****

DP2

PERFORMANCE

|  |
| --- |
| **Grupo -** G2-1 |
| **Miembros del grupo** |
| **DANIEL ARELLANO MARTÍNEZ** |
| **EDUARDO MIGUEL BOTÍA DOMINGO** |
| **JOSE MARTÍN SÁNCHEZ** |
| **JUAN NOGUEROL TIRADO** |
| **JOSÉ MANUEL SÁNCHEZ RUIZ** |
| **JAVIER VÁZQUEZ ZAMBRANO** |

ÍNDICE

[**Airline y Client**](#_30ujbz9nq0zw)3

[**Planes**](#_fjve241d3b59)6

[**Flight**](#_lvu6jn4a3bb9)10

[HU T-09: Trabajador consulta un vuelo](#_xuw0xlbp0x70) 10

[HU C-04: Cliente lista vuelos disponibles](#_uw72s38xjch8) 11

[HU T-07: Trabajador edita un vuelo](#_smxyne7lbk3i) 13

[HU T-08: Trabajador lista vuelos](#_l8pz7ep3itq5) 15

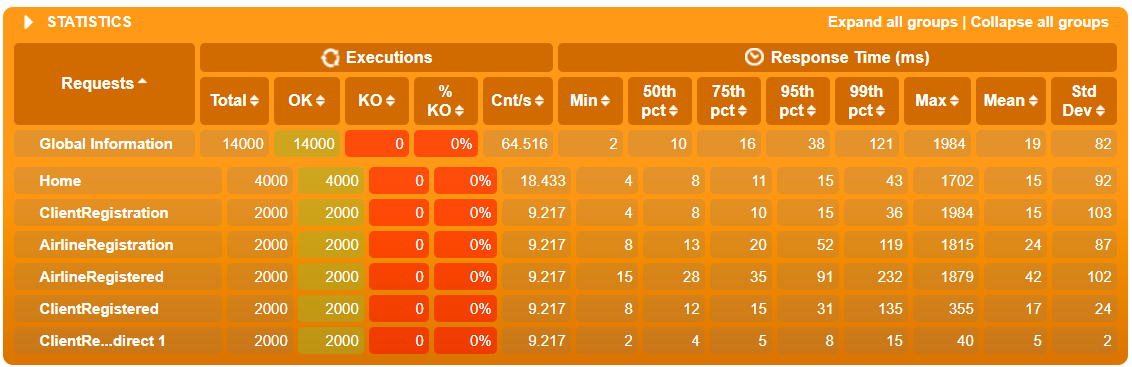
[HU T-06: Trabajador registra un vuelo](#_e916fztiy0vr) 17

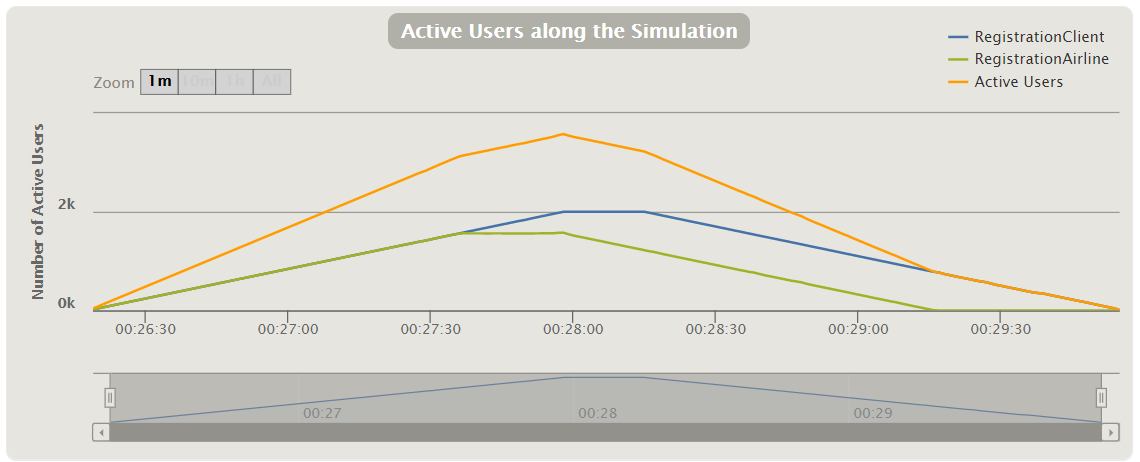
# Airline y Client

Tenemos diversas funcionalidades como el Registro, Acceso y Salida de cada aerolínea. En el caso de Client tenemos exactamente lo mismo que en el caso de Airline.

Los hemos agrupado ya que en la realidad se realizan diversas interacciones entre ambas entidades que vamos a analizar y realizan las mismas operaciones a la hora de registrarse y/o loguearse.

Los test de rendimiento los hemos realizado inicialmente sobre un número bajo de usuarios, en este caso en particular, sobre 2000 aerolíneas y 2000 client. Este número de usuarios concurrentes sabemos de sobra que no va a llegar a crear cuellos de botella. Obtenemos los siguientes resultados:

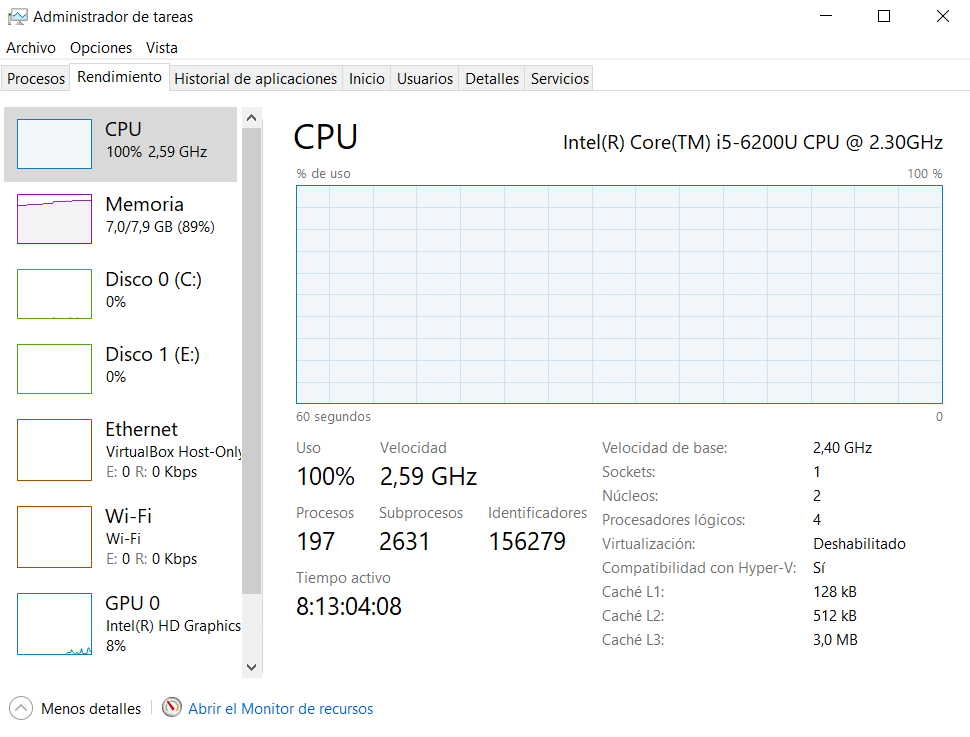




Vemos que el tiempo medio por cada petición es bajo, llegando como máximo a 42 ms en Airline Registered. Como estimamos antes, no ha ocurrido ningún error, lo cual nos indica que tenemos que subir el número de peticiones para ver cuál es el cuello de botella en este equipo.

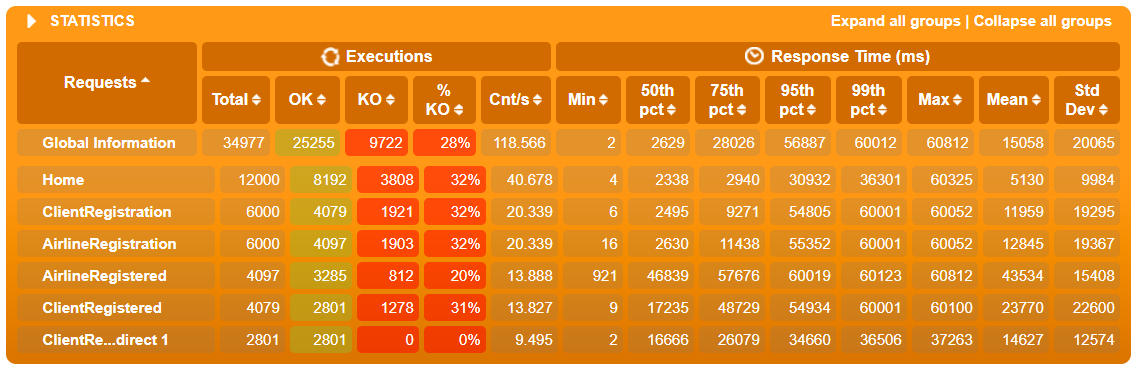
Las pruebas que vamos a realizar se toman con la base de datos de MySQL activa, realizando verdaderamente el rendimiento que va a suponer la aplicación y la base de datos sobre un sistema portátil. Nos aseguramos antes de correr la prueba de que no existen aplicaciones en segundo plano y que el navegador no posea muchas ventanas abiertas (que consumen mucha RAM, usando un navegador basado en Chromium como es Brave).

