ocasiones se ha obtenido hasta un total de peticiones correctas.

Santiago Martín Gómez

HU	Descripción
HU-002	Rechazar/Aceptar compra
HU-017	Crear un mensaje

## CARACTERÍSTICAS DEL SISTEMA

He utilizado Acer Aspire A515-51G, cuyas características son las siguientes.

<b>Processor(Model)/Frecuency/Nucleos/HT)</b>	Intel Core i7-7500U / 2.90GHz /2
<b>Cache L1</b>	128 kB
<b>Cache L2</b>	512 kB
<b>Cache L3</b>	4 MB
<b>Memory principal</b>	DDR4 8GB

## RESULTADOS

Todas las pruebas son en 10segundos

**HU-002: Rechazar compra**

## > Global Information



**STATISTICS** Expand all groups | Collapse all groups

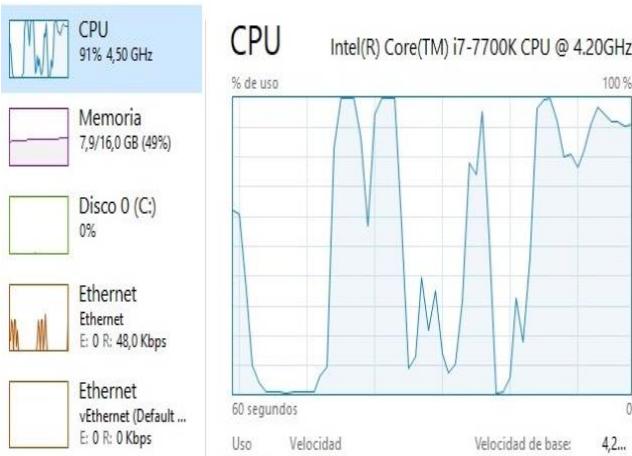
Requests	Executions					Response Time (ms)								
	Total	OK	KO	% KO	Cnt/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev	
Global Information	7	7	0	0%	0.179	3	27	137	292	313	318	90	120	
request_0	1	1	0	0%	0.026	27	27	27	27	27	27	27	0	
request_1	1	1	0	0%	0.026	9	9	9	9	9	9	9	0	
request_2	1	1	0	0%	0.026	3	3	3	3	3	3	3	0	
request_3	1	1	0	0%	0.026	41	41	41	41	41	41	41	0	
request_...direct 1	1	1	0	0%	0.026	3	3	3	3	3	3	3	0	
request_4	1	1	0	0%	0.026	318	318	318	318	318	318	318	0	
request_5	1	1	0	0%	0.026	232	232	232	232	232	232	232	0	

Esta gráfica es el resultado tras ejecutar la prueba con 1 solo usuario. La memoria del PC no se ve afectada prácticamente cuando se ejecuta.

> Global Information



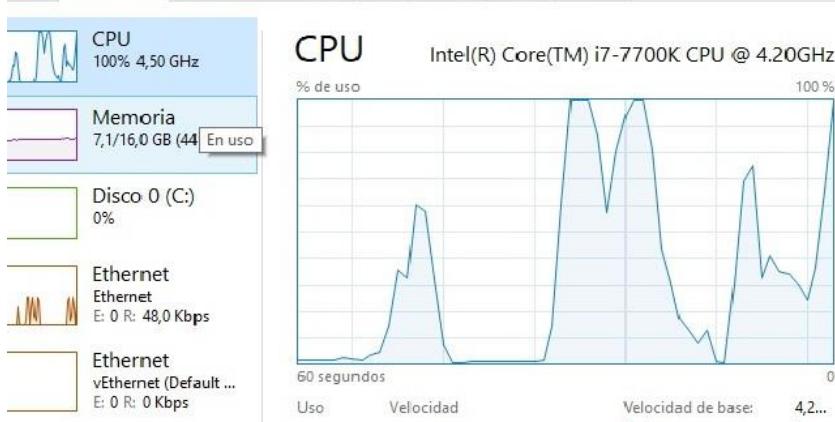
Tras hacer la prueba con 5000 usuarios cada 10 segundos podemos ver el uso de la CPU llega a alcanzar el 91%. En las estadísticas podemos apreciar como algunas de las tareas ya empiezan a demorarse un poco más a la hora de realizarse.



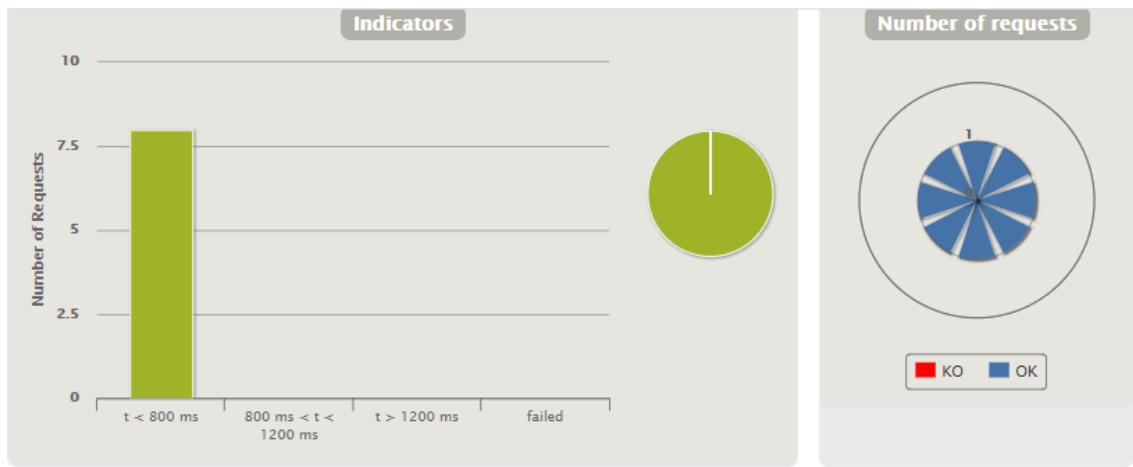
#### > Global Information



Esta prueba está realizada con 20000 usuarios. Ya se aprecia como el sistema empieza a fallar de forma considerable y que muchas peticiones empiezan a tardar un tiempo considerable. El uso de la CPU ya es pleno y se alcanza el 100%.

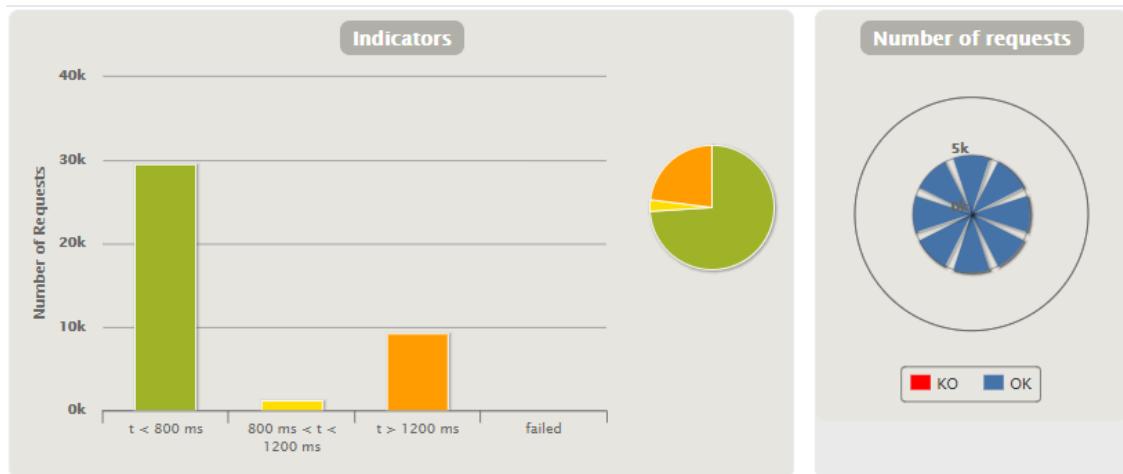


## HU-002: Crear Mensaje



▶ STATISTICS		Expand all groups   Collapse all groups													
Requests ^		🕒 Executions					🕒 Response Time (ms)								Std Dev ♦
		Total ♦	OK ♦	KO ♦	% KO ♦	Cnt/s ♦	Min ♦	50th pct ♦	75th pct ♦	95th pct ♦	99th pct ♦	Max ♦	Mean ♦		
Global Information		8	8	0	0%	0.105	2	3	18	128	160	168	30	55	
Home		1	1	0	0%	0.013	55	55	55	55	55	55	55	0	
Login		1	1	0	0%	0.013	2	2	2	2	2	2	2	0	
request_2		1	1	0	0%	0.013	2	2	2	2	2	2	2	0	
Logged		1	1	0	0%	0.013	5	5	5	5	5	5	5	0	
Logged Redirect 1		1	1	0	0%	0.013	2	2	2	2	2	2	2	0	
CreateMensajeForm		1	1	0	0%	0.013	168	168	168	168	168	168	168	0	
MensajesEnviados		1	1	0	0%	0.013	2	2	2	2	2	2	2	0	
Mensajes...direct 1		1	1	0	0%	0.013	3	3	3	3	3	3	3	0	

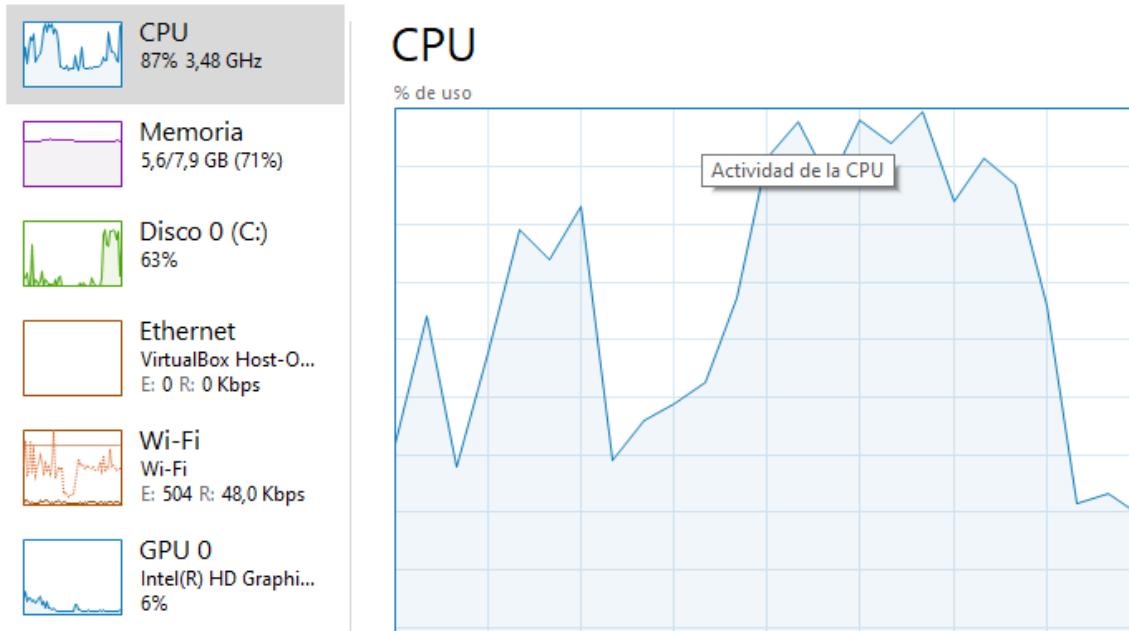
Al igual que con compra, no hay problema alguno a la hora de crear y enviar un mensaje con un solo usuario. Todas las peticiones se realizan en menos de 800 ms,

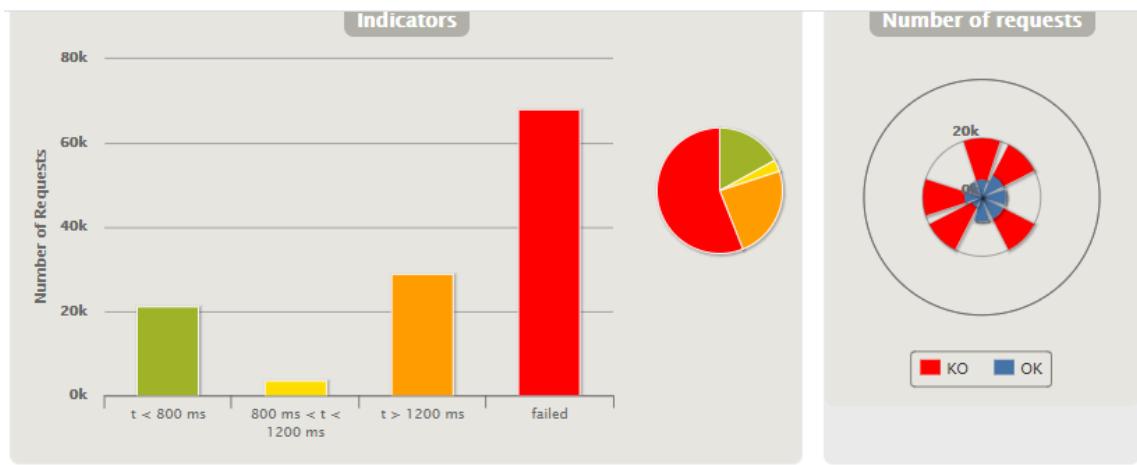


▶ STATISTICS Expand all groups | Collapse all groups

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	40000	40000	0	0%	289.855	0	55	904	28677	42554	55641	4076	9729		
Home	5000	5000	0	0%	36.232	3	330	763	1808	2445	3008	538	556		
Login	5000	5000	0	0%	36.232	1	63	202	637	867	1154	162	209		
request_2	5000	5000	0	0%	36.232	0	32	100	311	515	1000	79	113		
Logged	5000	5000	0	0%	36.232	1	19	39	100	225	304	31	37		
Logged Redirect 1	5000	5000	0	0%	36.232	0	12	27	70	136	247	20	25		
CreateMensajeForm	5000	5000	0	0%	36.232	9	23997	34509	44486	52364	55641	24807	12834		
MensajesEnviados	5000	5000	0	0%	36.232	0	2179	11402	24918	28471	29344	6474	8164		
Mensajes...direct 1	5000	5000	0	0%	36.232	0	4	1144	1981	2234	2438	498	726		

Esta es la prueba con 5000 usuarios. Aunque no falle ninguna solicitud, ya se aprecia como algunas de ellas empiezan a tardar más de 800ms, y casi 10000 llegan a más de 1200 ms. La CPU llega al 87%.





▶ 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		122157	53979	68178	56%	690.153	0	2532	5856	57320	60086	60541	7835	14437
Home		20000	5935	14065	70%	112.994	2014	3503	9680	14048	17287	21718	5949	4278
Login		20000	8192	11808	59%	112.994	1	5654	7365	9256	10285	60240	4911	5252
request_2		8192	8192	0	0%	46.282	0	312	1949	3310	4463	5472	1063	1218
Logged		20000	8192	11808	59%	112.994	1	2356	3131	3854	4365	5433	2132	1257
Logged Redirect 1		8192	8192	0	0%	46.282	0	477	712	2132	2486	3390	552	569
CreateMensajeForm		20000	3730	16270	81%	112.994	27	2663	53858	60122	60446	60541	21009	25010
MensajesEnviados		20000	5773	14227	71%	112.994	2022	2448	19233	60000	60002	60122	12869	17386
Mensajes...direct 1		5773	5773	0	0%	32.616	0	1077	1523	2453	2731	3092	1129	691

Esta prueba es con 200000 usuarios. Se puede ver como fallan la gran mayoría de las peticiones y que hay más peticiones que tardan más de 1200ms a las que tardan menos de 800ms. Y la CPU llegó al 100%. La captura es realizada al principio de la ejecución.



## CONCLUSIÓN

Como conclusión podemos sacar que el sistema a partir de los 5000 usuarios ambas pruebas empiezan ya a ser un poco renqueantes porque las solicitudes que se hacen empiezan a demorarse. Cuando llegamos a los 20000 usuarios ya el sistema falla en más peticiones de las que acierta, por tanto, se podría considerar que ahí está uno de los límites de los fallos. Claramente, cuando hacemos la prueba con un usuario no hay problema alguno.

## Alejandro Molina Morales

HU	Descripción
HU-018	Realizar oferta sobre una vivienda
HU-019	Acceder al Dashboard de administrador

## CARACTERÍSTICAS DEL SISTEMA

El equipo con el que se han realizado las pruebas de rendimiento para estas historias de usuario tiene las siguientes características:

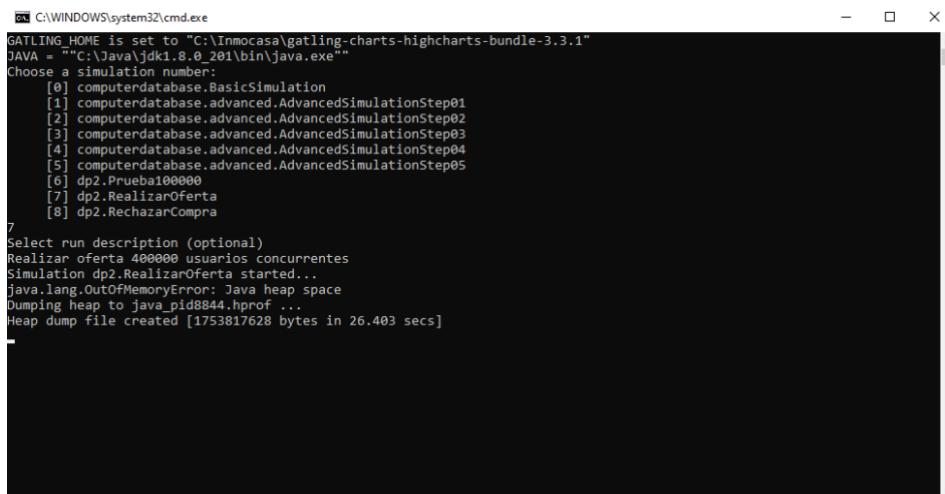
<b>Processor(Model)/Frecuency/Nucleos/HT</b>	Intel Core i7-7700K / 4.20 GHz
<b>Cache L1</b>	256 kB
<b>Cache L2</b>	1,0 MB
<b>Cache L3</b>	8,0 MB
<b>Memory principal</b>	16GB

## Procedimiento:

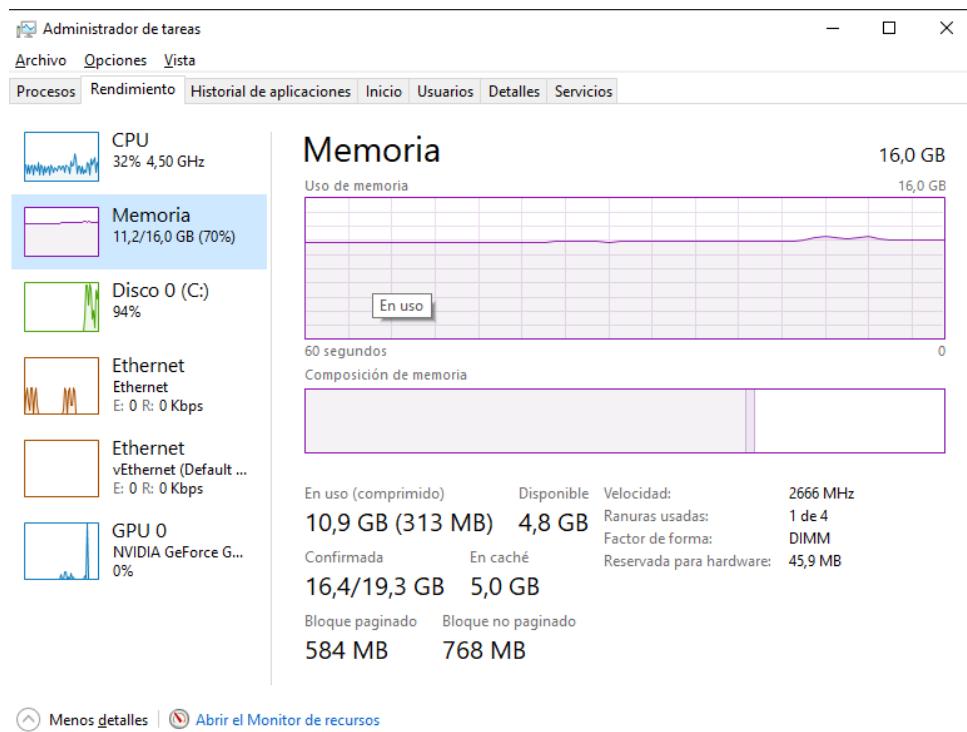
Las pruebas de rendimiento sobre ambas historias de usuario se han realizado de la misma manera: primeramente, se ha buscado el número mínimo de usuario concurrentes en los que cada HU no puede soportar. A continuación, y con el objetivo de buscar el número máximo de usuarios concurrentes que las HU pueden soportar y ofrecer un resultado óptimo. En ambos escenarios, se han inyectado los usuarios a lo largo de 10 segundos.