Vamos a probar ahora con 8000 aerolíneas y 8000 clientes, un número que puede estimarse una producción de cuello de botella. Tras ejecutarlo, obtenemos los siguientes resultados:



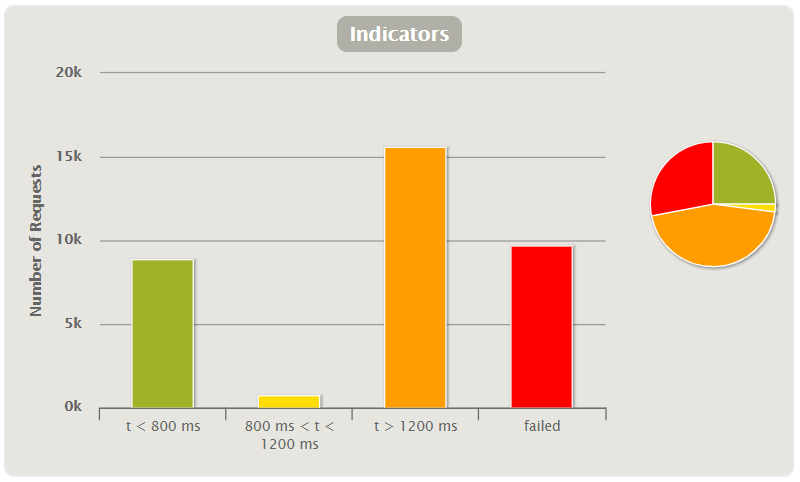
Durante la ejecución del test de rendimiento, en el administrador de tareas vemos que la CPU supone el cuello de botella, ya que se pone este al 100% de su capacidad y no da a basto a responder a todas las peticiones, lo que produce errores.

Ahora veamos las gráficas que nos devuelve la herramienta Gatling:



Podemos observar que ha ocurrido errores (KOs), siendo el menos afectado AirlineRegistered, ya que los demás rondan el 32% de errores relativos.

En cuanto a la media por cada petición se ha visto aumentado el tiempo notablemente, esto es debido a que el procesador produce un cuello de botella y evita que se respondan a tiempo cada una de las peticiones.



Como podemos apreciar, aproximadamente un cuarto de las peticiones se ha realizado correctamente, en un tiempo moderado (t<800 ms). Mientras que casi la mitad ha respondido de manera lenta y el otro cuarto restante se ha producido errores.

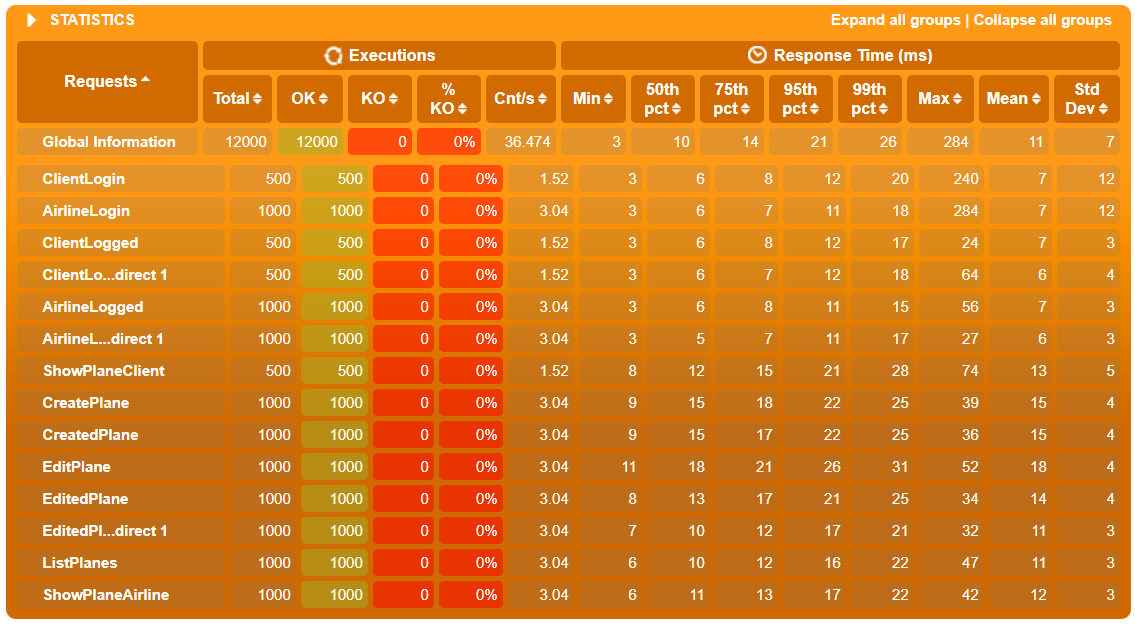
Por lo tanto, para reducir este número de peticiones fallidas o respondidas de manera lenta, debería de adquirirse un equipo con un mejor procesador.

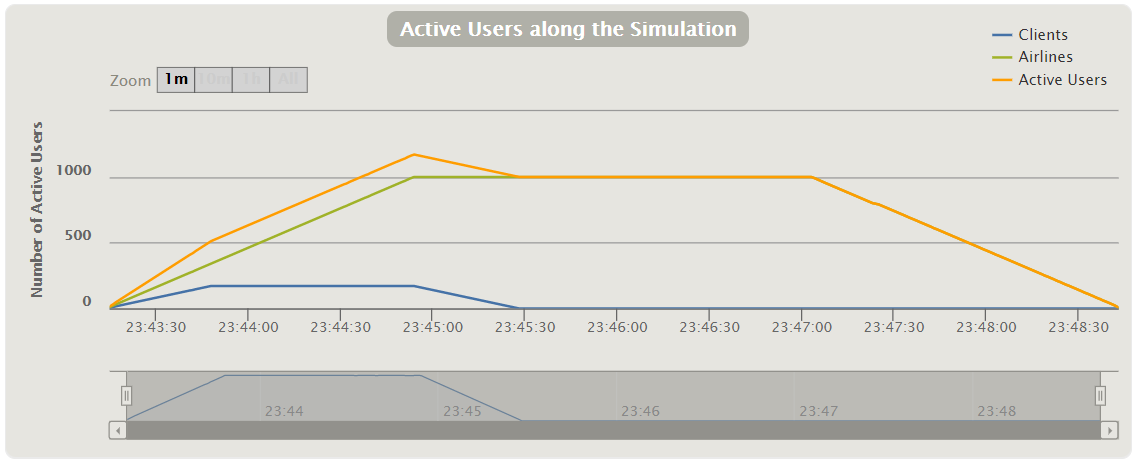
# 

# Planes

Aquí tratamos varias Historias de Usuario ligadas a creación (solo una aerolínea), listar (mis aviones, pertenecientes a una aerolínea), mostrar información y actualizar aviones (la aerolínea que lo haya creado solamente).

Para ello vamos a analizar mediante la herramienta de análisis de rendimiento Gatling, tomando 1000 peticiones de aerolíneas y 500 peticiones de clientes, realizándose primeras las de clientes. Al ser un número bajo de peticiones podemos comprobar que todas se llevan a cabo satisfactoriamente:

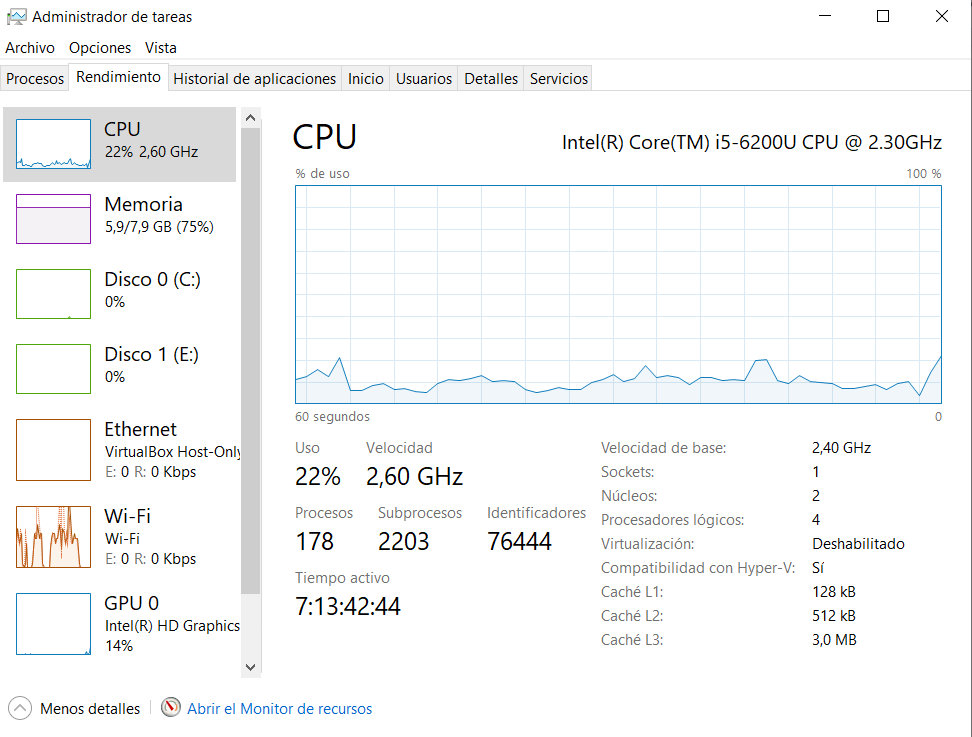




Podemos observar que el tiempo medio no supera los 20 ms, lo cual es un tiempo muy aceptable considerando que estamos realizando 1000 peticiones para las aerolíneas y 500 peticiones para los clientes.

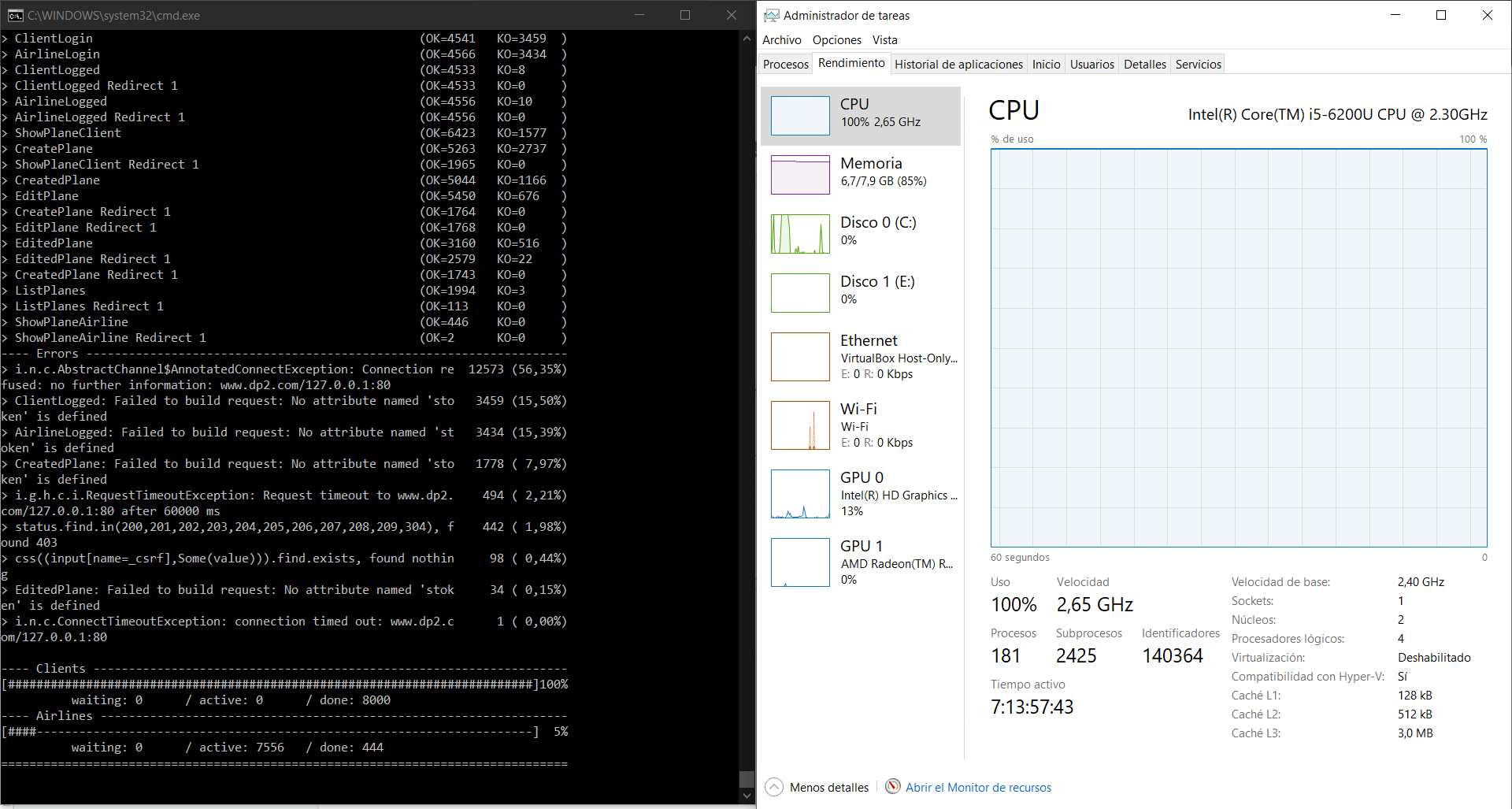
Ahora probemos con un mayor índice de peticiones, en concreto vamos a usar 8000 peticiones (nos aseguramos un número grande que haga que falle el ordenador) tanto de usuarios como de aerolíneas. Las pruebas que vamos a realizar se toman con la base de datos de MySQL activa, realizando verdaderamente el rendimiento que va a suponer la aplicación y la base de datos sobre un sistema portátil. Nos aseguramos antes de correr la prueba de que no existen aplicaciones en segundo plano y que el navegador no posea muchas ventanas abiertas (que consumen mucha RAM, usando un navegador basado en Chromium como es Brave).

Antes de iniciar dicha prueba, comprobamos el rendimiento con la aplicación corriendo es su funcionalidad junto con la base de datos:



Como vemos antes de ejecutar las pruebas la memoria RAM ya se encuentra bastante llena, quizás esto sea a priori el factor que más cause cuello de botella en el equipo.

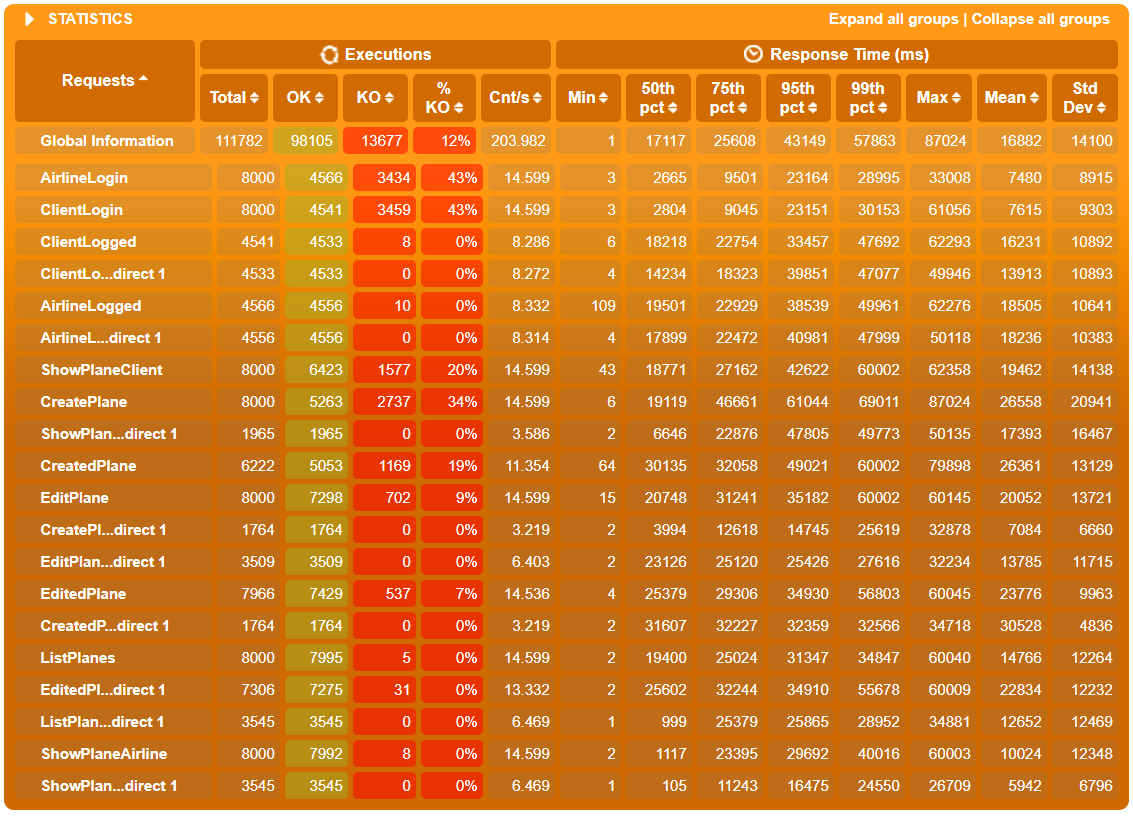
Ahora empezamos la prueba y nos fijamos en el consumo de recursos:



Como podemos observar, lo que causa cuello de botella en este caso es el procesador. Ya que para procesar 8000 llamadas necesitamos un mejor procesador, ya que un i5 6200U es insuficiente para correr con fluidez todas las peticiones.