### HU-18: Realizar una oferta sobre una vivienda

Para obtener el valor de **usuarios mínimos que el escenario no puede soportar**, obtenemos que con 400.000 usuarios concurrentes el sistema no responde. La ejecución de Gatling nos devuelve el siguiente mensaje:



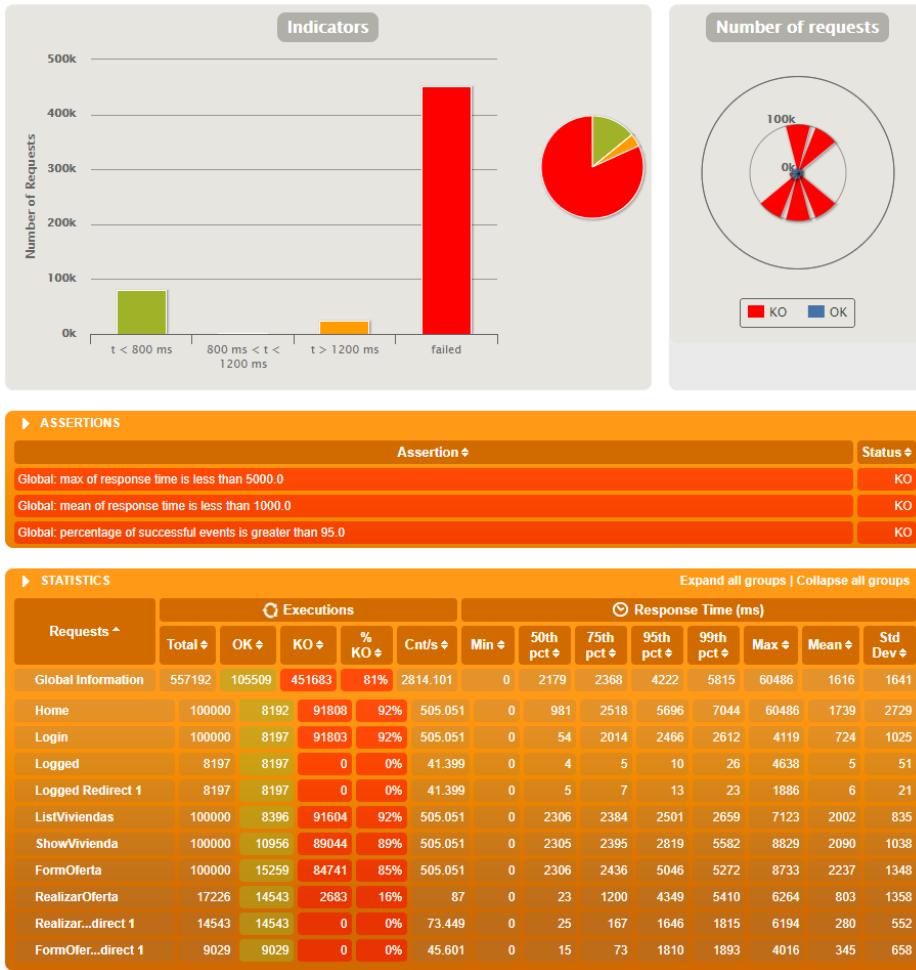
```
C:\WINDOWS\system32\cmd.exe
GATLING_HOME is set to "C:\Inmocasa\gatling-charts-highcharts-bundle-3.3.1"
JAVA = "\"C:\Java\jdk1.8.0_201\bin\java.exe\""
Choose a simulation number:
[0] computerdatabase.BasicSimulation
[1] computerdatabase.advanced.AdvancedSimulationStep01
[2] computerdatabase.advanced.AdvancedSimulationStep02
[3] computerdatabase.advanced.AdvancedSimulationStep03
[4] computerdatabase.advanced.AdvancedSimulationStep04
[5] computerdatabase.advanced.AdvancedSimulationStep05
[6] dp2.Prueba100000
[7] dp2.RealizarOferta
[8] dp2.RechazarCompra
7
Select run description (optional)
Realizar oferta 400000 usuarios concurrentes
Simulation dp2.RealizarOferta started...
java.lang.OutOfMemoryError: Java heap space
Dumping heap to java_pid8884.hprof ...
Heap dump file created [1753817628 bytes in 26.403 secs]
```



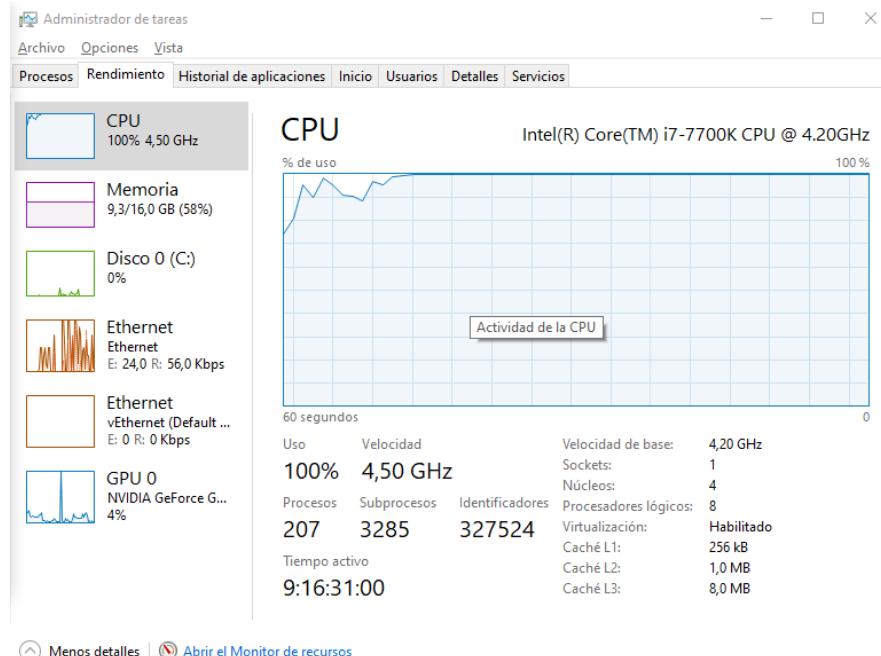
Además, en el rendimiento del equipo, podemos ver cómo, aunque la CPU no tenga un gran porcentaje de uso (llega a un 50%), se está usando un 70% de la memoria principal.

Tras obtener este valor, buscamos obtener **el número máximo de usuarios concurrentes que el escenario soporta correctamente**. Para ello, se ha disminuido el valor de usuarios concurrentes hasta dar con el valor en el que todas las peticiones son resueltas correctamente (peticiones OK). Además, para asegurarnos que dichos valores son óptimos, se han añadido 3 comprobaciones: que el tiempo de respuesta máximo sea de 5 segundos, que la media en el tiempo de respuesta en todas las peticiones sea menos de 1 segundo y que el número de respuestas correctas por la aplicación es mayor de un 95%.

Con 100.000 usuarios concurrentes, podemos ver como la gran mayoría de las respuestas son fallidas (KO). Además, en la tabla de estadísticas ('Statistics') podemos ver como ha habido alguna respuesta que ha tardado hasta 60s (un valor muy malo para la aplicación).



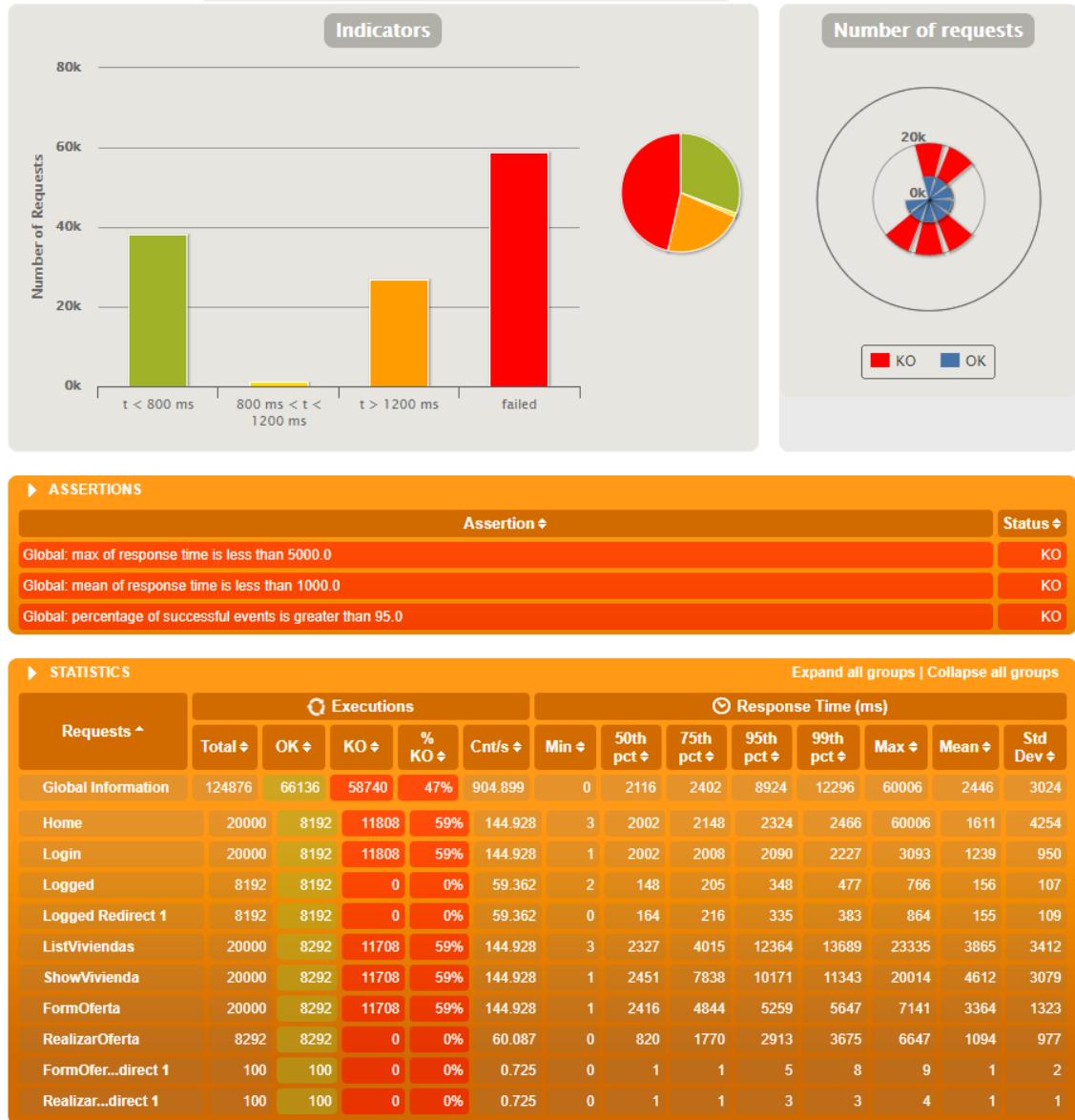
Además, el rendimiento del equipo durante la prueba ha sido el siguiente:



Como se puede observar en la imagen, durante la prueba la CPU ha estado al 100% de uso la mayoría del tiempo mientras que la memoria principal en un 58%. Por tanto, podemos deducir

que, aunque la CPU llega al 100%, lo que produce que la aplicación no funcione en absoluto es la diferencia en el porcentaje de uso de la memoria.

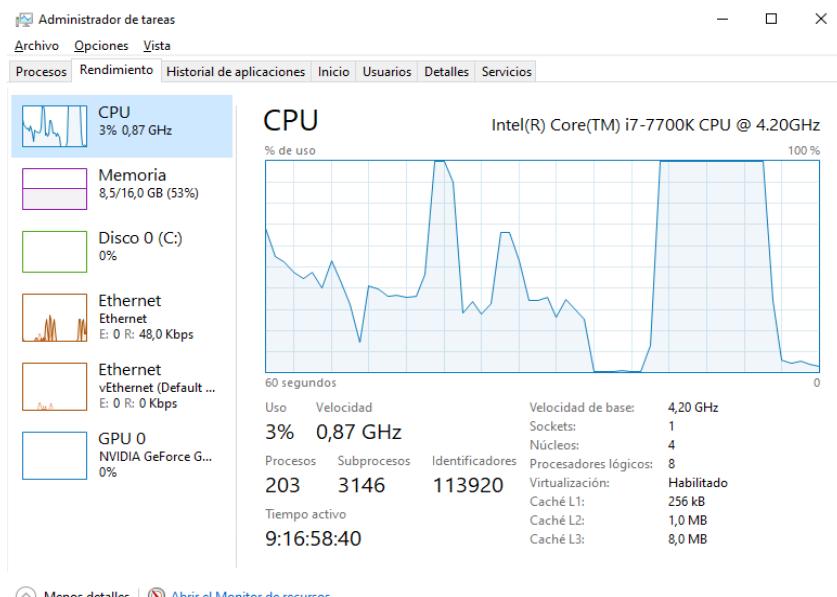
La siguiente prueba se ha realizado sobre 20.000 usuarios concurrentes. Podemos ver como el porcentaje de respuestas fallidas ha disminuido bastante, aunque sigue siendo un valor muy alto. Además, las comprobaciones han sido también fallidas.



Por último, se ha hecho la prueba con 5.000 usuarios concurrentes, y los resultados han sido los siguientes:



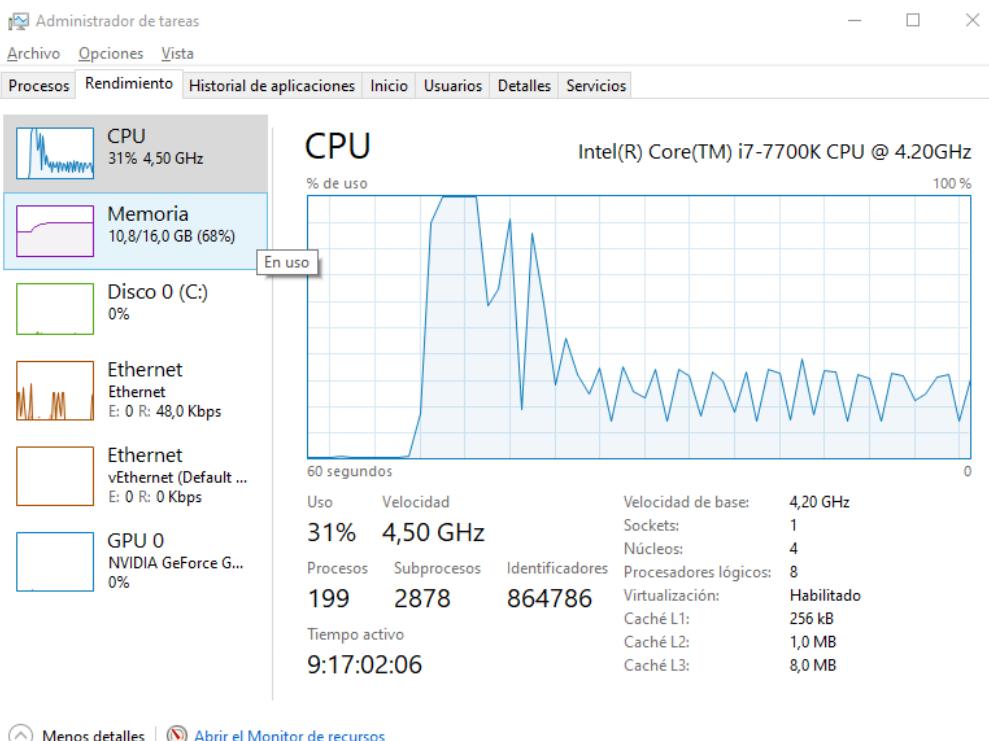
Como vemos, los resultados son óptimos y todas las respuestas son correctas. Aún así, hay algunas peticiones que han tardado más de 0,8s, e incluso más de 1,2s. Además, podemos ver como alguna de las respuestas ha tardado más de 5 segundos en responder (más concretamente 6s). El rendimiento del equipo durante esta prueba ha sido el siguiente:



Como vemos, la CPU alcanza en algún tramo el 100% de uso, pero la memoria principal no supera el 53%.

## HU-019: Acceder al Dashboard de administrador

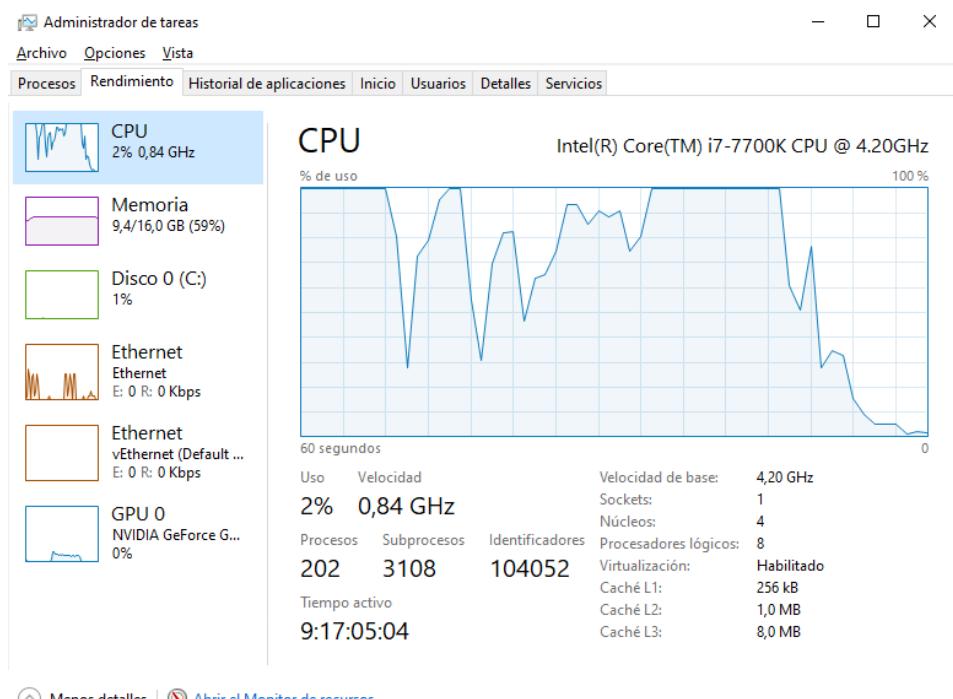
Esta historia de usuario corresponde al acceso por parte del administrador al Dashboard de la aplicación, donde se controla el uso de esta (número de viviendas compradas y número de ofertas realizadas). Se ha seguido la misma metodología que en la historia anterior y el **número mínimo de usuarios que el escenario no puede soportar** coincide con la prueba anterior: 400.000 usuarios.



Al igual que en el escenario anterior, la CPU no tiene un uso muy grande, sin embargo, la memoria principal sí, llegando hasta el 68%.

Para obtener el **número máximo de usuarios concurrentes que el escenario soporta correctamente**, hemos usado los mismos valores que en el caso anterior.