Ahora veamos las estadísticas que nos proporciona Gatling:





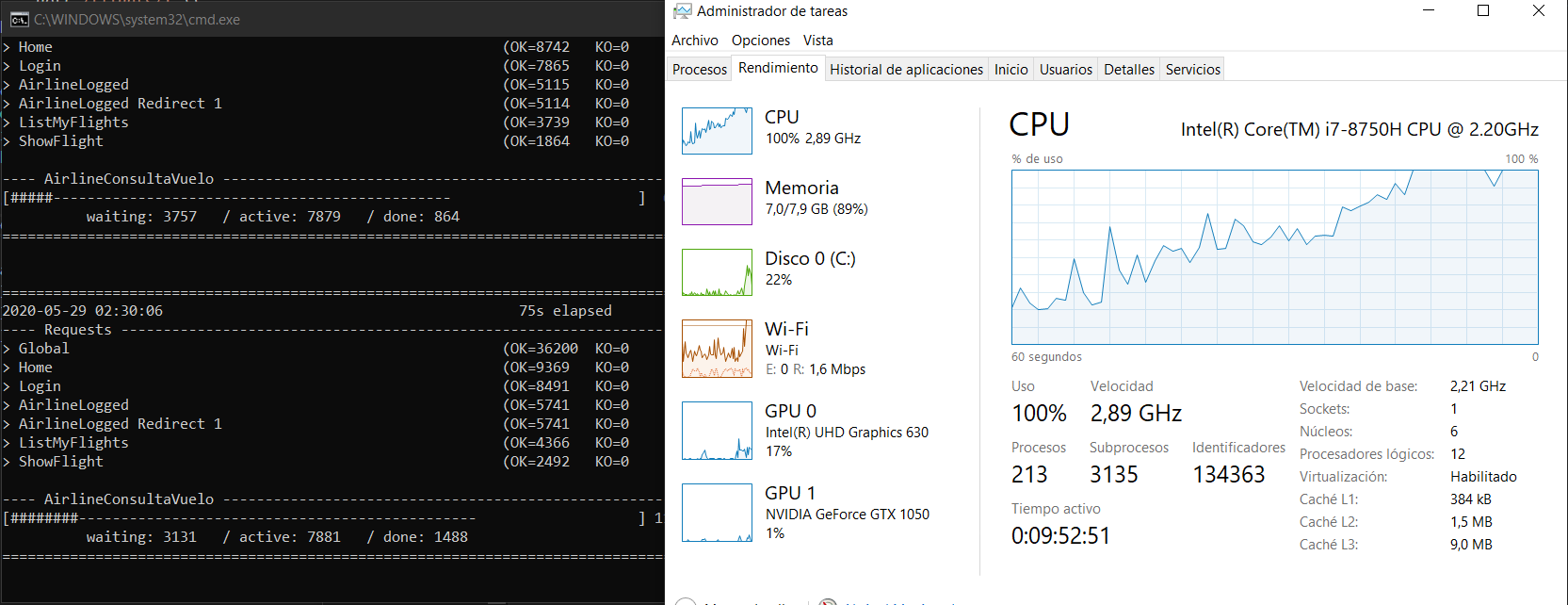
Como podemos observar, no ha dado la suficiente demanda de peticiones, ya que el procesador se ha saturado y no ha llegado a procesarlas todas correctamente. En particular los métodos de Login han sido el mayor punto de saturación (o de peor rendimiento), seguido de los métodos de Creación de Avión y Mostrar Avión.

Además podemos ver que el tiempo medio se ha incrementado en gran medida, debido a que al haber mayor número de peticiones, el procesador tarda más en responderle a cada una de las peticiones, y más teniendo en cuenta que lo estamos llevando a su límite físico.

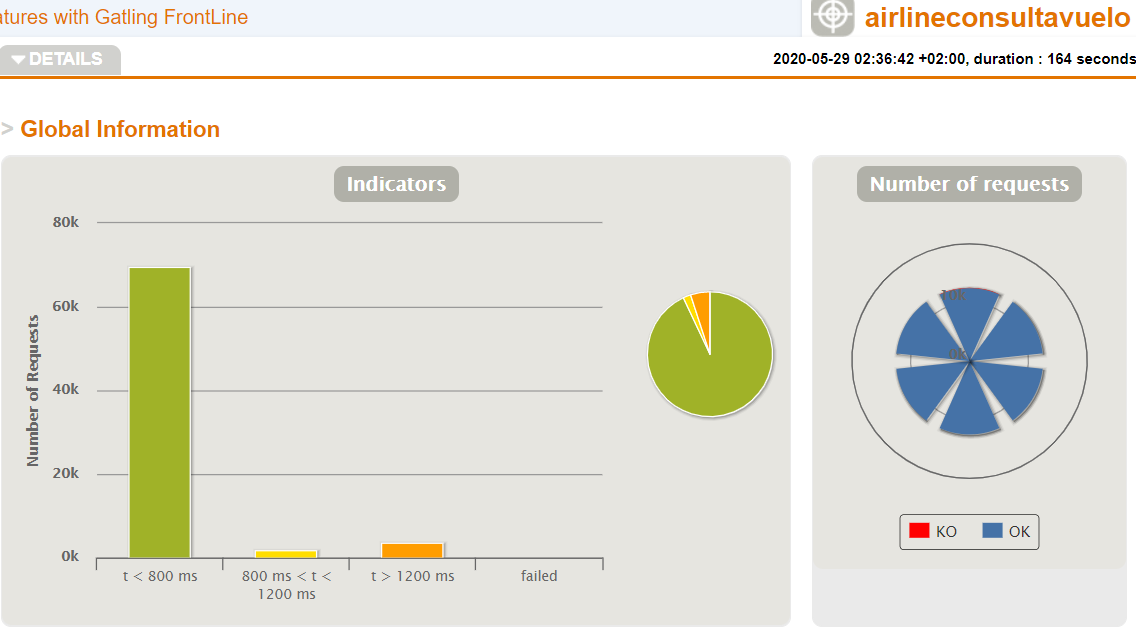
# Flight

## HU T-09: Trabajador consulta un vuelo

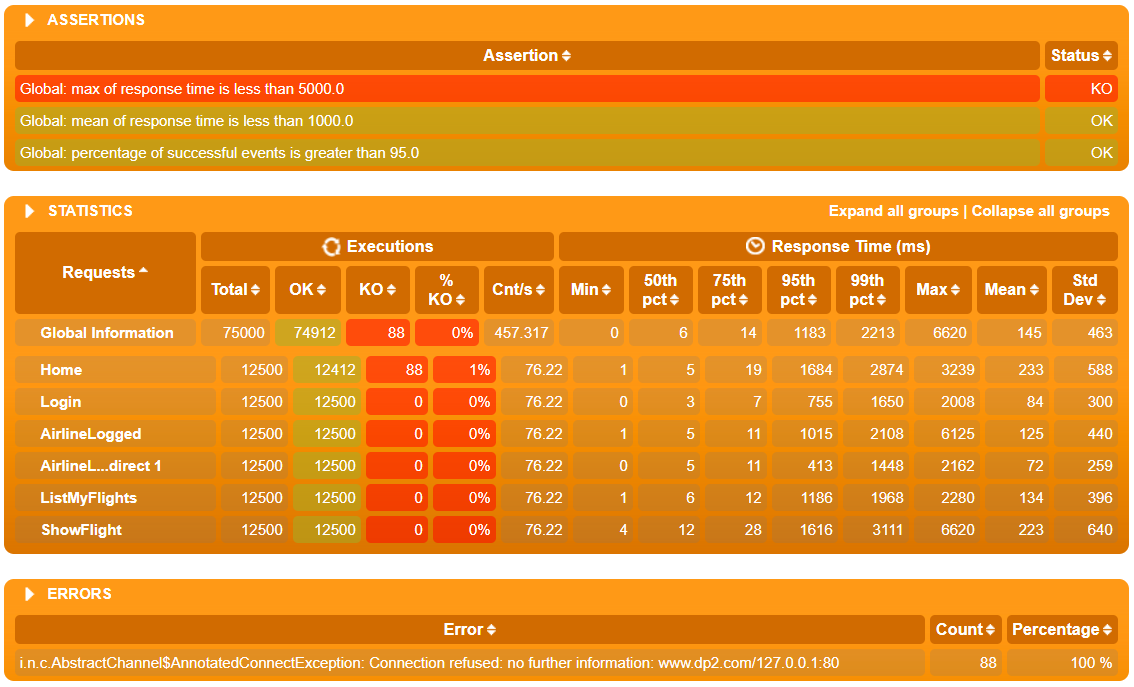
Usando 12500 usuarios



La memoria se ve afectada, aunque el principal cuello de botella es la CPU.