Con 100.000 usuarios concurrentes, la aplicación responde de la siguiente forma:



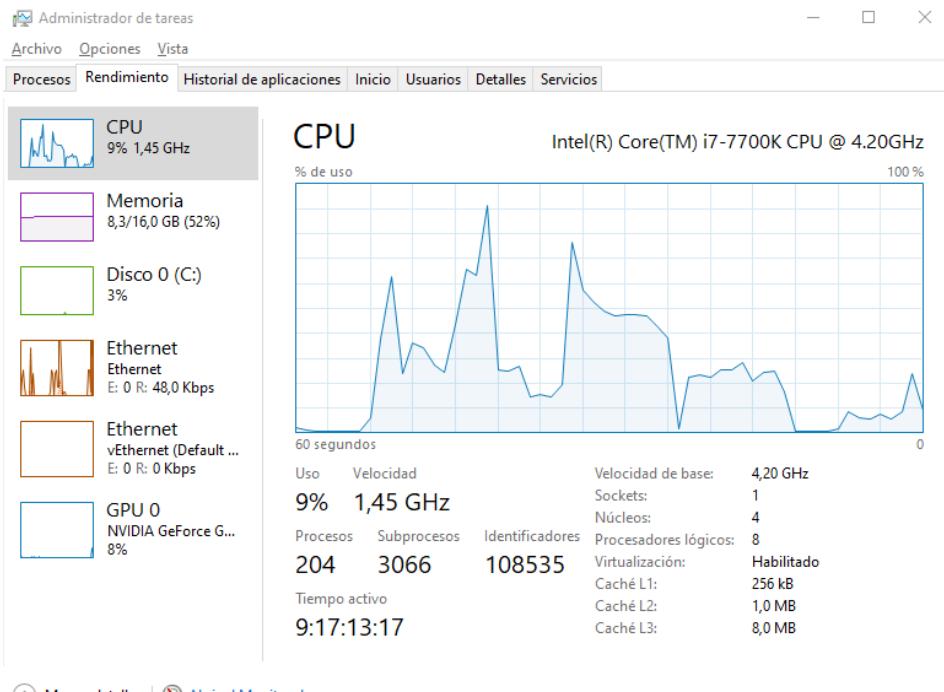
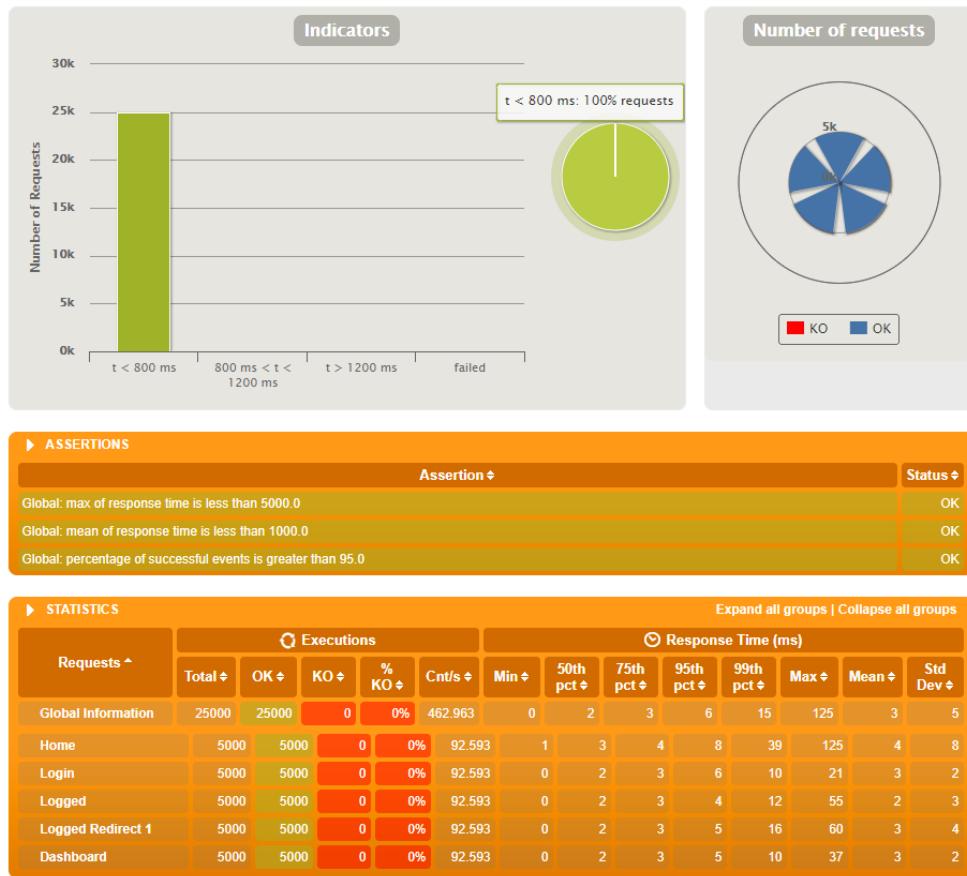
Vemos como la gran mayoría de respuestas por parte del servidor son fallidas y que consume mucha CPU. La memoria principal en este caso no llega al 60%.

Para 20.000 usuarios:



Observamos que el número de peticiones correctas supera al de peticiones fallidas, aunque sigue sin tener valores óptimos. Con respecto al rendimiento del equipo, la CPU tiene varios picos de uso del 100% pero mucho menor en su conjunto que las pruebas anteriores, y la memoria no alcanza el 55%.

Con 5000 usuarios concurrentes:



En este caso, vemos como todas las peticiones han sido favorables, no superando ninguna de ellas los 0,8 segundos de respuesta. Además, vemos como las comprobaciones que habíamos realizado se han pasado satisfactoriamente. El rendimiento del equipo ha sido mucho mejor que en las anteriores pruebas, teniendo un uso de CPU y memoria principal más moderado.

## Conclusiones

A partir de las pruebas realizadas, podemos concluir que, con las características de mi equipo, el sistema podría llegar a funcionar hasta los 400.000 usuarios concurrentes, pero no de una forma correcta. Los resultados que se dan con más de 20.000 usuarios no son para nada satisfactorios, ya que muchos de ellos son fallidos y tienen un tiempo muy grande de respuesta. A partir de 20.000 usuarios vemos que, dependiendo de la historia de usuario, obtenemos unos mejores resultados, pero que siguen sin ser satisfactorios. Por tanto, el número máximo de usuarios concurrentes que se pueden dar en estas historias de usuario se recomienda que sean de 5.000, ya que las respuestas pasan a ser correctas mayoritariamente y con unos datos de respuesta bastante óptimos.

Para mejorar el rendimiento de la aplicación en este equipo a partir de los datos que hemos obtenido de las pruebas, pienso que habría que mejorar la memoria principal por encima de la CPU, ya que las pruebas en las que la aplicación ni siquiera ha respondido, han sido aquellas en las que la memoria principal ha sobrepasado el 65%. Si es cierto que la CPU ha estado en una gran parte de la ejecución de las pruebas en un 100% de uso, lo que no es aconsejable ya que da resultados no deseados. Por ello, si fuera posible, se recomienda también su mejora.

Miguel Emmanuel Saavedra Gómez

HU	Descripción
HU-009	Visualizar listado de publicaciones recientes
HU-013	Añadir una valoración
HU-022	Descargar información de usuario
HU-005	Listar todas las valoraciones realizadas a mis viviendas

## CARACTERÍSTICAS DEL SISTEMA

El equipo con el que se han realizado las pruebas de rendimiento para estas historias de usuario tiene las siguientes características:

<b>Processor(Model)/Frecuency/Nucleos/HT</b>	Intel Core i7-8550U / 2.00 GHz
<b>Cache L1</b>	256 kB
<b>Cache L2</b>	1,0 MB
<b>Cache L3</b>	8,0 MB
<b>Memory principal</b>	8GB

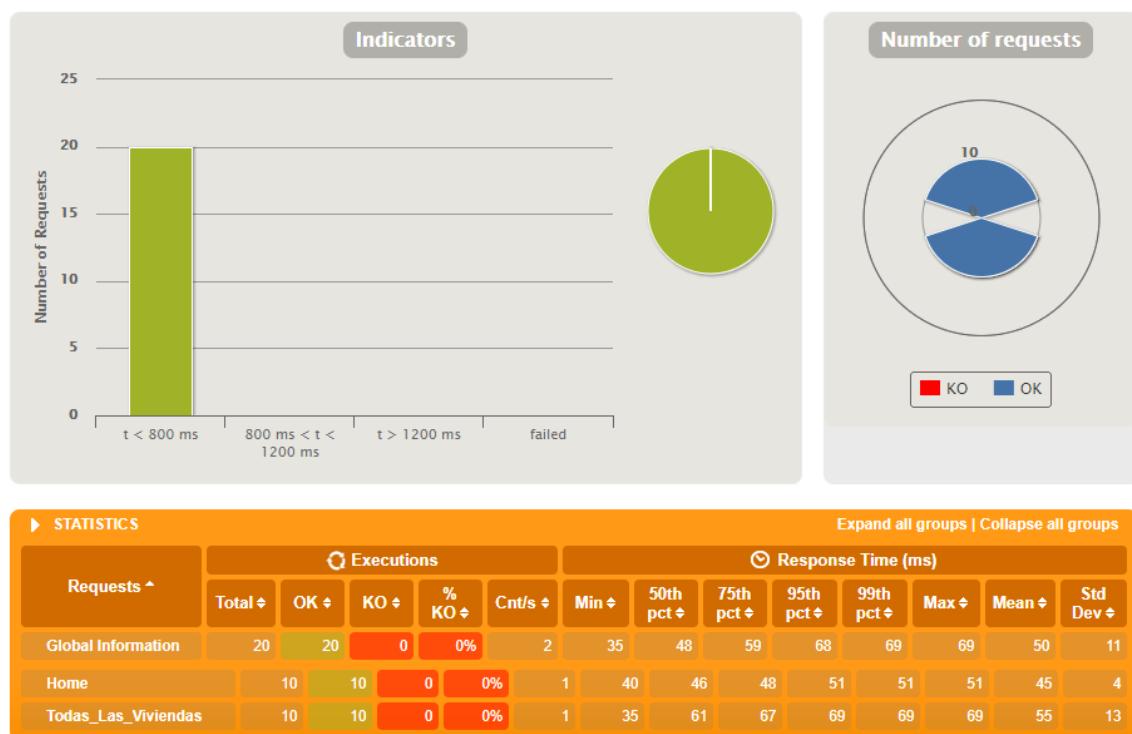
## Procedimiento:

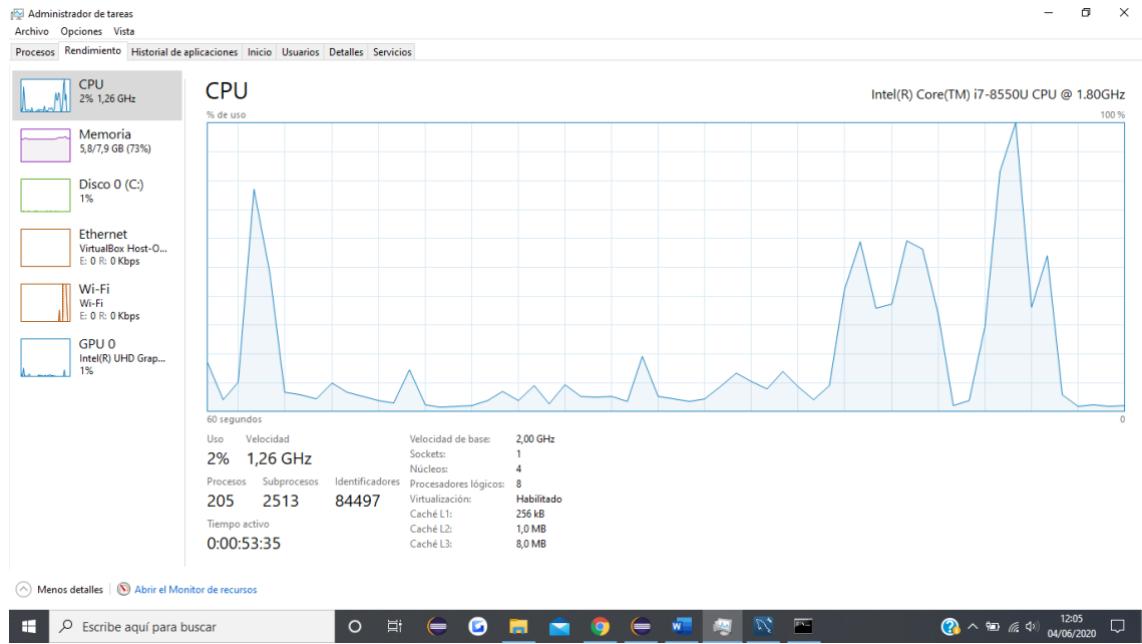
Por cada historia de usuario se han ido probando un número diferente de usuarios, analizando el rendimiento que ofrece el sistema y buscando el número máximo de usuarios para que la aplicación funcione. Una vez encontrados, hemos buscado el número mínimo de usuarios para que la aplicación no funcione.

## RESULTADOS

### HU-009

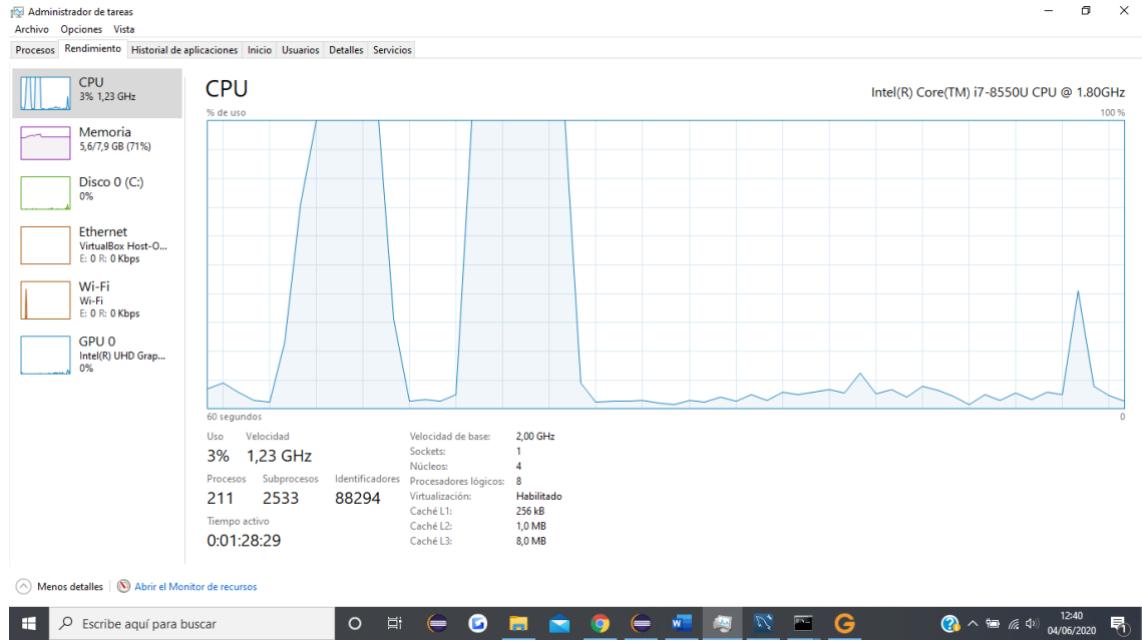
En primer lugar, he comenzado probando 10 usuarios en 10 segundos, para comprobar el correcto funcionamiento de la prueba con una cantidad mínima de usuarios. El rendimiento obtenido es el siguiente:



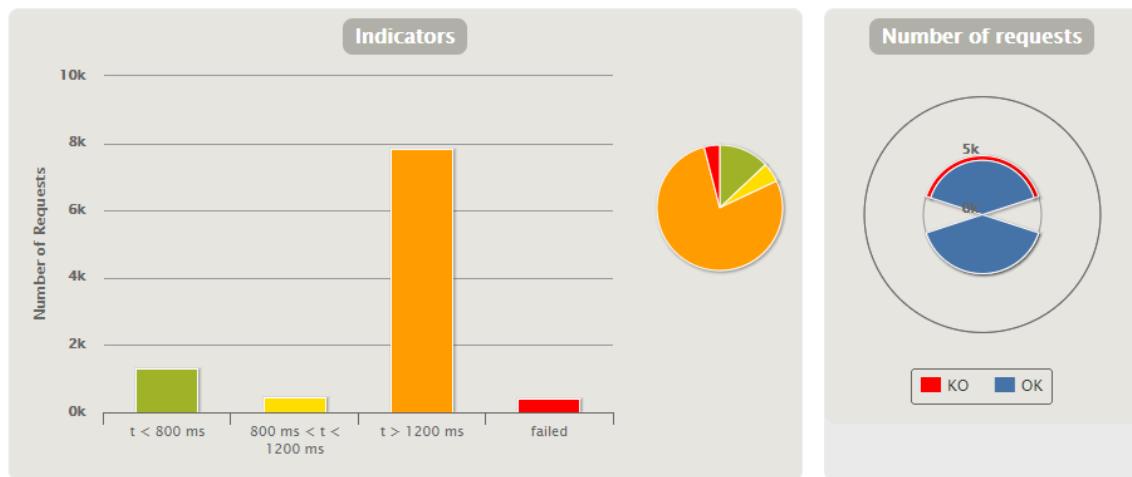


Mediante el informe que genera Gatling podemos observar que efectivamente, se realizan satisfactoriamente las peticiones.

### 5000 usuarios en 10 segundos



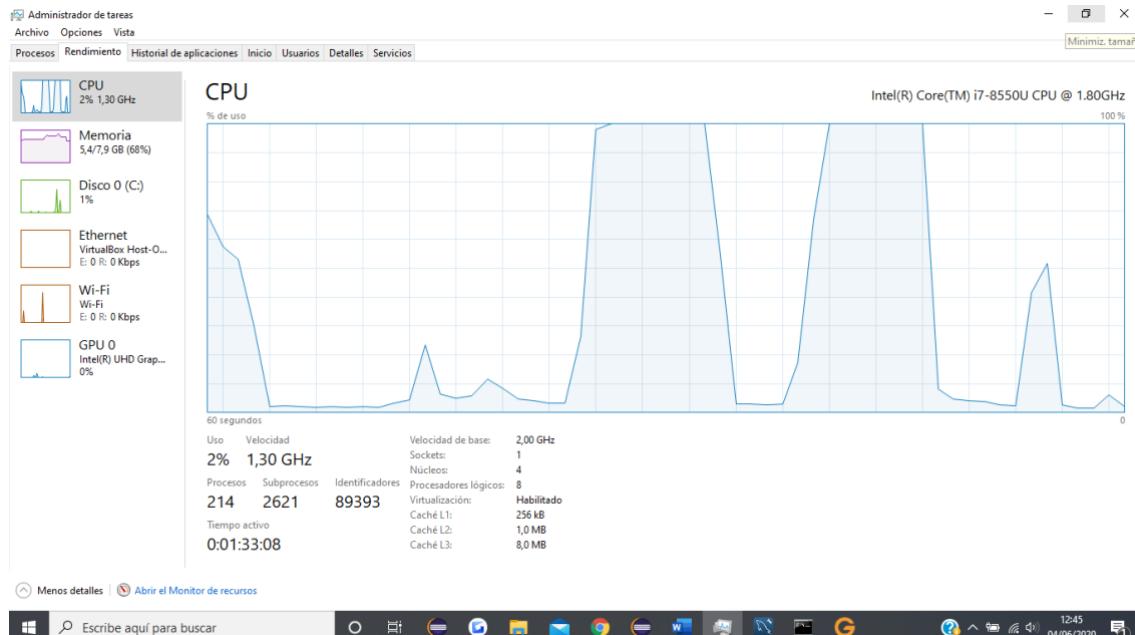
Produce un incremento de un 5% en memoria, sin embargo la CPU alterna consumos máximos y mínimos.



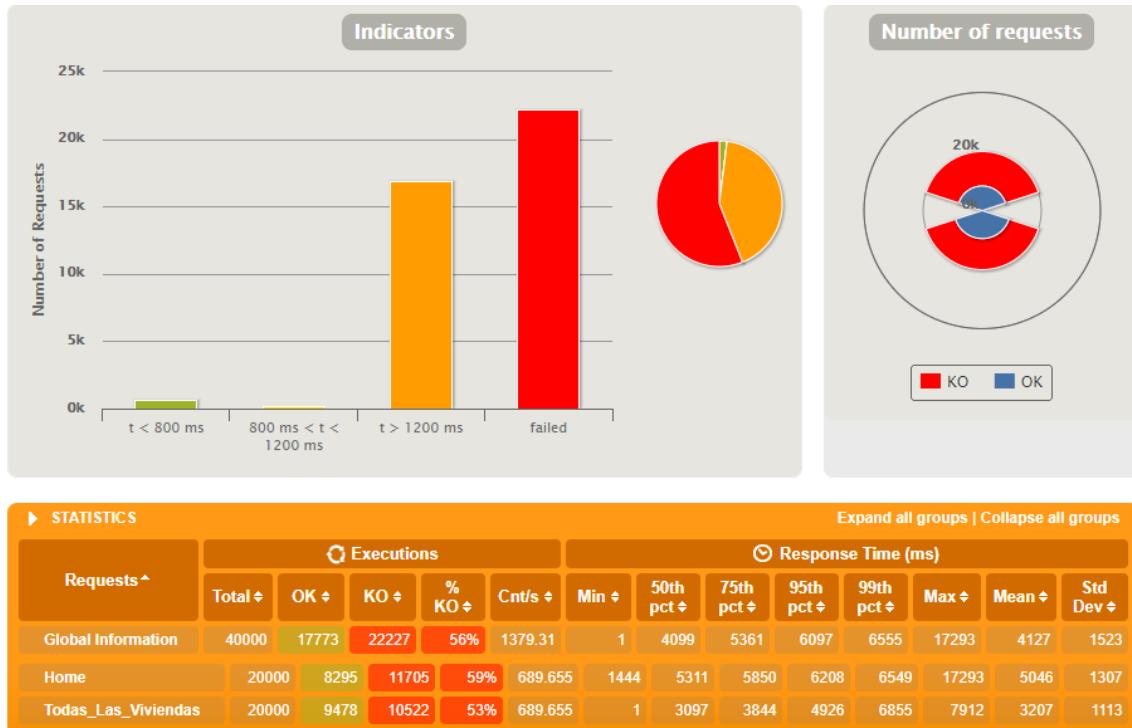
▶ STATISTICS		Expand all groups   Collapse all groups												
Requests ▲		⌚ Executions				⌚ Response Time (ms)								
		Total ▾	OK ▾	KO ▾	% KO ▾	Cnt/s ▾	Min ▾	50th pct ▾	75th pct ▾	95th pct ▾	99th pct ▾	Max ▾	Mean ▾	Std Dev ▾
Global Information		10000	9605	395	4%	588.235	3	2170	3595	4501	4626	4665	2368	1315
Home		5000	4605	395	8%	294.118	1222	3595	4224	4579	4642	4665	3402	898
Todas_Las_Viviendas		5000	5000	0	0%	294.118	3	1472	1942	2253	2376	2651	1334	717

Se empiezan a producir pequeños fallos

## 20.000 usuarios en 10 segundos

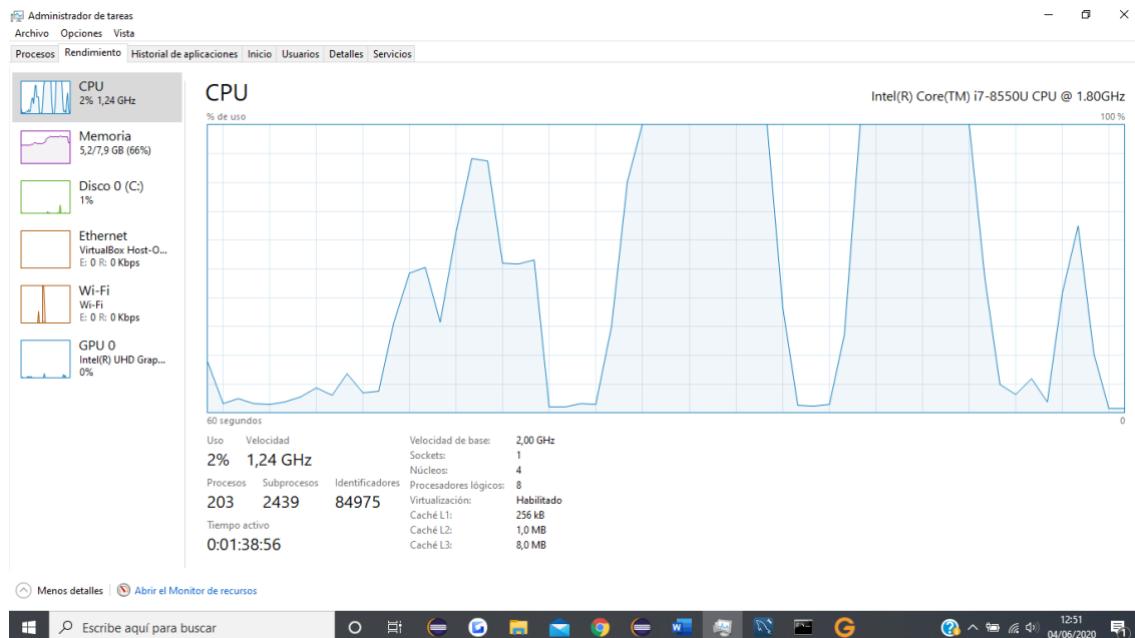


Se puede apreciar que la CPU vuelve a intercalar entre 100% y el consumo estándar cuando se ejecuta esta prueba. La memoria en este caso aumenta un 10% a diferencia de la otra vez.

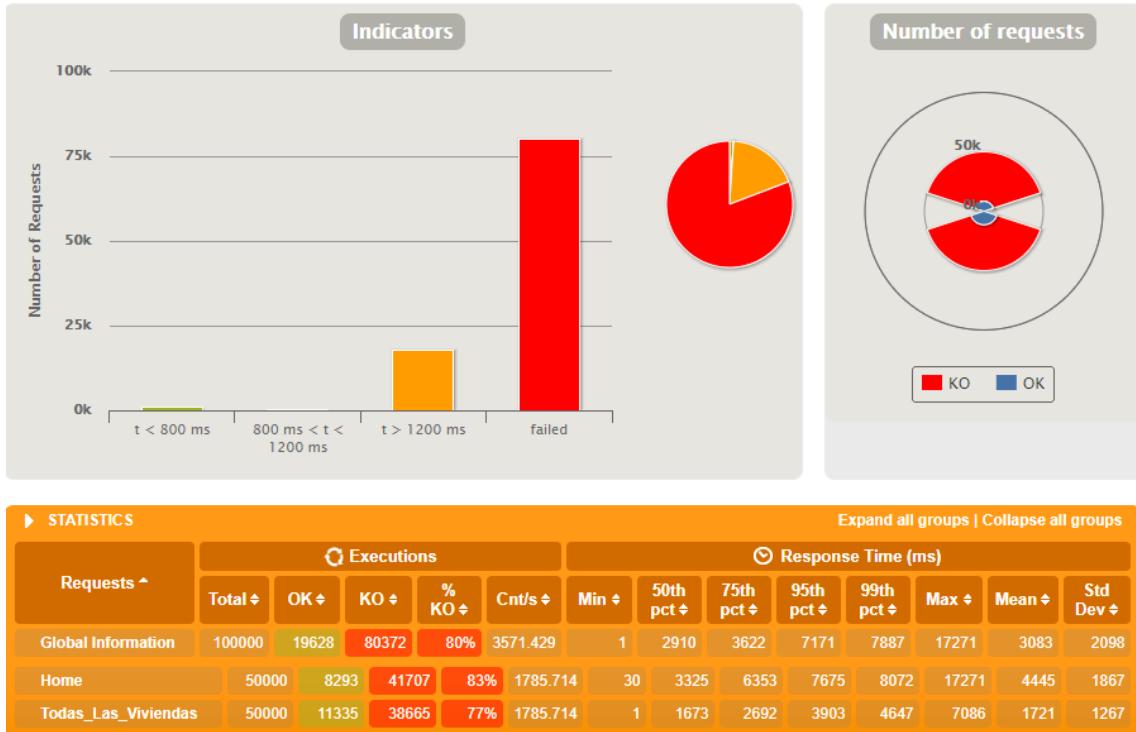


Para un poco más de la mitad de los usuarios se producen fallos, y para la gran mayoría que se producen sin fallos tienen tiempos de espera más largo de lo deseado

### 50.000 usuarios en 10 segundos

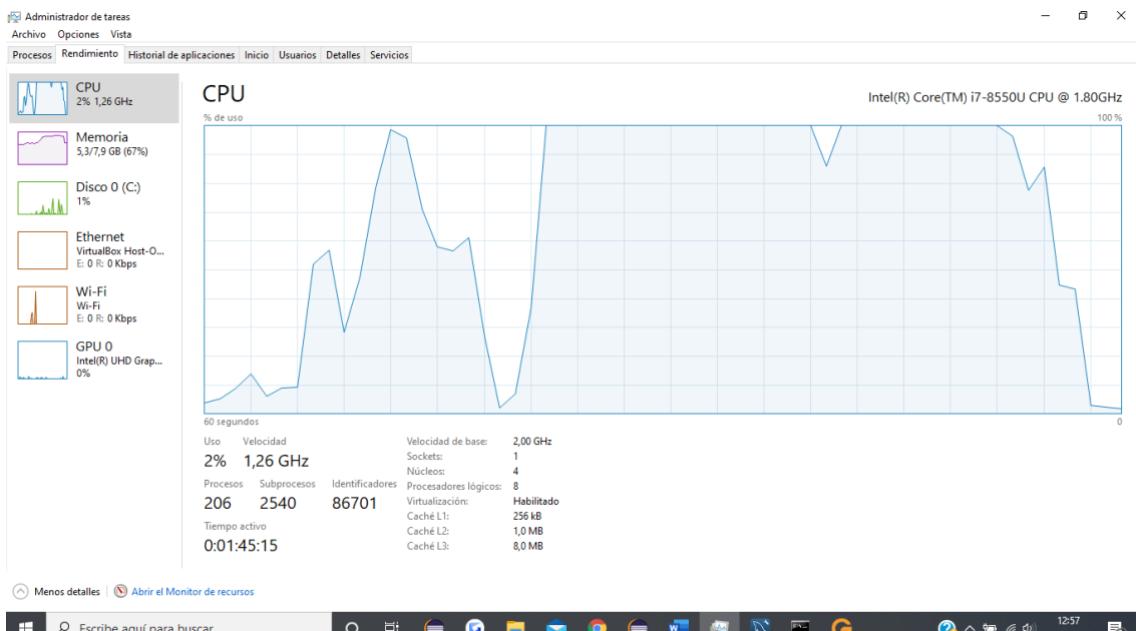


Esta vez los períodos en los que la CPU se encuentra al 100% son como siempre dos, pero esta vez mucho más largos.

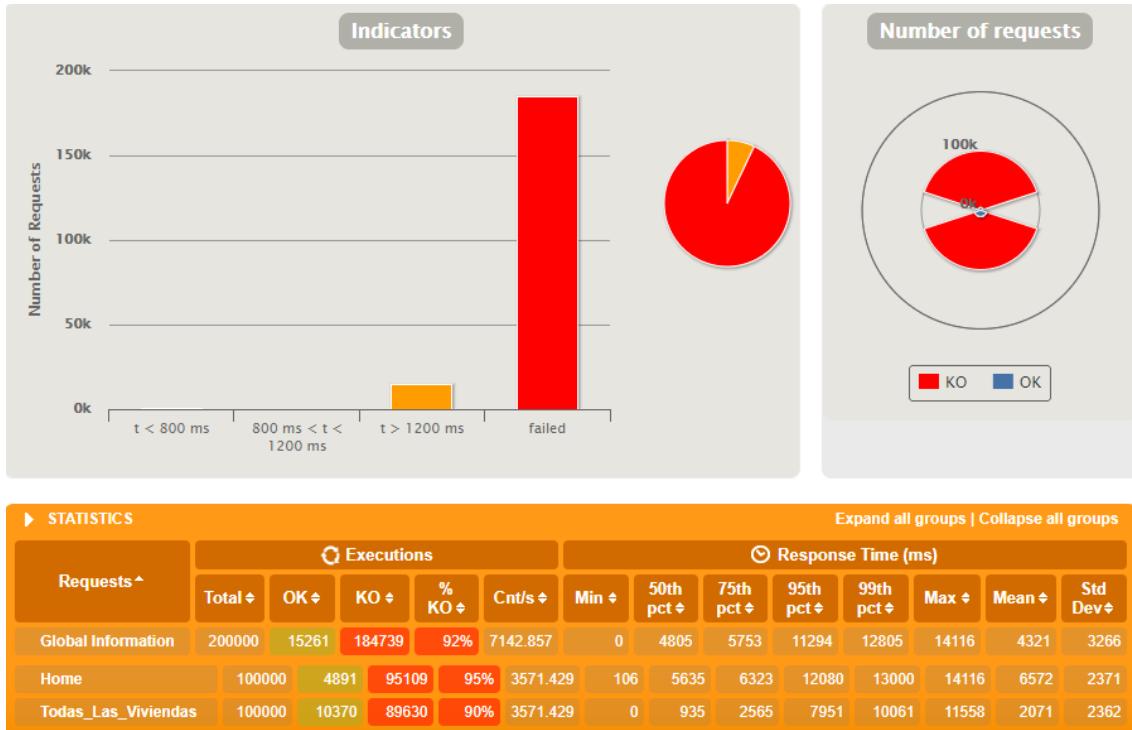


Esta vez la gran mayoría de los usuarios reciben errores, y solo una muy pequeña parte tienen tiempos de respuesta menores a 800 ms, lo cual lo hace una mala prueba.

### 100.000 usuarios en 10 segundos



Esta vez la CPU se pone al 100% durante un solo periodo y casi de forma constante.



Para 100.000 usuarios podemos apreciar que casi todos reciben fallos y solo una pequeña parte lo reciben de forma satisfactoria con largos periodos de espera.

### 400.000 usuarios en 10 segundos

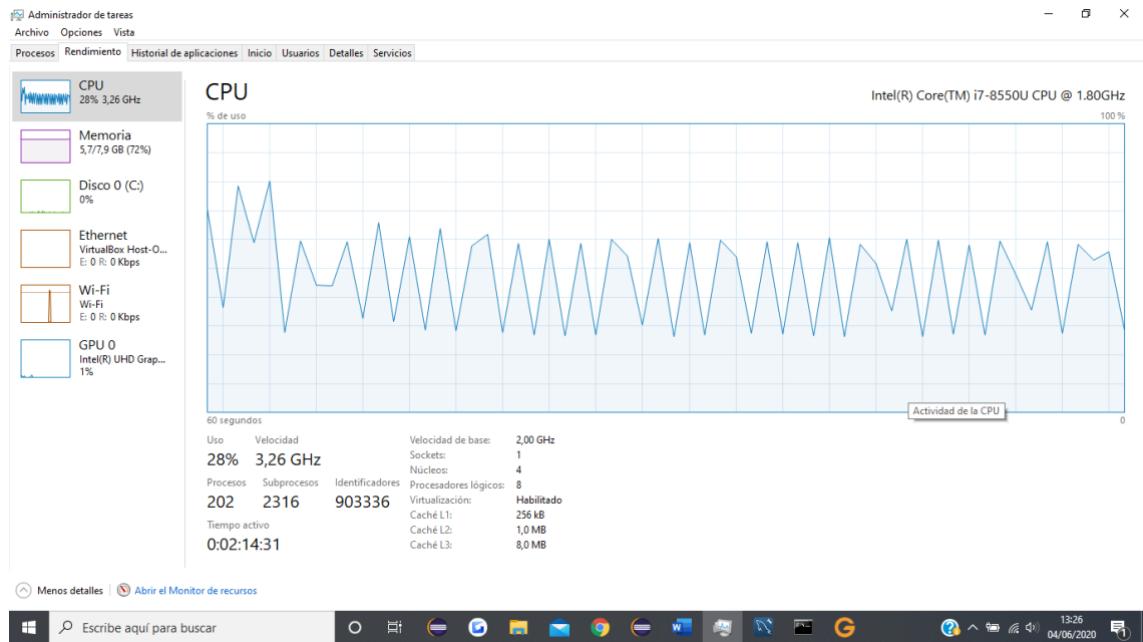
```
GATLING_HOME is set to "C:\gatling-charts-highcharts-bundle-3.3.1-bundle\gatling-charts-highcharts-bundle-3.3.1"
JAVA = ""C:\Program Files\Java\jdk1.8.0_221\bin\java.exe"""
Choose a simulation number:
[0] computerdatabase.BasicSimulation
[1] computerdatabase.advanced.AdvancedSimulationStep01
[2] computerdatabase.advanced.AdvancedSimulationStep02
[3] computerdatabase.advanced.AdvancedSimulationStep03
[4] computerdatabase.advanced.AdvancedSimulationStep04
[5] computerdatabase.advanced.AdvancedSimulationStep05
[6] dp2.TodasLasViviendasPerfomanceTest
[7] dp2.ValoracionVisitaTest
6
Select run description (optional)

Simulation dp2.TodasLasViviendasPerfomanceTest started...
13:07:06.696 [WARN ] i.g.c.c.i.Injector - unhandled event Tick in state StoppedInjecting
java.lang.OutOfMemoryError: Java heap space
Dumping heap to java_pid11648.hprof ...
Heap dump file created [1773779583 bytes in 14.770 secs]
```

Finalmente con 400.000 usuarios el sistema es incapaz de realizar la prueba.

## HU-013

En primer lugar, he comenzado probando 10 usuarios en 10 segundos, para comprobar el correcto funcionamiento de la prueba con una cantidad mínima de usuarios. El rendimiento obtenido es el siguiente:



Se mantiene un uso de la CPU y de la memoria constante.

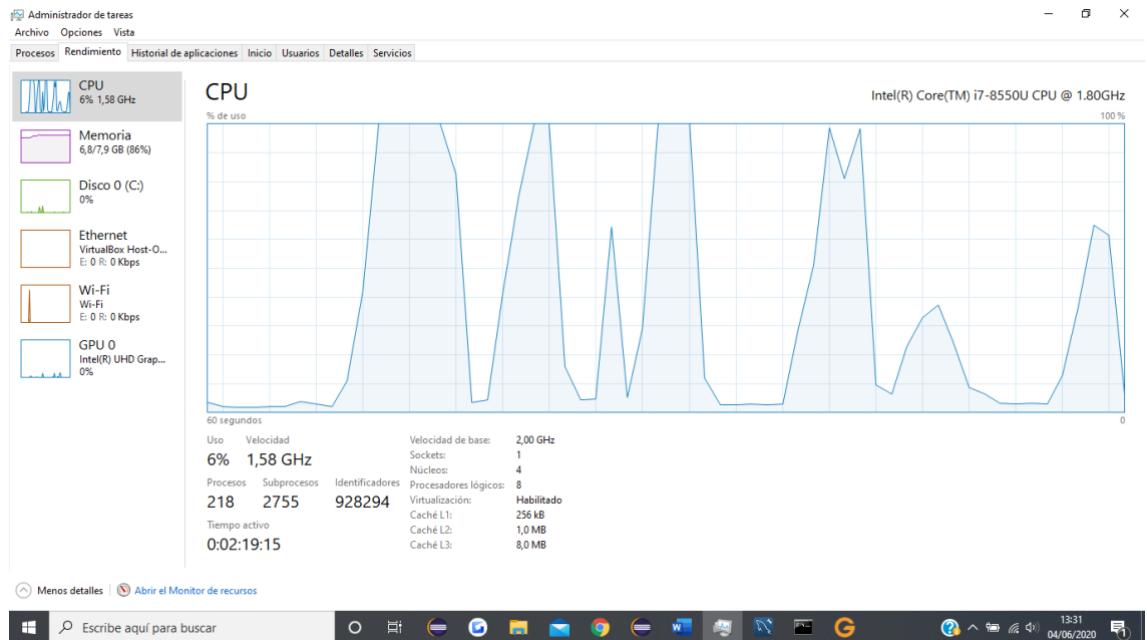


**▶ STATISTICS** Expand all groups | Collapse all groups

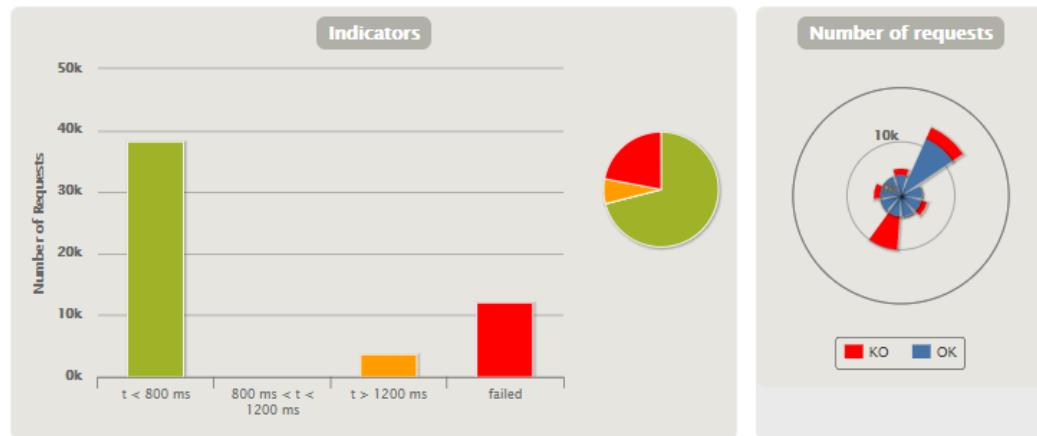
Requests ^	🕒 Executions					⌚ Response Time (ms)								
	Total ↴	OK ↴	KO ↴	% KO ↴	Cnt/s ↴	Min ↴	50th pct ↴	75th pct ↴	95th pct ↴	99th pct ↴	Max ↴	Mean ↴	Std Dev ↴	
Global Information	120	120	0	0%	2.222	0	2	5	600	809	834	89	200	
Home	10	10	0	0%	0.185	69	73	74	82	88	89	74	5	
Login	30	30	0	0%	0.556	1	4	5	8	10	11	4	2	
Login Redirect 1	10	10	0	0%	0.185	1	2	2	2	2	2	2	0	
Lista_Visitas	10	10	0	0%	0.185	2	2	3	4	4	4	2	1	
Lista_Vi...direct 1	10	10	0	0%	0.185	1	2	2	4	4	4	2	1	
Formulario_Visita	20	20	0	0%	0.37	1	346	692	825	832	834	385	385	
Formular...direct 1	10	10	0	0%	0.185	0	1	1	1	1	1	1	0	
Valorar_Visita	10	10	0	0%	0.185	1	2	4	6	7	7	3	2	
Valorar_-direct 1	10	10	0	0%	0.185	0	2	3	6	7	7	2	2	

Se puede observar que todos los usuarios reciben una respuesta aceptable, aunque un pequeño porcentaje de los usuarios tienen un tiempo de espera un poco mayor.