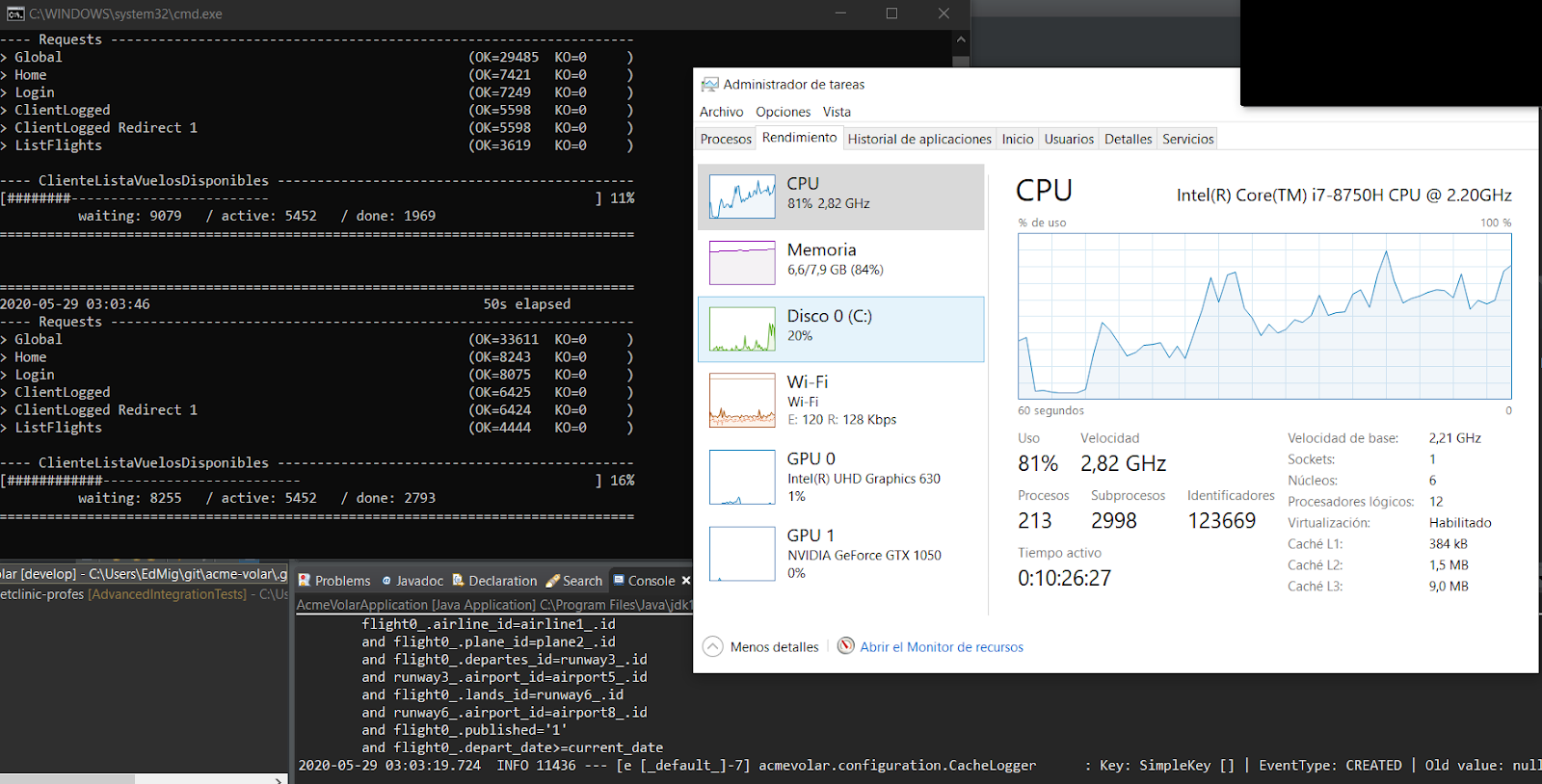
Airport



## HU C-04: Cliente lista vuelos disponibles

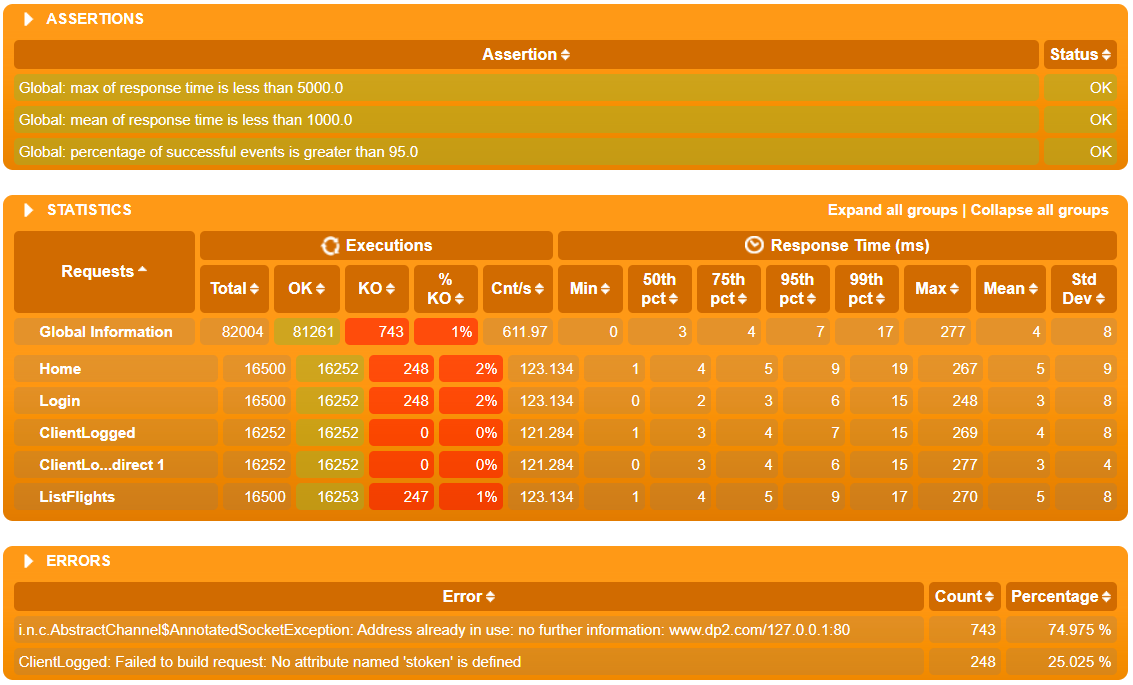
Si se añaden más de 16500 usuarios comienza a fallar.

Al haber realizado profiling para la caché, los resultados están muy optimizados.



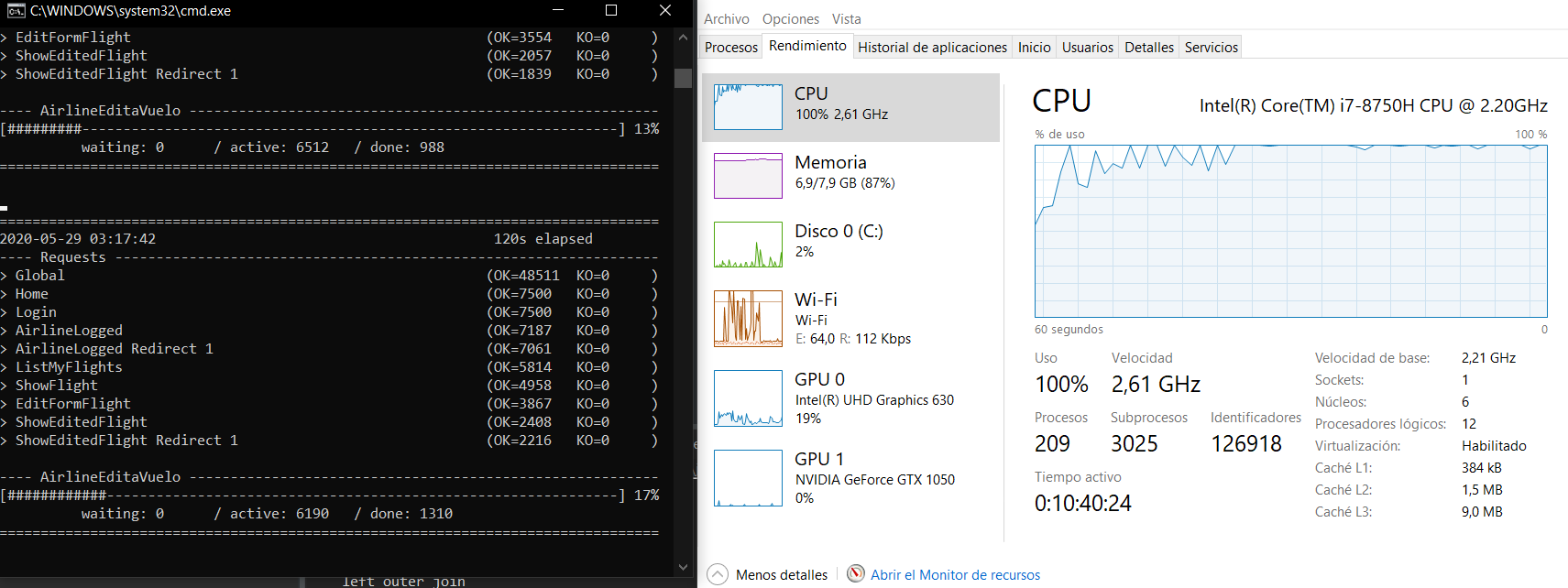
El cuello de botella en este caso sería la memoria.