## 5000 usuarios en 10 segundos



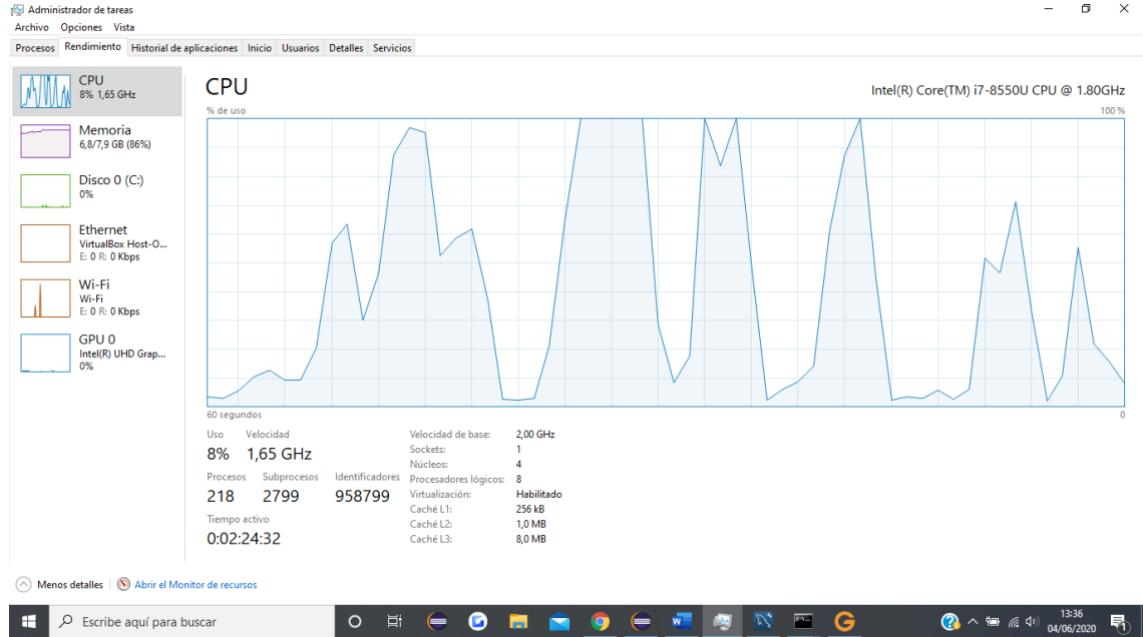
Se puede observar que como en pruebas anteriores la CPU vuelve a intercalar 100% de uso con usos muy bajos cuando se realiza la prueba con 5000 usuarios.



Requests ▲	STATISTICS												<a href="#">Expand all groups</a>   <a href="#">Collapse all groups</a>	
	Executions				Response Time (ms)									
Requests ▲	Total	OK	KO	% KO	Cnt/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev	
Global Information	54135	42097	12038	22%	933.362	0	6	18	3282	3722	4373	288	886	
Home	5000	3827	1173	23%	86.207	118	3380	3595	3825	4071	4373	2915	925	
Login	13827	11481	2346	17%	238.397	0	15	36	440	511	640	57	119	
Login Redirect 1	3827	3827	0	0%	65.983	0	7	14	31	45	54	10	10	
Lista_Visitas	5000	3827	1173	23%	86.207	0	5	9	19	26	33	7	5	
Lista_Vi...direct 1	3827	3827	0	0%	65.983	0	5	11	20	28	35	7	7	
Formulario_Visita	10000	3827	6173	62%	172.414	0	3	5	13	24	33	4	4	
Formular...direct 1	3827	3827	0	0%	65.983	0	4	8	14	20	25	5	4	
Valorar_Visita	5000	3827	1173	23%	86.207	0	5	9	19	27	32	7	6	
Valorar_...direct 1	3827	3827	0	0%	65.983	0	4	8	20	31	37	7	7	

Con 5000 usuarios ya se empiezan a producir algunos fallos, pero aún así hay un gran porcentaje de usuarios que reciben el tiempo de espera deseado.

### 20.000 usuarios en 10 segundos



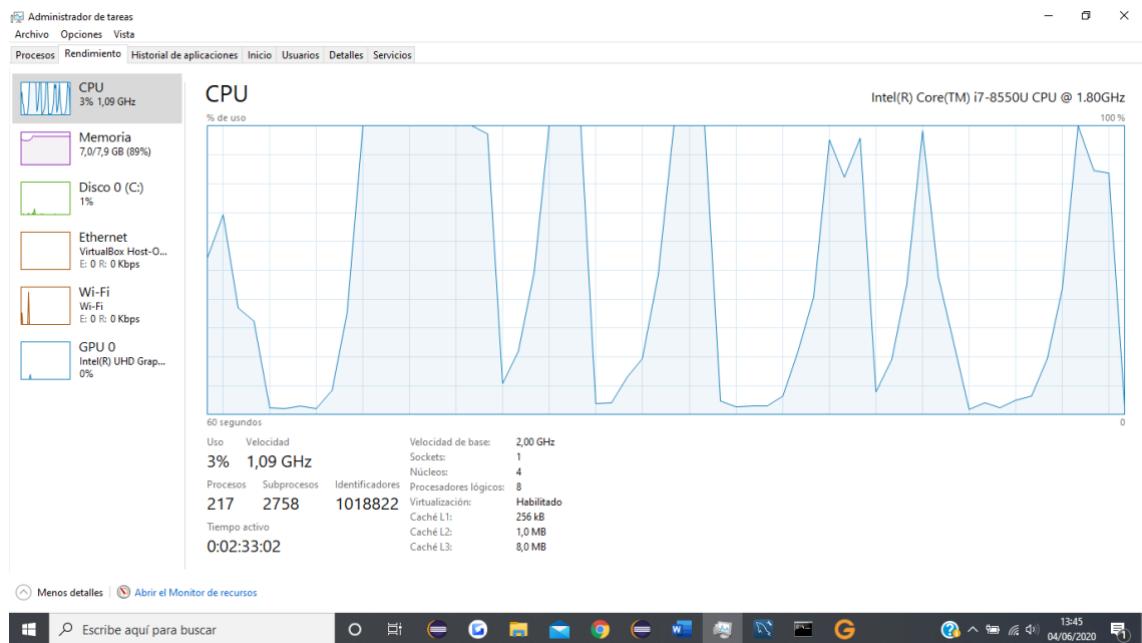
Como siempre, la CPU se intercala entre consumos muy grandes y muy pequeños. El uso de la memoria casi siempre se mantiene constante.



▶ 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		150135	42097	117038	74%	2743.707	0	6	24	1640	4417	4998	268	775	
Home		20000	3827	16173	81%	344.828	32	1538	1936	4491	4680	4998	2011	1133	
Login		43827	11481	32346	74%	755.638	0	9	24	158	277	393	30	56	
Login Redirect 1		3827	3827	0	0%	65.983	0	13	30	60	76	97	21	20	
Lista_Visitas		20000	3827	16173	81%	344.828	0	4	7	14	24	60	5	5	
Lista_Vi...direct 1		3827	3827	0	0%	65.983	0	4	8	10	13	64	5	3	
Formulario_Visita		40000	3827	36173	90%	689.655	0	3	6	18	30	95	5	6	
Formular...direct 1		3827	3827	0	0%	65.983	0	5	9	28	56	74	8	10	
Valorar_Visita		20000	3827	16173	81%	344.828	0	7	25	77	126	268	18	27	
Valorar_...direct 1		3827	3827	0	0%	65.983	0	7	21	31	40	56	12	10	

Con 20.000 usuarios la mayoría de usuarios reciben errores, sin embargo, hay más usuarios que reciben el tiempo de espera deseado que los que tienen un tiempo de espera mayor. Por la gran cantidad de errores el caso en el que haya tantos usuarios no es nada favorable en esta HU.

50.000 usuarios en 10 segundos



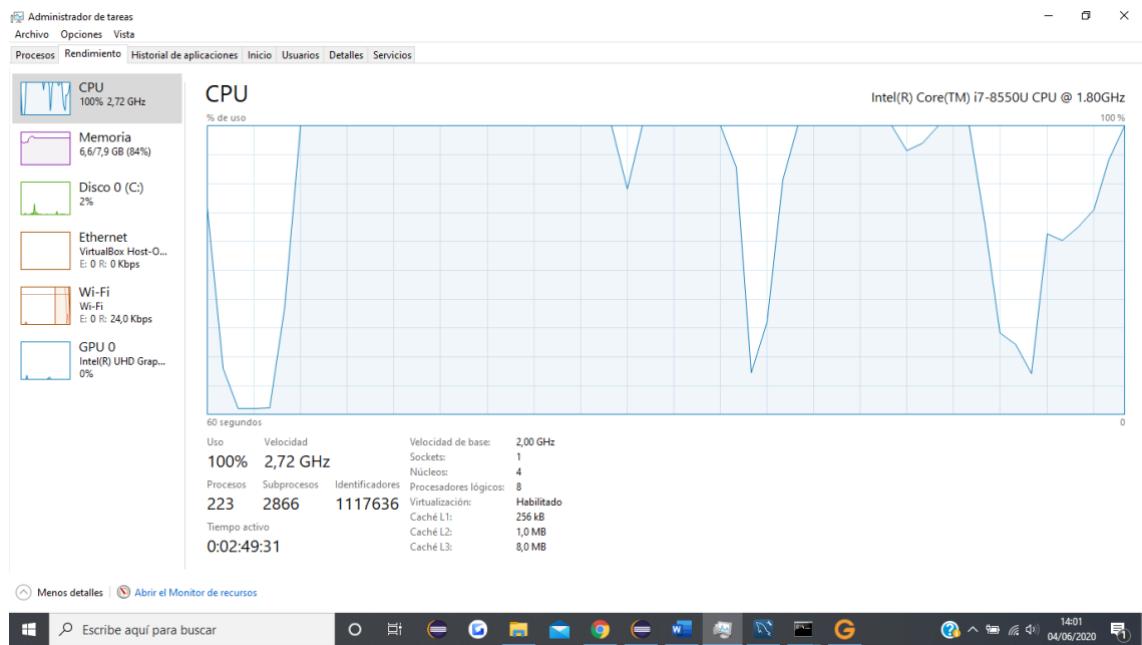
Se puede observar que durante todo el intervalo se intercalan usos de CPU al 100% y porcentajes muy bajos, como en todas las pruebas que se han realizado en este equipo.



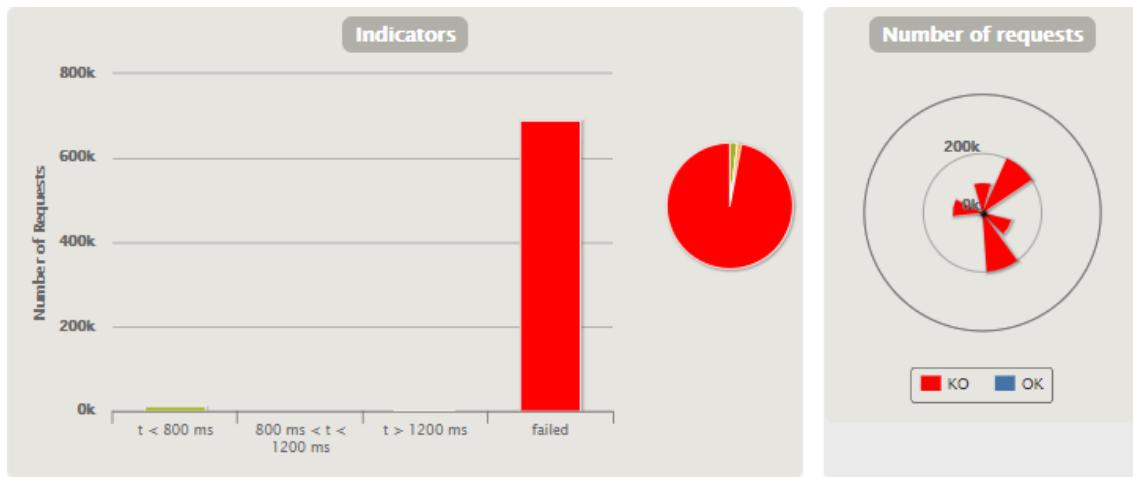
► STATISTICS											Expand all groups   Collapse all groups						
Requests ^	Executions					Response Time (ms)											
	Total ♦	OK ♦	KO ♦	% KO ♦	Cnt/s ♦	Min ♦	50th pct ♦	75th pct ♦	95th pct ♦	99th pct ♦	Max ♦	Mean ♦	Std Dev ♦				
Global Information	360135	42097	327038	89%	6152.25	0	6	26	2988	5840	7372	436	1141				
Home	50000	3827	46173	92%	833.333	48	2906	3081	6510	7016	7372	3135	1084				
Login	103827	11481	92348	89%	1730.45	0	6	19	68	111	827	16	24				
Login Redirect 1	3827	3827	0	0%	63.783	0	27	75	101	120	130	40	36				
Lista_Visitas	50000	3827	46173	92%	833.333	0	4	8	22	36	63	6	8				
Lista_Vi...direct 1	3827	3827	0	0%	63.783	0	7	10	20	26	30	7	6				
Formulario_Visita	100000	3827	96173	96%	1666.667	0	4	11	47	83	127	10	17				
Formular...direct 1	3827	3827	0	0%	63.783	0	5	8	11	14	43	5	5				
Valorar_Visita	50000	3827	46173	92%	833.333	0	3	10	74	242	357	15	40				
Valorar_...direct 1	3827	3827	0	0%	63.783	0	11	17	33	43	75	13	11				

Con 50.000 usuarios casi todos reciben un error. Los que reciben una respuesta correcta tienen un tiempo de espera adecuado y solo una pequeña cantidad de los usuarios tienen tiempos de espera largos.

### 100.000 usuarios en 10 segundos



Esta vez el consumo de la CPU es casi en todo momento del 100%, teniendo muy pocos segundos en los que esta sea menor.

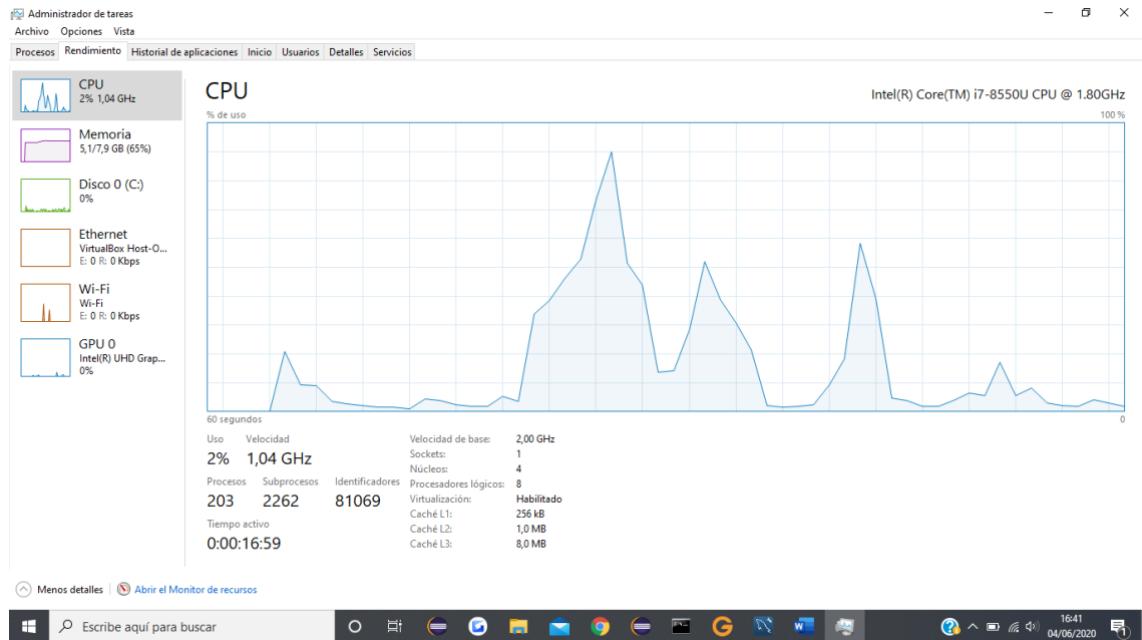


STATISTICS											Expand all groups   Collapse all groups			
Requests ^	Executions					Response Time (ms)								
	Total ♦	OK ♦	KO ♦	% KO ♦	Cnt/s ♦	Min ♦	50th pct ♦	75th pct ♦	95th pct ♦	99th pct ♦	Max ♦	Mean ♦	Std Dev ♦	
Global Information	706269	16357	689912	98%	9416.92	0	136	507	5935	6609	17314	1020	2086	
Home	100000	1253	98747	99%	1333.333	1311	5759	6054	6845	12104	13170	5762	1425	
Login	201253	3759	197494	98%	2683.373	0	146	283	868	1966	4337	247	355	
Login Redirect 1	1253	1253	0	0%	16.707	0	4	10	272	501	503	34	98	
Lista_Visitas	100000	1253	98747	99%	1333.333	0	218	337	887	2158	2877	305	374	
Formulario_Visita	200000	3821	196179	98%	2666.667	0	12	366	973	5635	17314	289	854	
Lista_Vi...direct 1	1253	1253	0	0%	16.707	0	53	218	524	525	525	114	144	
Formular...direct 1	1255	1255	0	0%	16.733	0	5	18	85	98	105	14	24	
Valorar_Visita	100000	1255	98745	99%	1333.333	0	10	37	351	1028	1405	60	157	
Valorar_...direct 1	1255	1255	0	0%	16.733	0	74	91	203	365	366	80	70	

Casi todos los usuarios reciben un error y muy pocos de ellos reciben una respuesta con el tiempo de espera adecuado.

## HU-022

En primer lugar, he comenzado probando 10 usuarios en 10 segundos, para comprobar el correcto funcionamiento de la prueba con una cantidad mínima de usuarios. El rendimiento obtenido es el siguiente:

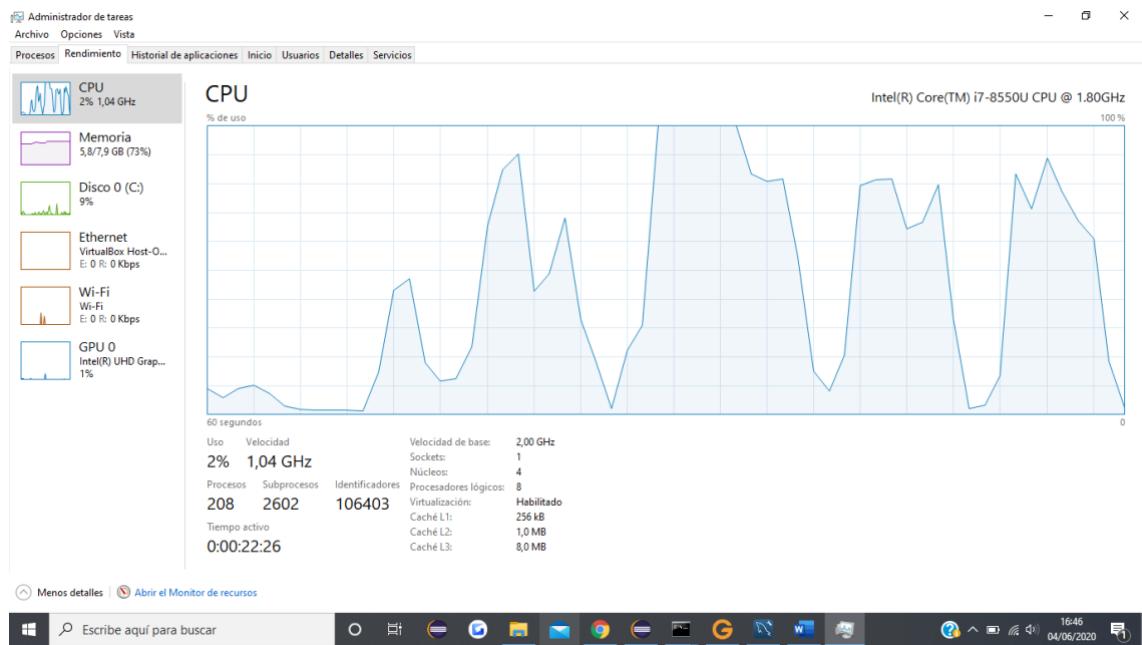


El consumo del a CPU sube momentáneamente casi alcanzando el 100%, luego se va normalizando poco a poco.



Se puede observar que todas las peticiones de los usuarios se realizan en el tiempo deseado y no se produce ningún error, confirmando que la Historia de Usuario funciona correctamente.