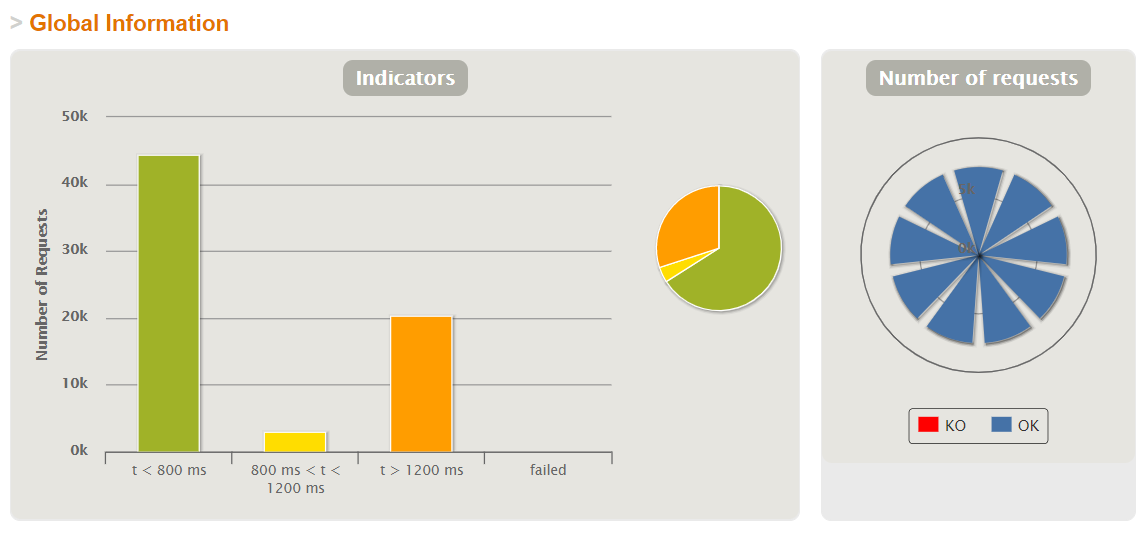


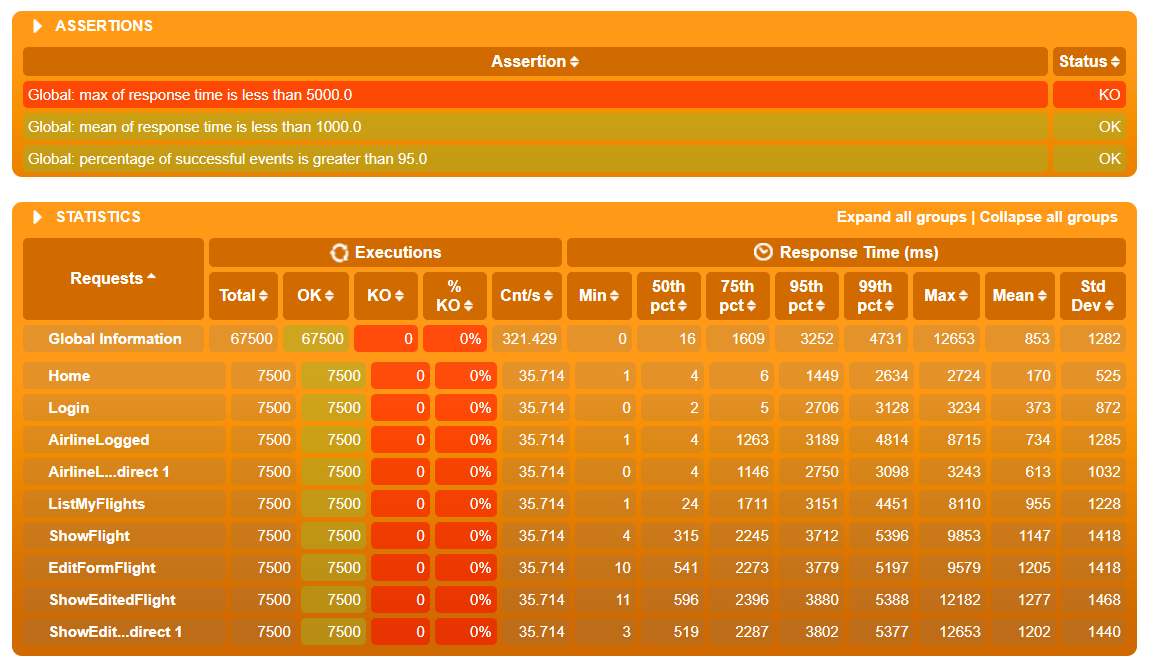
## HU T-07: Trabajador edita un vuelo

En esta prueba usaremos 7500 usuarios.



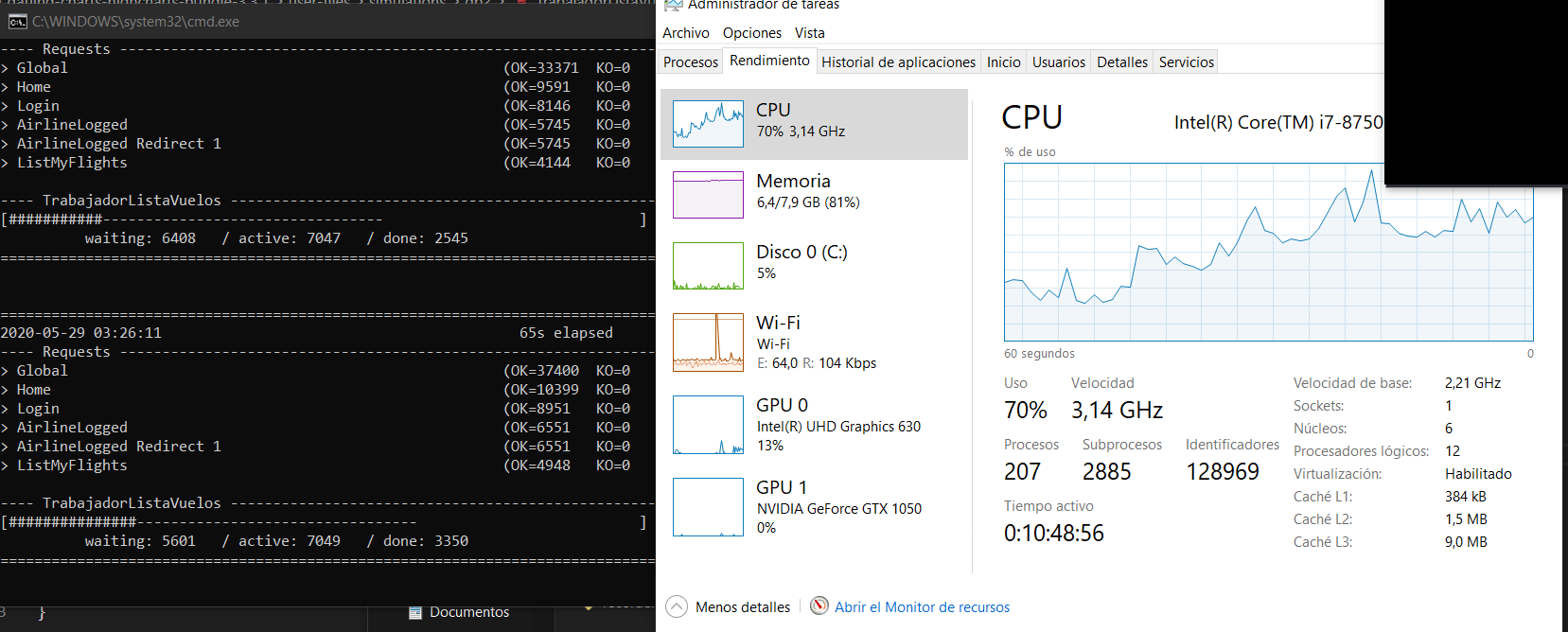
Podemos asociar el cuello de botella a la CPU, aunque la memoria se ve afectada en gran medida.