## 5.000 usuarios en 10 segundos



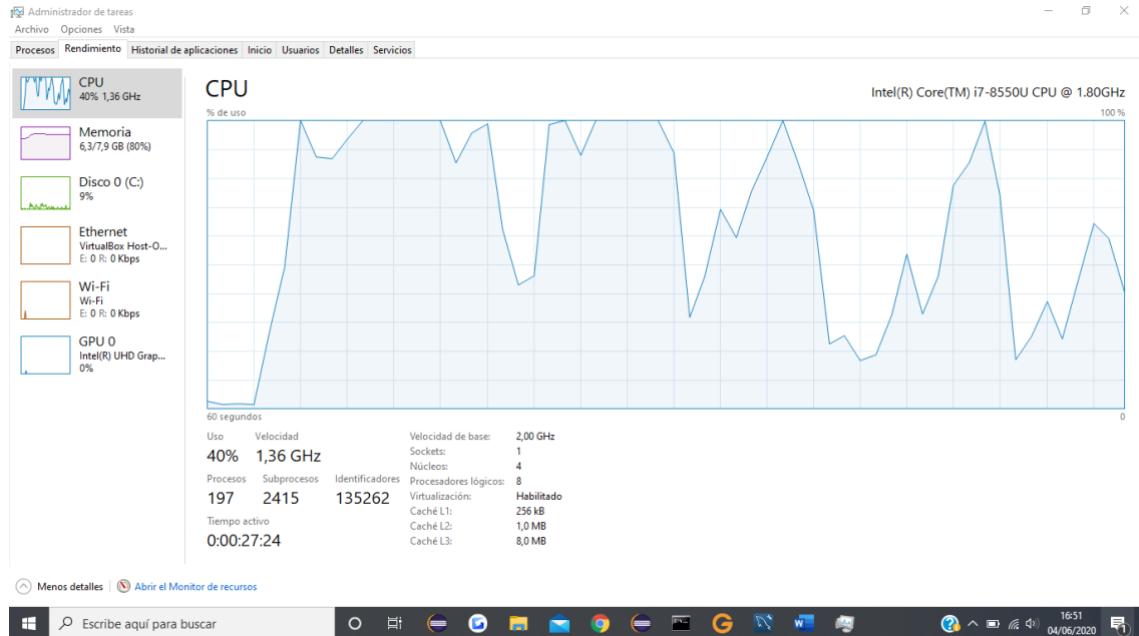
Se puede observar que el consumo del CPU es del 100% en ciertos instantes de la prueba, cuando las primeras peticiones de los usuarios se realizan el consumo se normaliza hasta llegar a mínimos.



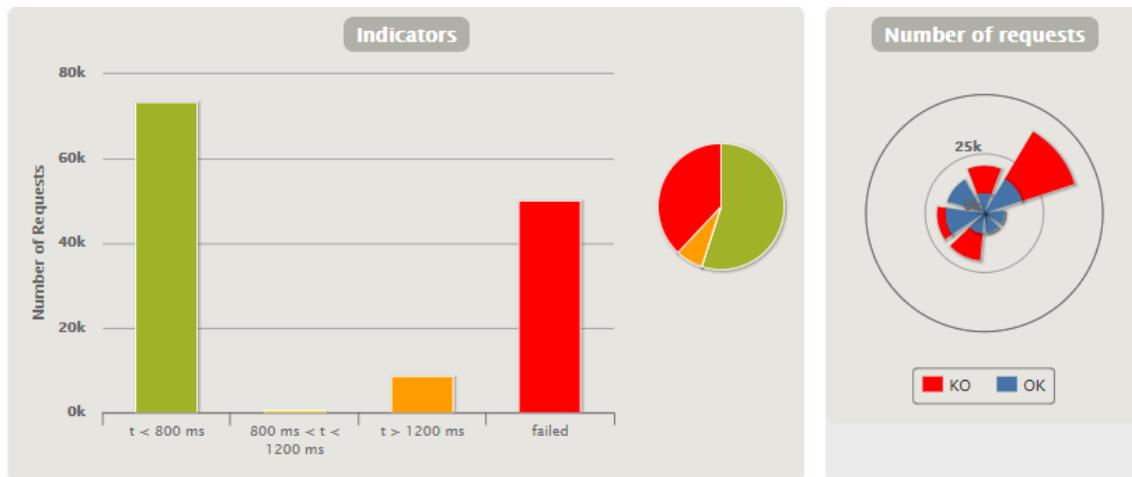
► STATISTICS											Expand all groups   Collapse all groups						
Requests ▲	🕒 Executions					⌚ Response Time (ms)											
	Total ▲	OK ▲	KO ▲	% KO ▲	Cnt/s ▲	Min ▲	50th pct ▲	75th pct ▲	95th pct ▲	99th pct ▲	Max ▲	Mean ▲	Std Dev ▲				
Global Information	40000	39979	21	0%	740.741	0	14	68	5463	7251	8225	640	1723				
Home	5000	4979	21	0%	92.593	1190	4768	6516	7483	7799	8225	4904	1718				
Login	10000	10000	0	0%	185.185	1	22	54	161	260	515	43	56				
request_2	5000	5000	0	0%	92.593	0	41	79	162	222	274	54	54				
Login Redirect 1	5000	5000	0	0%	92.593	0	6	31	116	172	281	25	39				
Mi_Perfil	5000	5000	0	0%	92.593	2	13	51	177	209	242	40	55				
Descarga_PDF	5000	5000	0	0%	92.593	0	4	8	29	49	75	8	9				
Descarga...direct 1	5000	5000	0	0%	92.593	0	3	7	32	54	82	7	10				

Con 5000 usuarios la aplicación aún funciona correctamente en este equipo, sin embargo, ya se empiezan a apreciar retardos indeseados

### 20.000 usuarios en 10 segundos



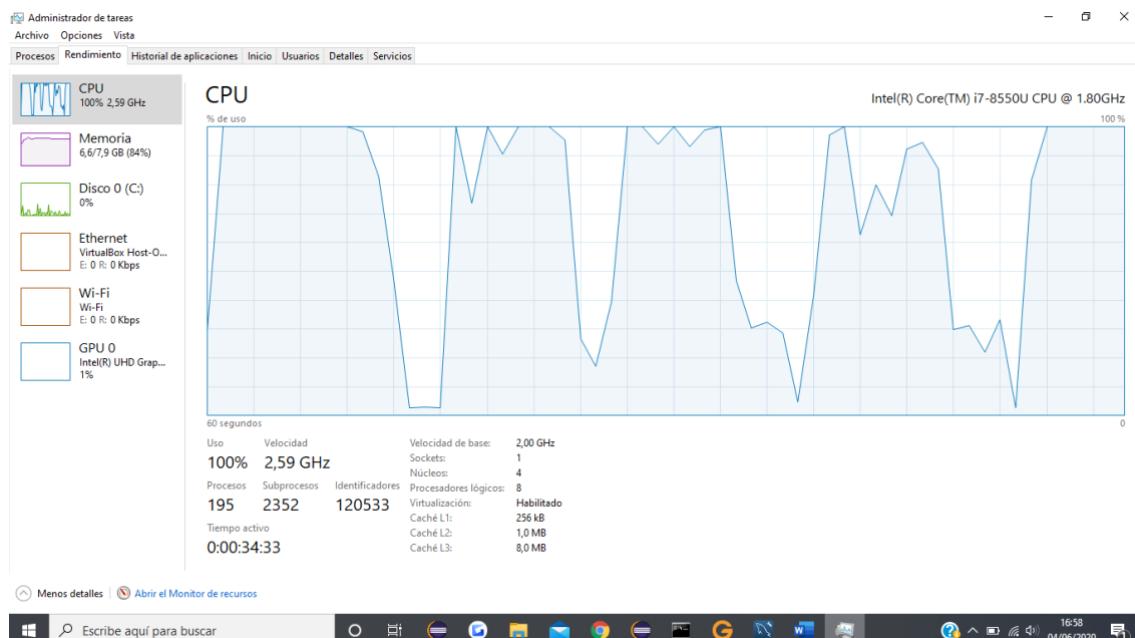
Esta vez el consumo del a CPU tarda mucho más en normalizarse pues se tienen que realizar todas las peticiones de los 20.000 usuarios.



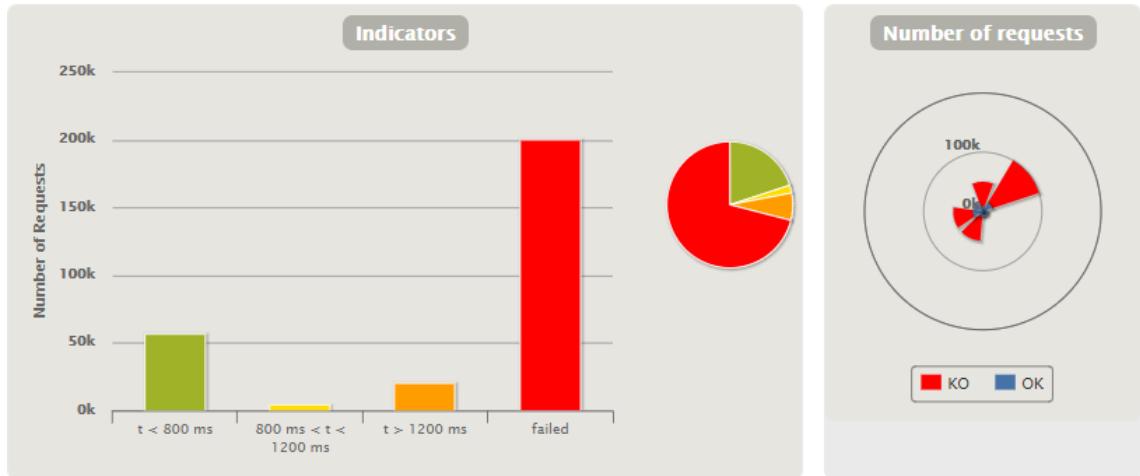
STATISTICS											Expand all groups   Collapse all groups				
Requests ^	Executions					Response Time (ms)									
	Total	OK	KO	% KO	Cnt/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev		
Global Information	133060	82980	50080	38%	1371.753	0	265	2144	8393	9576	49071	1812	3039		
Home	20000	8369	11631	58%	206.186	108	7697	8689	9734	10460	49071	7280	4367		
Login	40000	16738	23262	58%	412.371	0	2007	2071	2453	2751	3233	1299	1017		
request_2	8369	8369	0	0%	86.278	0	71	141	305	671	1108	108	121		
Login Redirect 1	8369	8369	0	0%	86.278	0	15	67	194	319	426	49	70		
Mi_Perfil	20000	8491	11509	58%	206.186	0	2004	2052	2258	2387	2679	1263	976		
Descarga_PDF	20000	16322	3678	18%	206.186	1	81	410	5112	5975	6432	784	1605		
Descarga...direct 1	16322	16322	0	0%	168.268	0	18	103	262	692	3389	80	151		

Con 20.000 usuarios ya aparecen una buena cantidad de fallos, pero aún así sigue siendo mayor el número de usuarios con respuesta adecuada y tiempo de espera deseado.

## 50.000 usuarios en 10 segundos



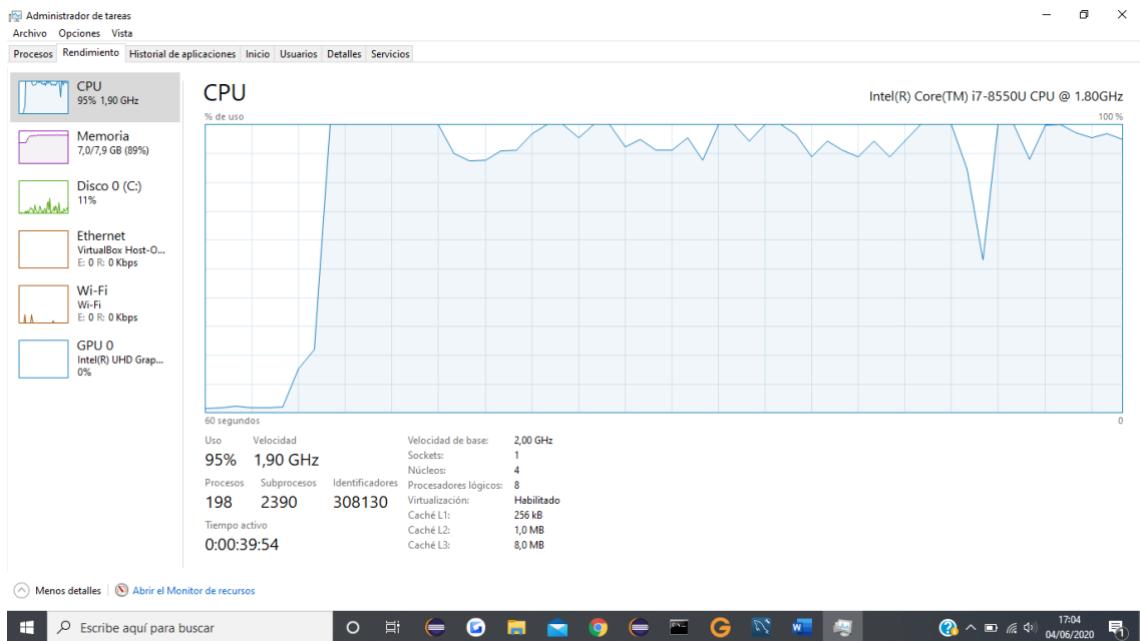
Se pueden observar muchas más intercalaciones entre el 100% del consumo de la CPU y un uso mínimo. También se puede ver un incremento en el uso de la memoria del equipo.



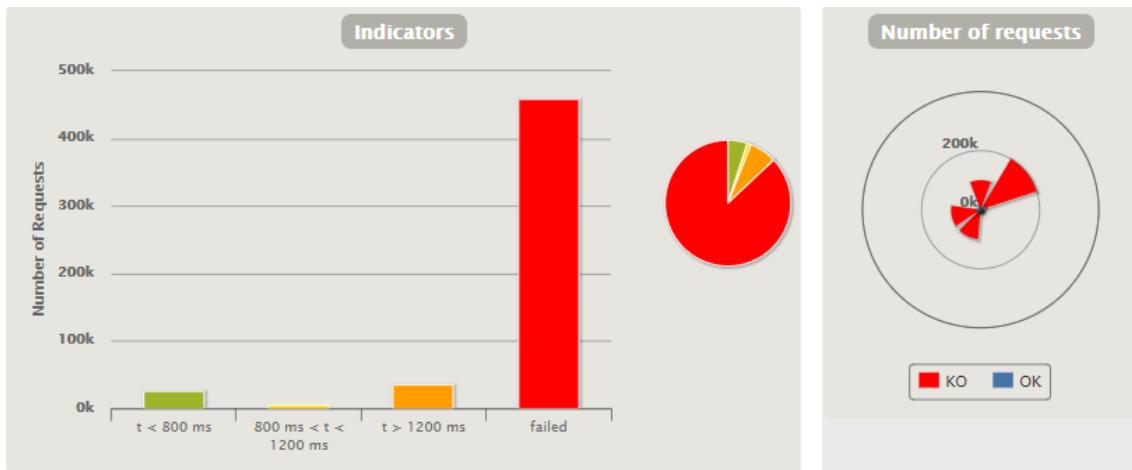
▶ STATISTICS											Expand all groups   Collapse all groups						
Requests ^	🕒 Executions					🕒 Response Time (ms)											
	Total ♦	OK ♦	KO ♦	% KO ♦	Cnt/s ♦	Min ♦	50th pct ♦	75th pct ♦	95th pct ♦	99th pct ♦	Max ♦	Mean ♦	Std Dev ♦				
Global Information	282909	82404	200505	71%	2694.371	0	2007	2875	8292	10029	54400	2108	2582				
Home	50000	8293	41707	83%	476.19	2693	3873	8312	10035	10396	54400	5658	3355				
Login	100000	16586	83414	83%	952.381	0	858	2152	2762	3265	5682	1242	1037				
request_2	8293	8293	0	0%	78.981	0	185	434	751	1583	2976	284	320				
Login Redirect 1	8293	8293	0	0%	78.981	0	154	436	1114	1379	1947	303	344				
Mi_Perfil	50000	8293	41707	83%	476.19	0	1595	2062	2389	2838	3265	1151	1039				
Descarga_PDF	50000	16323	33677	67%	476.19	0	1102	3040	6254	8225	10707	2080	2174				
Descarga...direct 1	16323	16323	0	0%	155.457	0	213	506	8564	9096	9131	1410	2590				

Con 50.000 usuarios sí que se puede ver como casi todos los usuarios reciben un error, mientras que un pequeño porcentaje reciben la respuesta en menos de 800 ms, aunque los usuarios que lo reciben en un tiempo superior son menores.

### 100.000 usuarios en 10 segundos



Con 100.000 usuarios el consumo de la CPU se mantiene casi siempre constante, empleando el 100% de la misma.

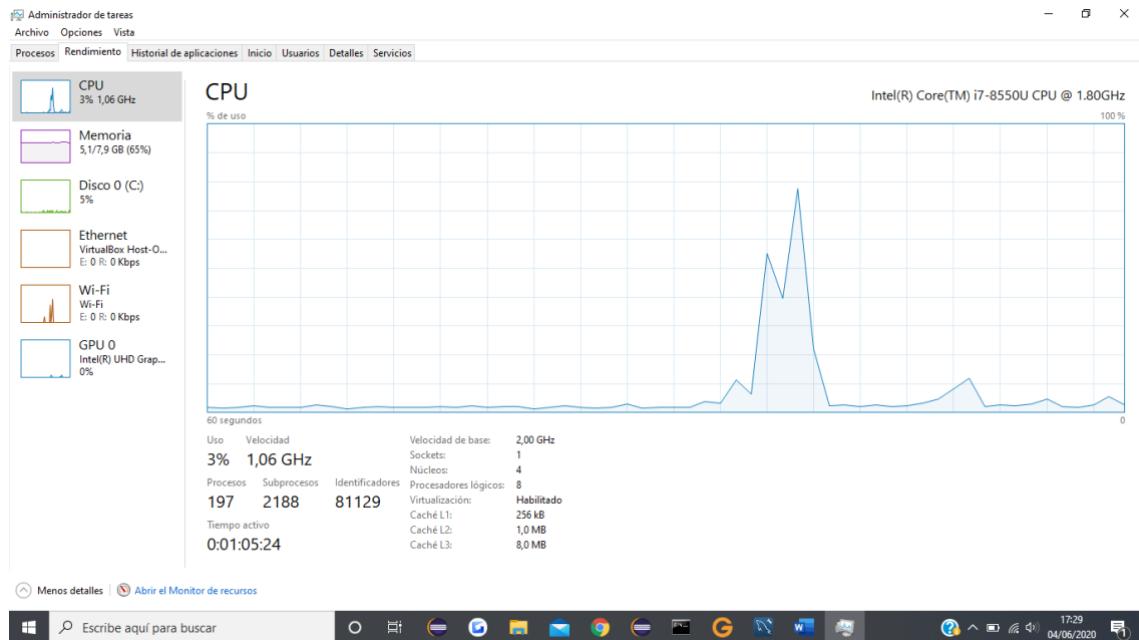


▶ STATISTICS												Expand all groups   Collapse all groups			
Requests ^	🕒 Executions					🕒 Response Time (ms)									
	Total ♦	OK ♦	KO ♦	% KO ♦	Cnt/s ♦	Min ♦	50th pct ♦	75th pct ♦	95th pct ♦	99th pct ♦	Max ♦	Mean ♦	Std Dev ♦		
Global Information	526878	68613	458265	87%	5488.312	0	1771	4772	12065	16259	36936	3369	3753		
Home	100000	6487	93513	94%	1041.667	3741	7168	8769	16319	20874	25963	8403	4015		
Login	200000	16470	183530	92%	2083.333	0	1777	2660	9848	12915	36936	2655	2815		
request_2	8178	8178	0	0%	85.188	0	1142	2477	12551	12745	21295	2989	4053		
Login Redirect 1	8292	8292	0	0%	86.375	0	576	1450	1861	2012	3189	866	649		
Mi_Perfil	100000	8370	91630	92%	1041.667	0	1370	1997	5625	9880	10511	1806	1770		
Descarga_PDF	100000	10408	89592	90%	1041.667	0	1064	2043	8739	10183	11830	1881	2281		
Descarga...direct 1	10408	10408	0	0%	108.417	0	203	615	893	1603	4686	321	359		

Con 100.000 usuarios la cantidad de errores es mucho mayor que la de aciertos, siendo muy pocos los usuarios que consiguen tener una respuesta satisfactoria.

## HU-005

En primer lugar, he comenzado probando 10 usuarios en 10 segundos, para comprobar el correcto funcionamiento de la prueba con una cantidad mínima de usuarios. El rendimiento obtenido es el siguiente:



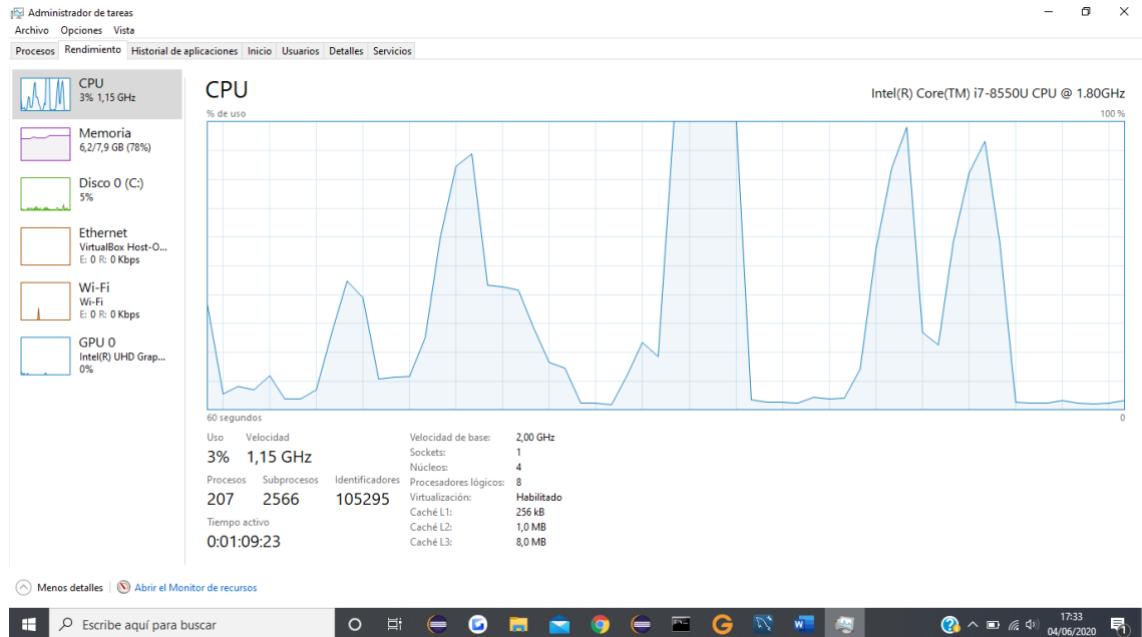
Al momento de ejecutarse esta prueba se utiliza un 70% de la CPU y a los pocos segundos se normaliza en un 3%.



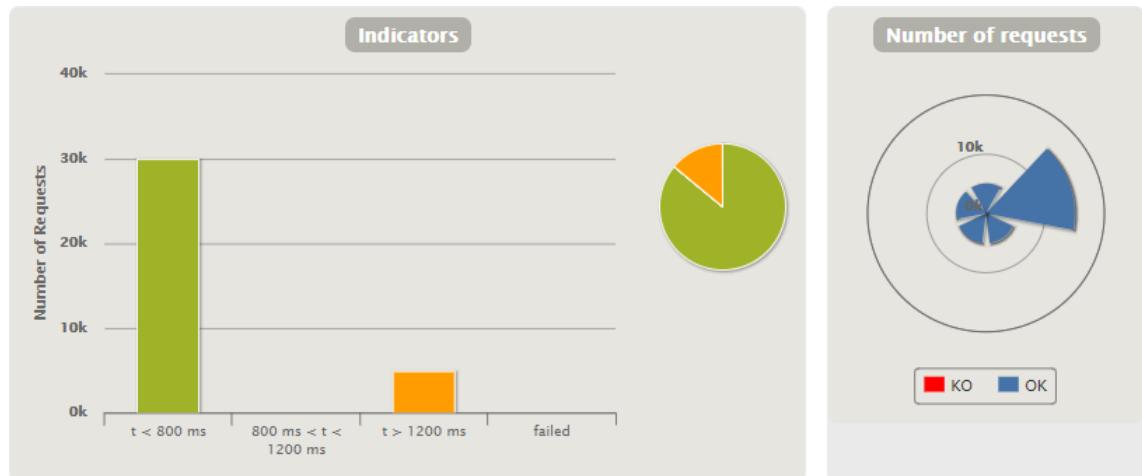
► STATISTICS		Expand all groups   Collapse all groups													
Requests ^		🕒 Executions					⌚ Response Time (ms)								
		Total ♦	OK ♦	KO ♦	% KO ♦	Cnt/s ♦	Min ♦	50th pct ♦	75th pct ♦	95th pct ♦	99th pct ♦	Max ♦	Mean ♦	Std Dev ♦	
Global Information		70	70	0	0%	2.121	2	7	10	59	94	97	15	21	
Home		10	10	0	0%	0.303	56	58	61	95	97	97	65	15	
Login		30	30	0	0%	0.909	2	6	9	11	12	12	6	3	
Login Redirect 1		10	10	0	0%	0.303	3	7	8	12	12	12	7	3	
Lista_Valoraciones		10	10	0	0%	0.303	3	7	8	9	9	9	6	2	
Lista_Va...direct 1		10	10	0	0%	0.303	2	7	7	11	11	11	6	3	

Todos los usuarios tienen un tiempo de respuesta aceptable y no se comete ningún error al confirmar que el caso de uso funciona correctamente.

5.000 usuarios en 10 segundos



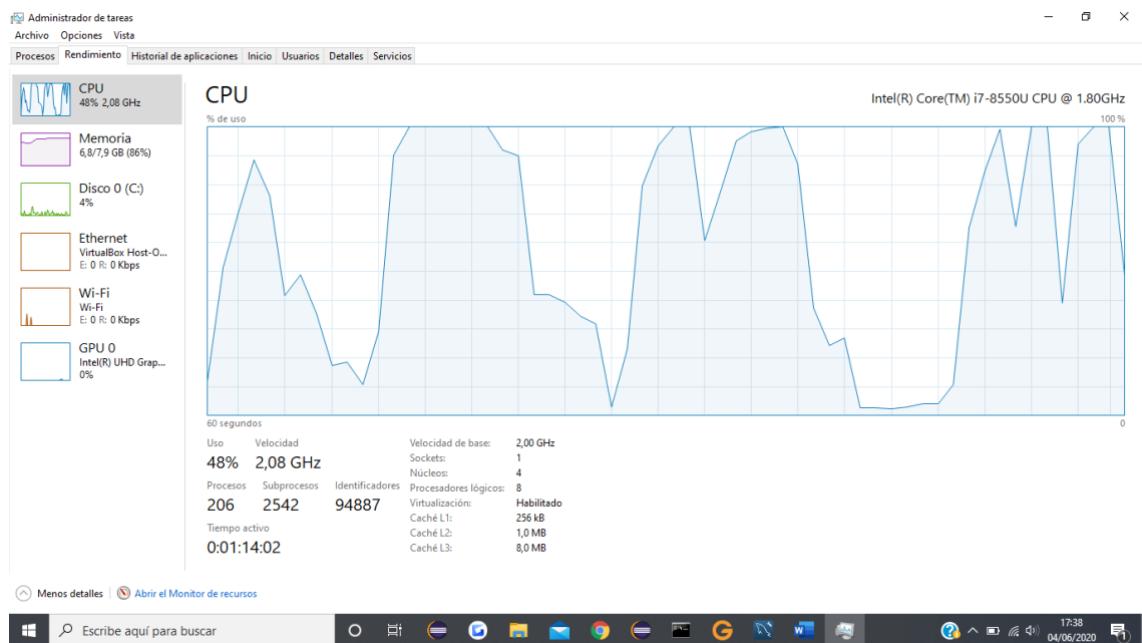
El consumo de la CPU va alternando entre hasta el 100% de su capacidad y el 3% hasta que la prueba finaliza.



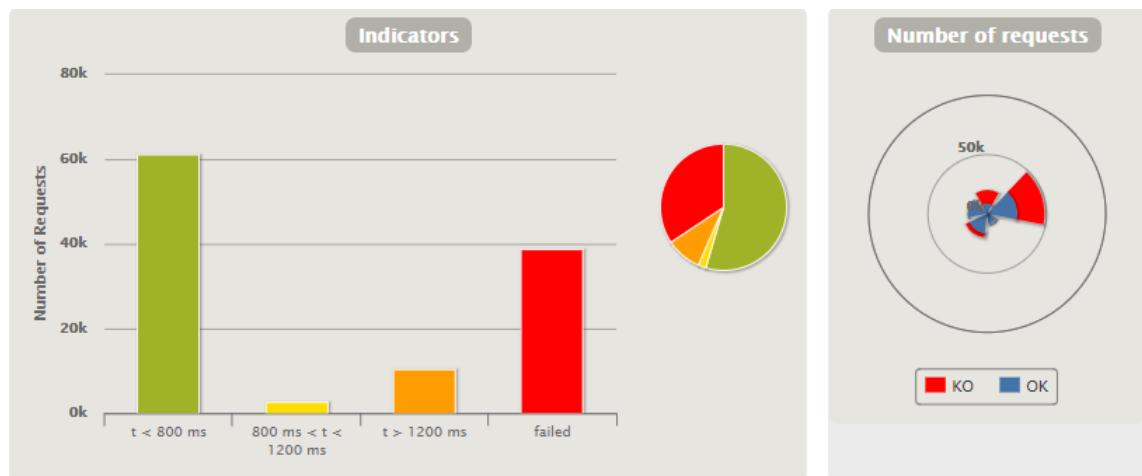
STATISTICS										Expand all groups   Collapse all groups				
Requests <span>▲</span>	Executions					Response Time (ms)								
	Total <span>♦</span>	OK <span>♦</span>	KO <span>♦</span>	% KO <span>♦</span>	Cnt/s <span>♦</span>	Min <span>♦</span>	50th pct <span>♦</span>	75th pct <span>♦</span>	95th pct <span>♦</span>	99th pct <span>♦</span>	Max <span>♦</span>	Mean <span>♦</span>	Std Dev <span>♦</span>	
Global Information	35000	35000	0	0%	921.053	0	46	202	3456	3756	4219	525	1101	
Home	5000	5000	0	0%	131.579	122	3308	3532	3783	4053	4219	3162	536	
Login	15000	15000	0	0%	394.737	0	69	135	365	462	800	100	106	
Login Redirect 1	5000	5000	0	0%	131.579	0	139	324	476	528	1039	169	175	
Lista_Valoraciones	5000	5000	0	0%	131.579	1	15	26	48	94	102	20	17	
Lista_Va...direct 1	5000	5000	0	0%	131.579	0	19	32	43	65	106	20	16	

No se produce ningún error, lo único que ocurre es que algunos usuarios tienen unos tiempos de espera más grandes que otros.

## 20.000 usuarios en 10 segundos



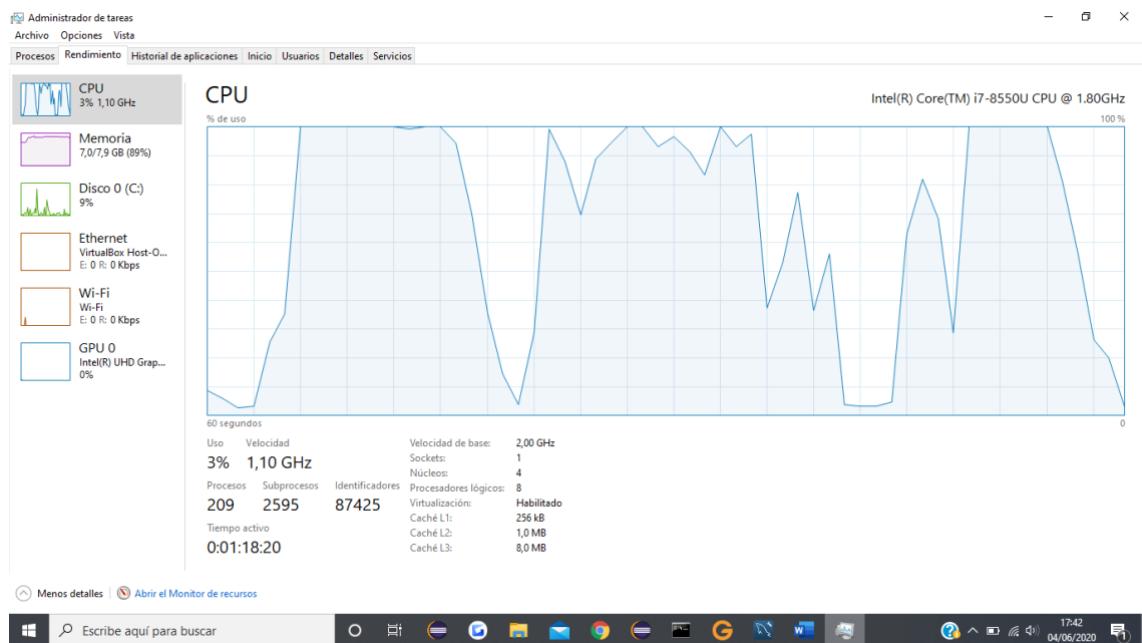
El consumo de la CPU es del 100% casi de forma constante, descontando algunos intervalos en los que el consumo se mantiene bajo. La memoria aumenta ligeramente durante todo el proceso.



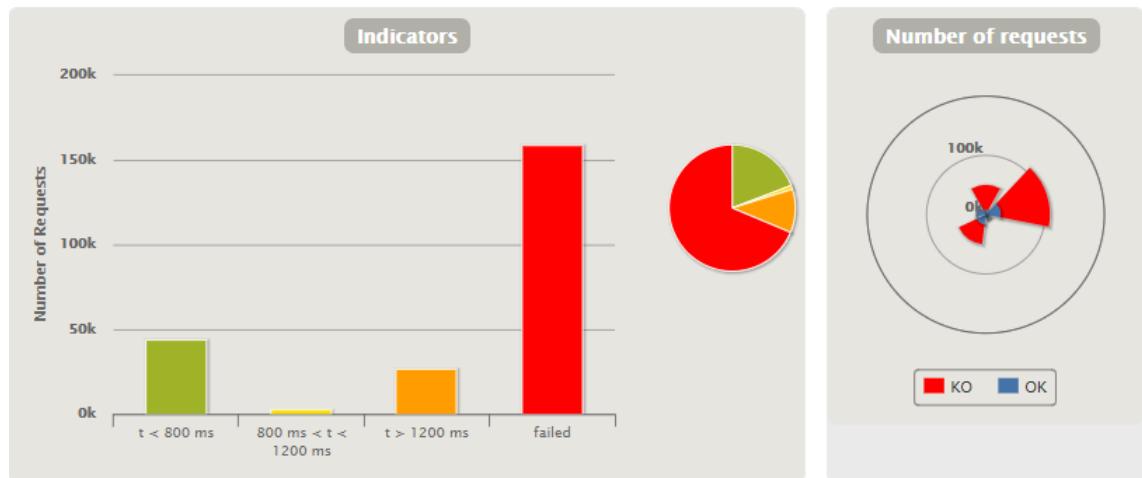
Requests ^	Executions				Response Time (ms)									
	Total ^	OK ^	KO ^	% KO ^	Cnt/s ^	Min ^	50th pct ^	75th pct ^	95th pct ^	99th pct ^	Max ^	Mean ^	Std Dev ^	
Global Information	112915	74126	38789	34%	1590.352	0	520	2707	6349	6688	36857	1773	2345	
Home	20000	8296	11704	59%	281.69	2085	6021	6388	6700	6886	36857	5705	2526	
Login	48296	24888	23408	48%	680.225	0	1367	2370	3001	3317	3577	1401	1121	
Login Redirect 1	8296	8296	0	0%	116.845	0	180	365	526	628	1294	221	172	
Lista_Valoraciones	20000	16323	3677	18%	281.69	1	110	736	3927	4138	4780	724	1202	
Lista_Va...direct 1	16323	16323	0	0%	229.901	0	26	177	373	1569	4276	129	305	

Con 20.000 usuarios ya empieza a haber una gran cantidad de errores. Aún así el número de aciertos es mayor que el de fallos.

## 50.000 usuarios en 10 segundos



Como en todas las pruebas el proceso es el mismo. El consumo de la CPU va variando entre grandes pequeños porcentajes. La memoria como siempre incrementa y se mantiene constante durante el proceso.

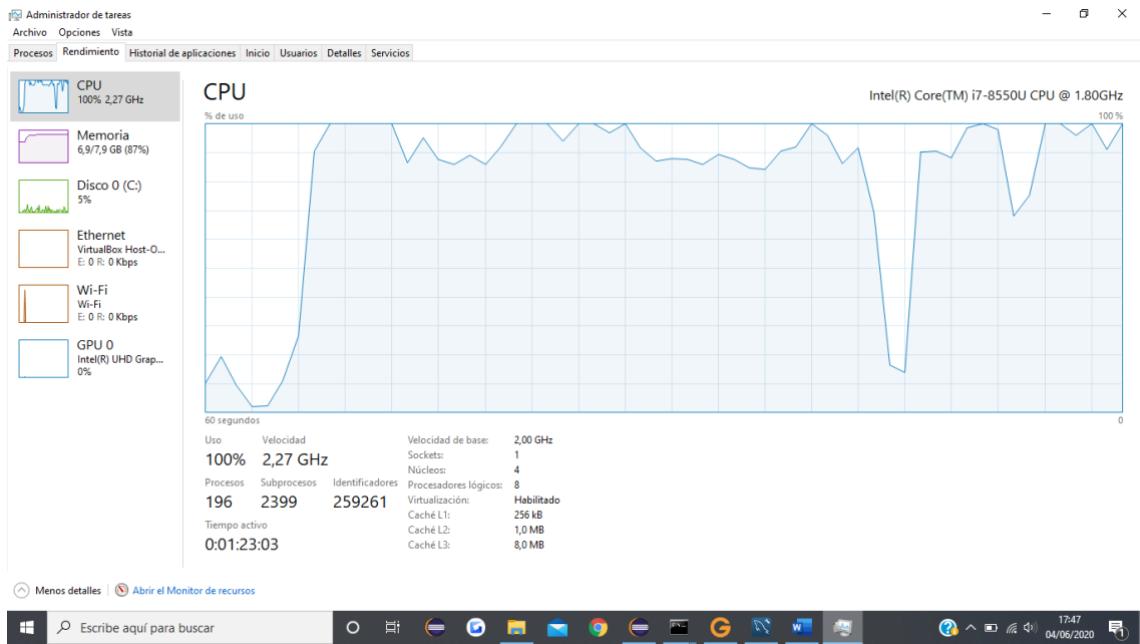


► STATISTICS Expand all groups | Collapse all groups

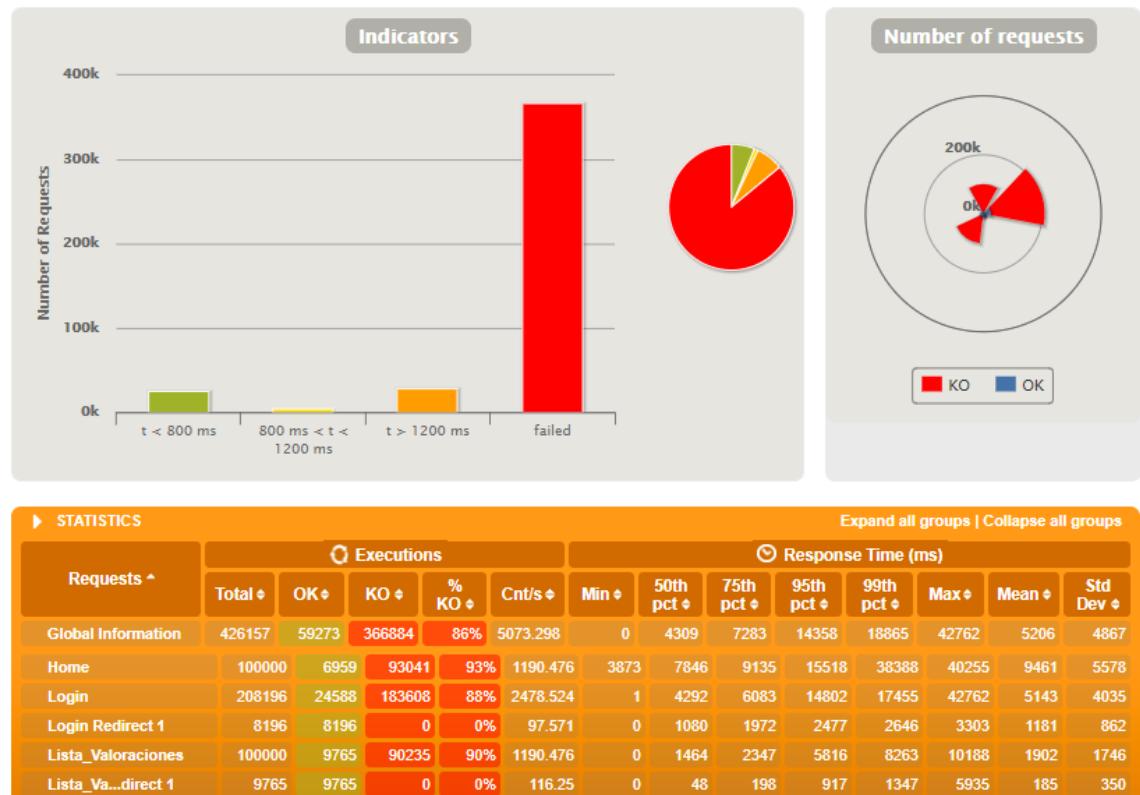
Requests ^	Executions					Response Time (ms)									
	Total ♦	OK ♦	KO ♦	% KO ♦	Cnt/s ♦	Min ♦	50th pct ♦	75th pct ♦	95th pct ♦	99th pct ♦	Max ♦	Mean ♦	Std Dev ♦		
Global Information	232916	74125	158791	68%	3024.883	0	2043	3681	7638	10412	43588	2383	2662		
Home	50000	8293	41707	83%	649.351	2955	4177	7351	10400	10683	43588	5691	2980		
Login	108293	24879	83414	77%	1406.403	0	1550	2221	6429	7418	9461	1820	1928		
Login Redirect 1	8293	8293	0	0%	107.701	0	30	131	1199	1592	2699	195	374		
Lista_Valoraciones	50000	16330	33670	67%	649.351	0	1996	2103	2795	3407	4967	1351	1040		
Lista_Va...direct 1	16330	16330	0	0%	212.078	0	229	326	591	1734	2132	263	253		

Con 50.000 usuarios el número de errores es mucho mayor que el número de aciertos, haciendo que muy pocos usuarios reciban la respuesta deseada. Con esto podemos concluir que a partir de 50.000 usuarios es completamente inviable esta Historia de Usuario en este dispositivo.