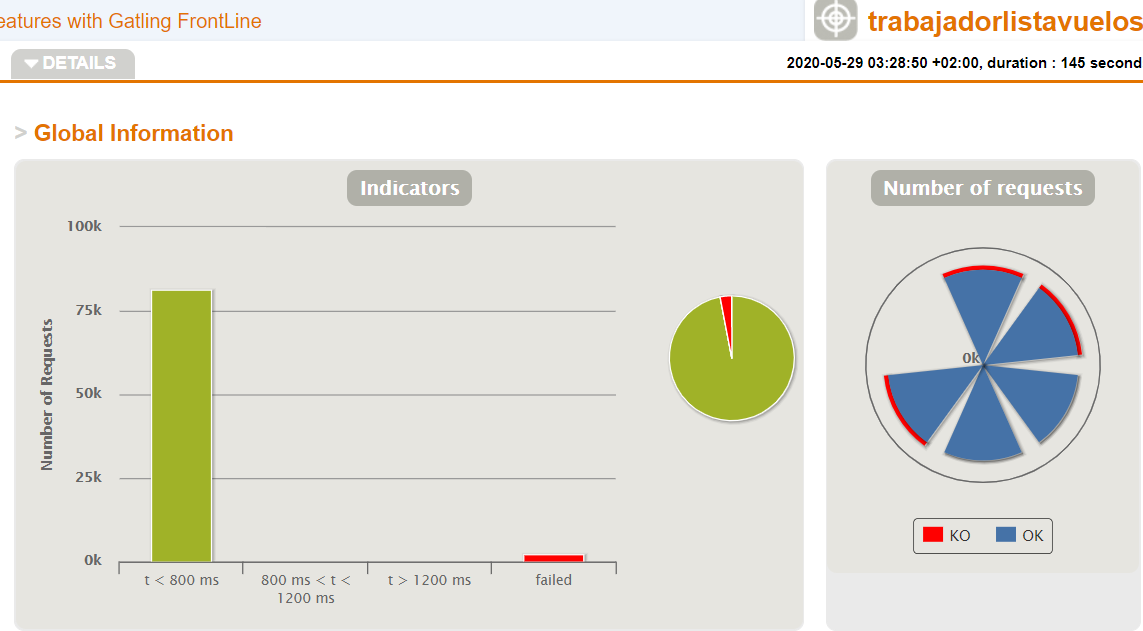


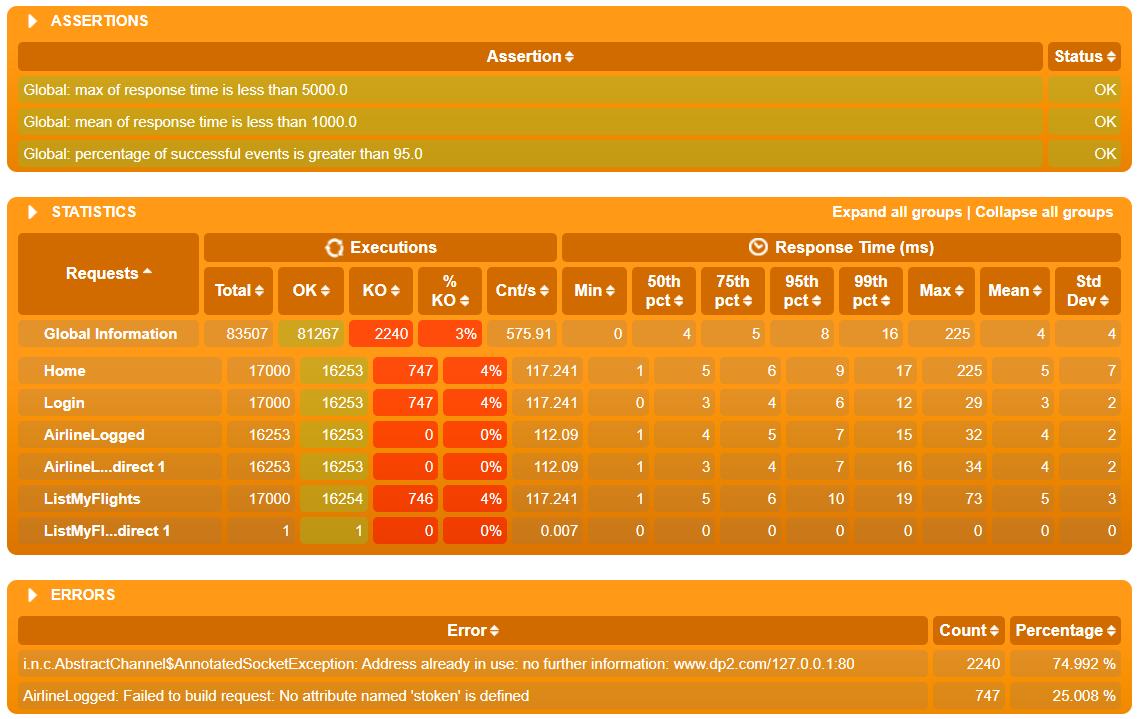


## HU T-08: Trabajador lista vuelos

Se van a usar 17000 usuarios, con el profiling de caché actuando.