## 100.000 usuarios en 10 segundos



Con 100.000 usuarios el consumo de la CPU se mantiene casi en todo momento constante, rondando entre el 80% y el 100%, teniendo algún bajón puntual a los mínimos.



Con 100.000 usuarios el número de aciertos es mínimo en comparación al número de fallos, haciendo completamente inviable esta situación en la realidad.

## Conclusiones

A partir de las pruebas realizadas, podemos concluir que, con las características de mi equipo, el sistema no puede funcionar con 400.000 usuarios. También podemos afirmar que a partir de 20.000 usuarios los resultados no son para nada óptimos ya que el número de fallos es mayor que el número de acierto, cosa inviable en una situación real. El máximo de usuarios que podría tolerar la aplicación sin dar fallos debe rondar los 7.500, mientras que el máximo número de usuarios que se podría permitir sin que el número de errores sea excesivamente grande debe rondar los 10.000 usuarios.

Para mejorar el rendimiento de la aplicación en mi dispositivo creo que habría que mejorar la capacidad de CPU ya que durante las pruebas con más de 5.000 usuarios la capacidad de CPU se encuentra en el 100% casi en todo momento, lo cual no es nada recomendable pues da lugar a errores no deseados.

## Álvaro Esteban Castrillo

HU	Descripción
HU-003	Creación de una vivienda.
HU-011	Denunciar anuncios que sean una estafa.
HU-012	Guardar casas como favoritas.

## CARACTERÍSTICAS DEL SISTEMA

El equipo con el que se han realizado las pruebas ha sido un MacBook Pro de 15", versión 2014. Las características del sistema son:

<b>Processor(Model)/Frecuencia/Núcleos/HT</b>	2,2 GHz Intel Core i7 de 4 núcleos
<b>Memory principal</b>	16 GB 1600 MHz DDR3

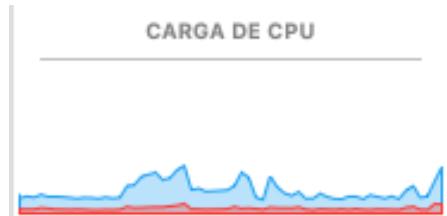
## Procedimiento:

Por cada historia de usuario se han ido probando un número diferente de usuarios, analizando el rendimiento que ofrece el sistema y buscando el número máximo de usuarios para que la aplicación funcione. Una vez encontrados, hemos buscado el número mínimo de usuarios para que la aplicación no funcione.

## RESULTADOS

### HU-003 – Crear vivienda

En primer lugar, he comenzado probando 10 usuarios en 10 segundos, para comprobar el correcto funcionamiento de la prueba con una cantidad mínima de usuarios. La carga de la cpu ha sido:

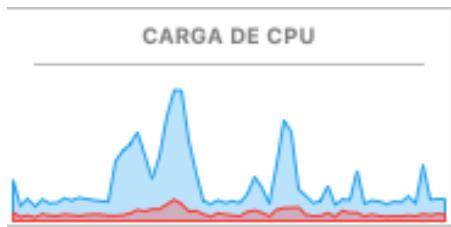


Donde todas las peticiones han sido OK:

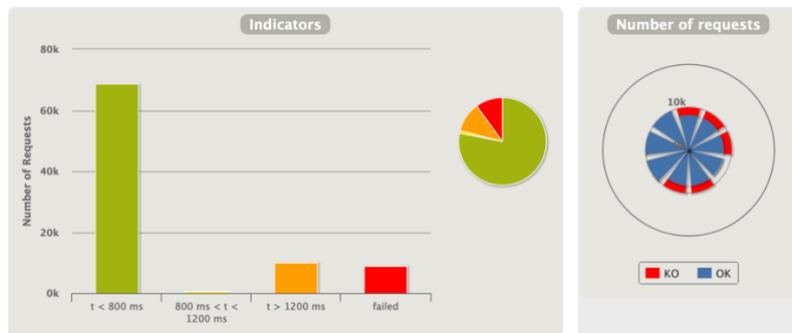


### Con 10.000 usuarios en 10 segundos

La carga de la cpu ha sido :



Y en este caso, ya ha habido ciertas peticiones que no se han podido realizar correctamente:



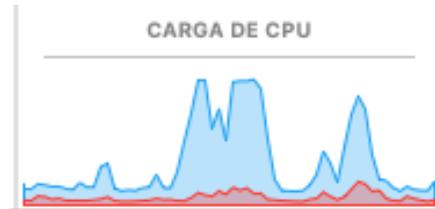
En este caso, ya se tomaría como mala las pruebas puesto que los tiempos de respuesta exceden a los que hemos estimado oportunos:

ASSERTIONS										Status ↴				
Assertion ↴										Status ↴				
Global: max of response time is less than 5000.0 Global: mean of response time is less than 1000.0 Global: percentage of successful events is greater than 95.0														
Expand all groups   Collapse all groups														
STATISTICS		Executions					Response Time (ms)							
Requests ↴		Total ↴	OK ↴	KO ↴	% KO ↴	Cnt/±	Min ↴	50th pct ↴	75th pct ↴	95th pct ↴	99th pct ↴			
Global Information		88192	79352	8840	10%	434.443	0	3	383	10002	10006	80006	1605	3760
home		10000	8192	1808	18%	49.261	1	4	266	10002	10519	60006	2344	6896
login		10000	8192	1808	18%	49.261	0	1	3	10005	10059	10080	1811	3852
logged		10000	8192	1808	18%	49.261	0	2	3	10004	10006	10028	1810	3849
logged Redirect 1		8192	8192	0	0%	40.356	0	1	1	2	3	12	1	1
Todas las viviendas		10000	8292	1708	17%	49.261	7	5051	6990	10004	10006	10013	5024	3095
form nueva vivienda		10000	8292	1708	17%	49.261	0	1002	5084	10003	10005	10012	2984	3697
vivienda creada		10000	10000	0	0%	49.261	0	3	6	35	58	105	7	12
vivienda...direct 1		10000	10000	0	0%	49.261	0	1	3	20	44	66	4	8
show viv...detalles		10000	10000	0	0%	49.261	2	19	335	644	698	810	170	219

En este caso se han tomado como medidas el tiempo máximo de respuesta sea 5 segundos, la media de las respuestas tiene que ser menor a 1 segundo y el porcentaje total de fallos tiene que ser menor al 5%.

### 150.000 usuarios en 10 s

La carga de la cpu ha aumentado considerablemente:



En este caso ya, la mayoría de peticiones ya han sido rechazadas, aunque el sistema sigue funcionando:



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	1066538	78867	987871	93%	5796.402	0	26	1319	7120	10055	81393	1380	2815	
home	150000	8192	141808	95%	815.217	82	6749	7148	8045	15281	81393	6687	2192	
login	150000	8192	141808	95%	815.217	0	25	88	213	10006	10619	200	1146	
logged	150000	8192	141808	95%	815.217	0	5	33	228	10002	10653	158	1046	
logged Redirect 1	8192	8192	0	0%	44.522	0	6	1039	1311	1438	1657	357	547	
Todas las viviendas	150000	8192	141808	95%	815.217	0	2	9	1107	4482	12029	179	800	
form nueva vivienda	150000	8192	141808	95%	815.217	0	1763	2552	3350	10238	12397	1841	1741	
vivienda creada	150000	8348	141654	94%	815.217	0	4	19	146	10019	10128	271	1548	
vivienda...direct 1	8348	8348	0	0%	45.359	0	2	4	18	55	650	5	21	
show viv... details	150000	13023	136977	91%	815.217	0	20	149	672	10010	42423	457	2843	

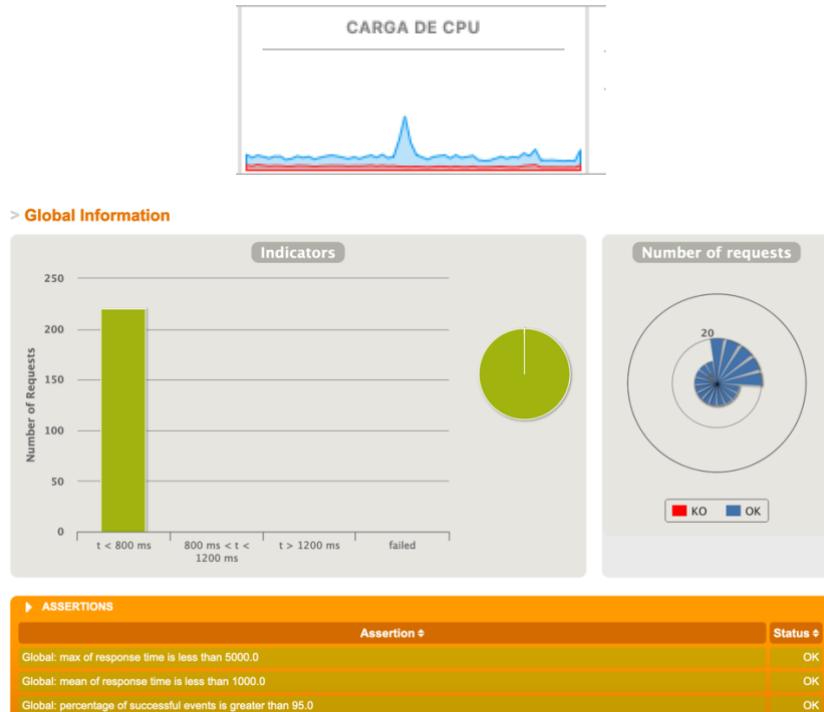
ERRORS		
Error #	Count #	Percentage #
i.n.c.ChannelException: Failed to open a socket.	753449	78.285 %
j.n.UnknownHostException: www.dp2.com	218234	22.096 %
i.n.c.ConnectTimeoutException: connection timed out: www.dp2.com/127.0.0.1:80	12988	1.315 %
j.I.IOException: Premature close	2861	0.29 %
i.g.h.c.RequestTimeoutException: Request timeout to www.dp2.com/127.0.0.1:80 after 60000 ms	100	0.01 %
i.n.c.AbstractChannel\$AnnotatedSocketException: Connection reset by peer: www.dp2.com/127.0.0.1:80	37	0.004 %
j.n.UnknownHostException: www.dp2.com: nodename nor servname provided, or not known	2	0 %

En este caso se puede ver que ya no se puede acceder por la cantidad de peticiones simultáneas que se están produciendo.

## HU-011 y HU-012 – Añadir viviendas a favoritos y denunciar vivienda

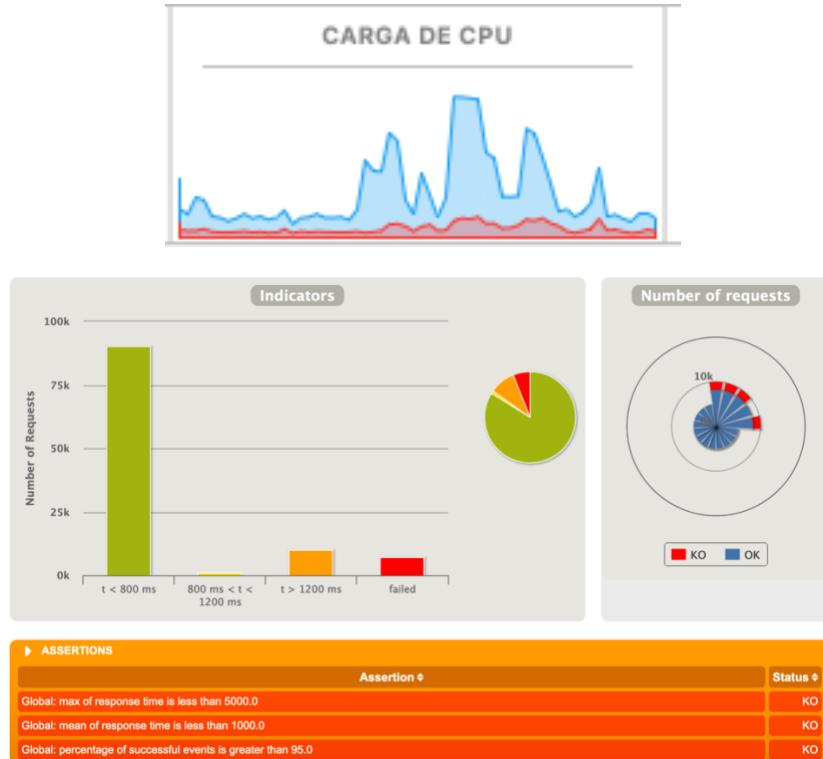
En este caso, se van a realizar la prueba de rendimiento de las dos historias de usuario juntas para ver el rendimiento conjunto. Ambas realizan los mismos pasos al principio de la prueba.

Para ver el funcionamiento normal y lógico que debería tener, se ha realizado una prueba inicial con 10 usuarios cargados en 10 segundos. Los resultados de la carga de CPU y de gatling son los siguientes:



## 5.000 usuarios cargados en 10 s

En este caso podemos ver como ya empieza a fallar algunas peticiones concurrentes. A pesar de que fallan todas las restricciones que hemos puesto, realmente el total de los fallos no ha sido muy elevado. El sistema sigue funcionando bien dentro de un pequeño margen. Veamos la carga de la cpu y los fallos:

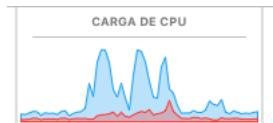


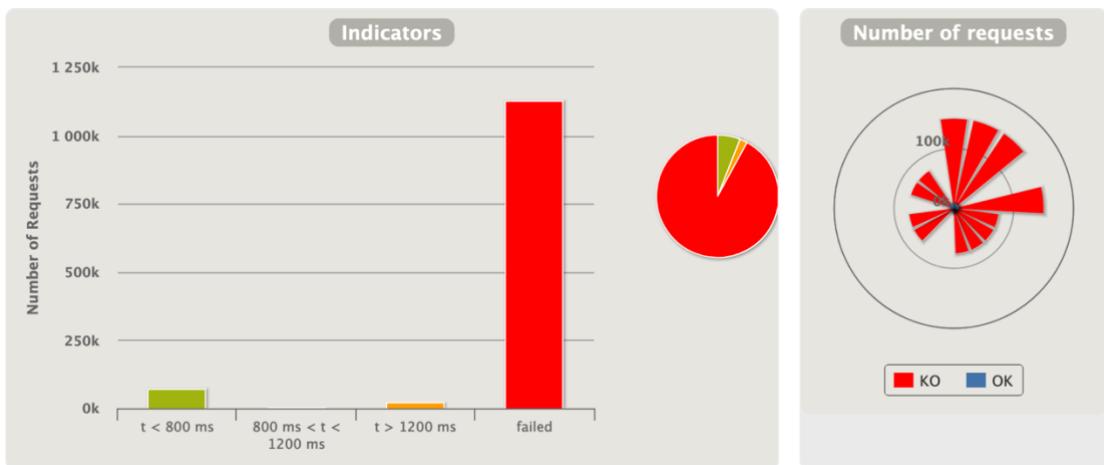
Y los fallos obtenidos han sido:

▶ ERRORS			
Error	Count	Percentage	
i.n.c.ConnectTimeoutException: connection timed out: www.dp2.com/127.0.0.1:80	6927	98.507 %	
i.g.n.c.i.RequestTimeoutException: Request timeout to www.dp2.com/127.0.0.1:80 after 60000 ms	100	1.422 %	
i.n.c.AbstractChannel\$AnnotatedSocketException: Connection reset by peer: www.dp2.com/127.0.0.1:80	4	0.057 %	
j.i.IOException: Premature close	1	0.014 %	

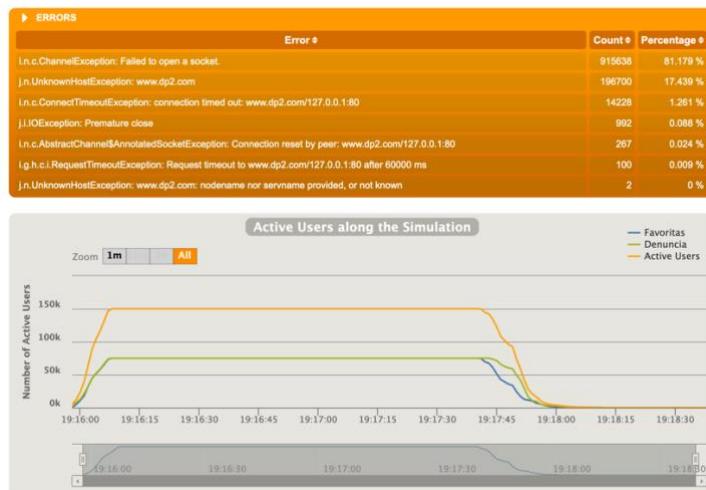
## 75.000 usuarios de forma concurrente cargados en 10 s

En este caso estamos cargando 75.000 usuarios de forma concurrente por cada una de las dos pruebas, haciendo un total de 150.000 en el pico mas alto de usuarios concurrentes. En este caso, las pruebas han dado bastantes mas fallos en las peticiones que peticiones correctas. Podemos ver el comportamiento del sistema a continuación:





A continuación, podemos ver también como ha ido la carga de usuarios en el sistema:

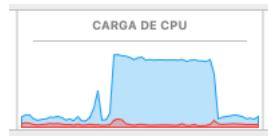


### 150.000 usuarios de forma concurrente cargados en 10 s

En este caso el sistema ya no soporta la carga de usuarios, al contrario que en la prueba anterior. Realmente, el número de usuarios concurrentes que se han intentado cargar han sido 300.000, puesto que se han iniciado 150.000 por cada historia de usuario. La memoria asignada a java ya no puede aguantar mas y no carga la prueba:

```
[6] dp2.DenunciaAndFavoritas
[7] dp2.DenunciaAndFavoritasUsers
[8] dp2.DenunciaAndFavoritasUsersAssertion
[9] dp2.basicTests
[10] dp2.createViviendaTest
[11] dp2.createViviendaTestUsers
[12] dp2.createViviendaTestUsersAssertions
8
Select run description (optional)

Simulation dp2.DenunciaAndFavoritasUsersAssertion started...
19:11:05.444 [WARN ] i.g.c.c.i.Injector - unhandled event Tick in state StoppedI
njecting
java.lang.OutOfMemoryError: Java heap space
Dumping heap to java_pid3001.hprof ...
Unable to create java_pid3001.hprof: Too many open files
./gatling.sh: line 54: 3001 Killed: 9                      "$JAVA" $DEFAULT_JAVA_OPTS
$JAVA_OPTS -cp "$GATLING_CLASSPATH" io.gatling.app.Gatling "$@"
alvaroesteban@MacBook-Pro-de-Alvaro bin %
```



### Conclusión:

Como podemos observar, mi sistema no soporta mas usuarios cuando llega a los 300.000 usuarios concurrentes. Por debajo de esta cantidad, funciona el sistema pero la cantidad de peticiones fallidas es bastante elevada. Por otro lado, Vemos que con 10.000 usuarios concurrentes el sistema es bastante fiable, teniendo muy pocas peticiones fallidas.

Para mejorar el rendimiento de la aplicación, deberíamos de mejorar la cpu de nuestro sistema para que pueda soportar la carga computacional de la aplicación. Además, se podría mejorar la memoria del mismo o dejar la aplicación sola en el entorno, sin ninguna aplicación mas corriendo, para acaparar mas cantidad de espacio de memoria.