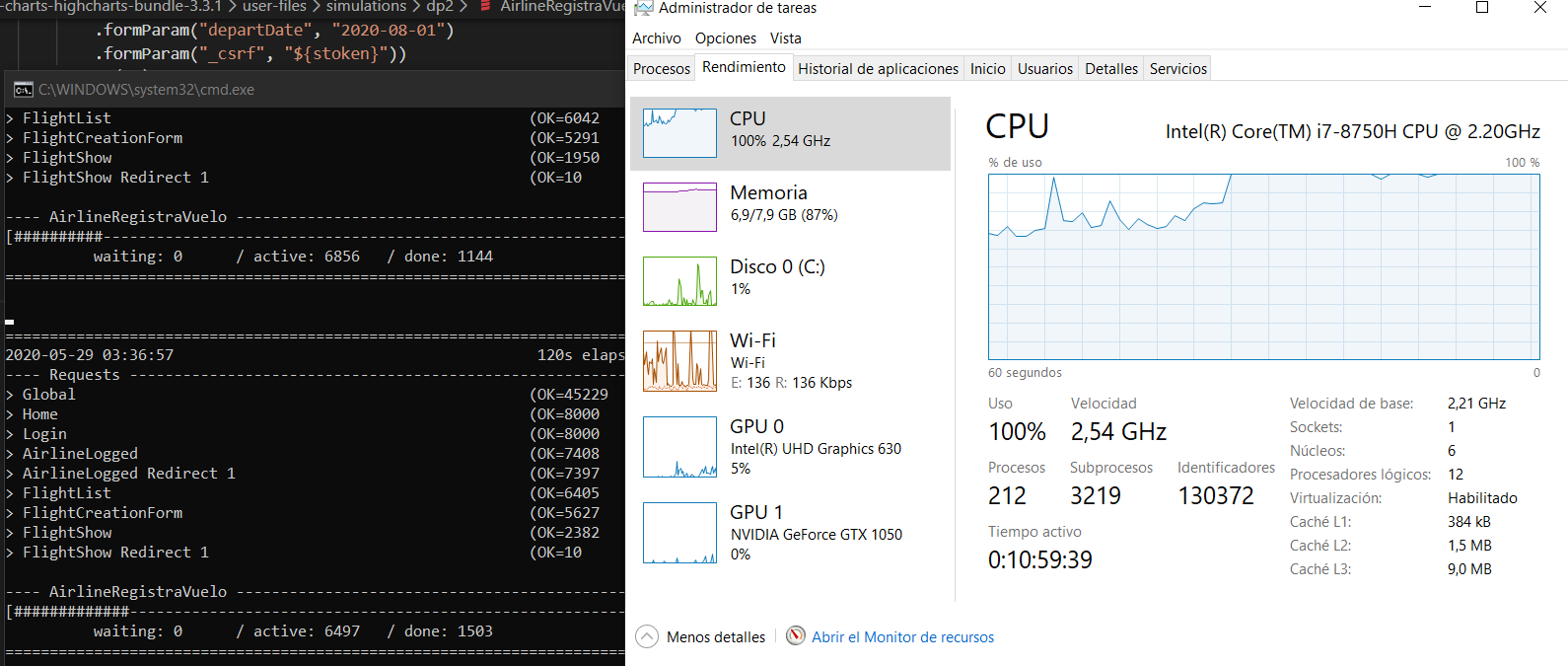


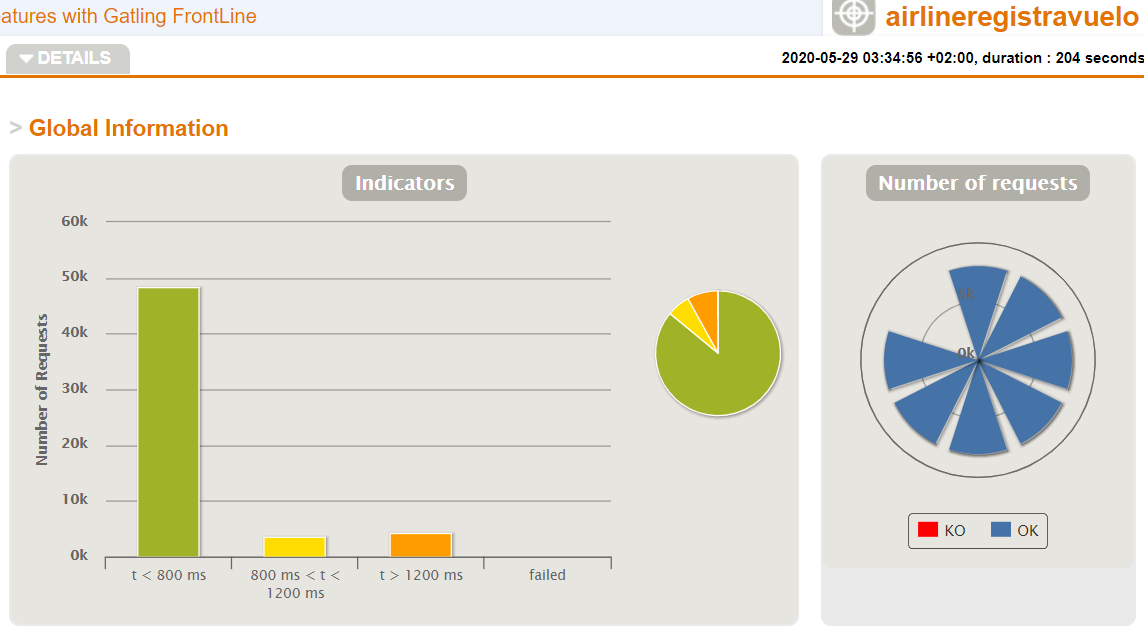


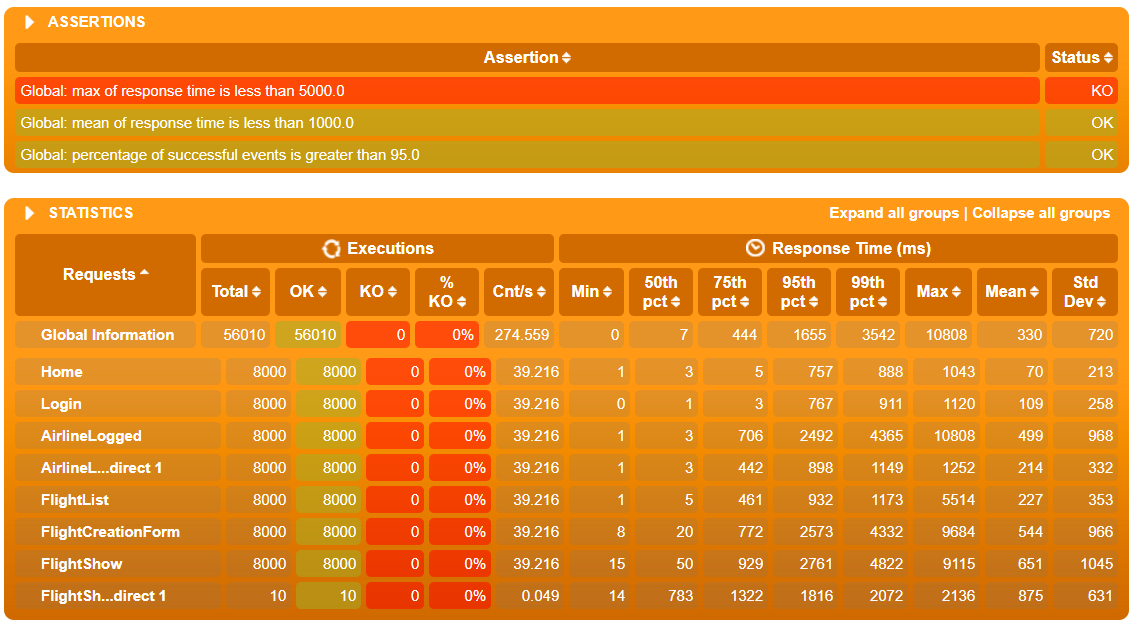


## HU T-06: Trabajador registra un vuelo

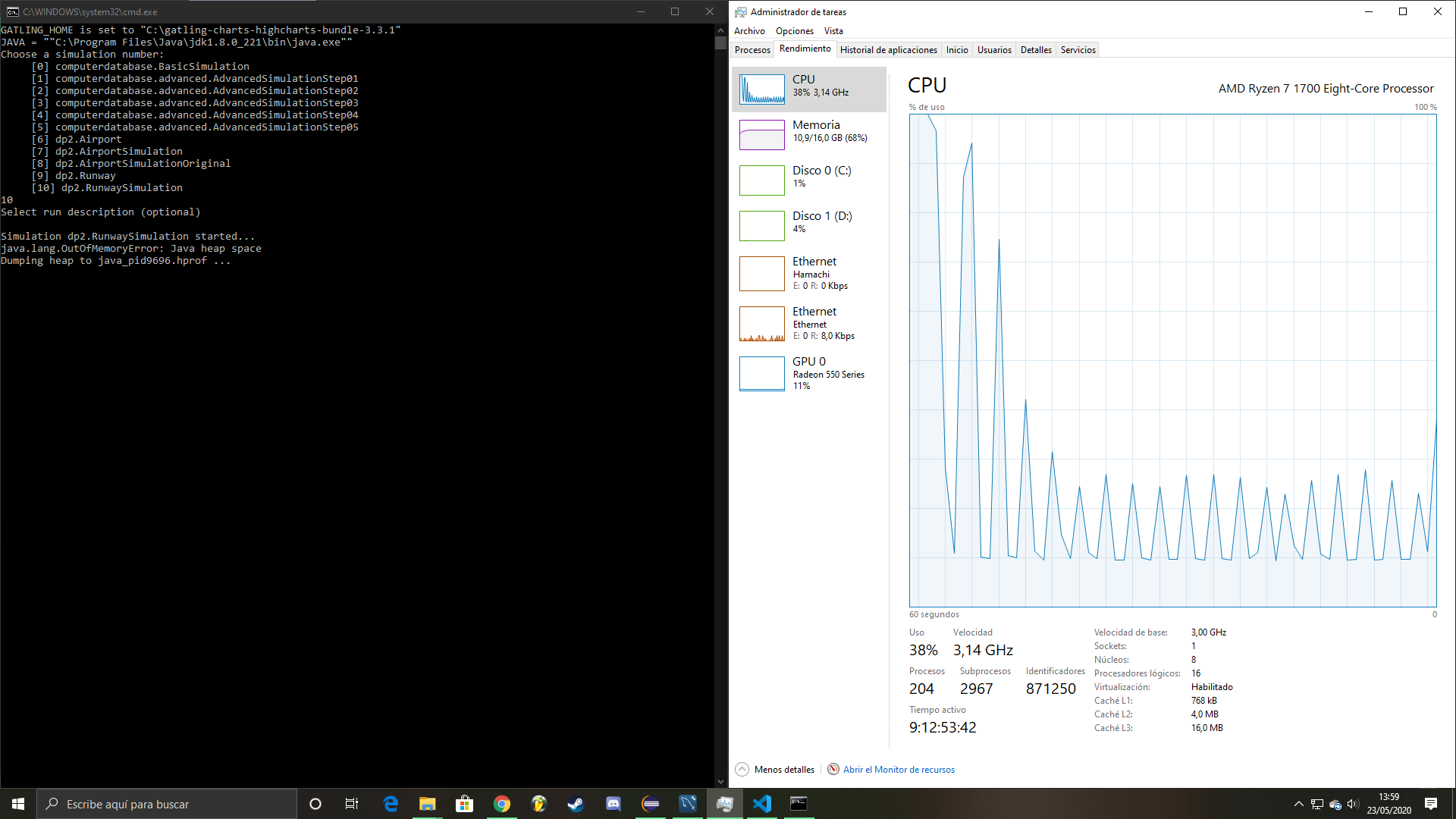
Van a intervenir 8000 usuarios

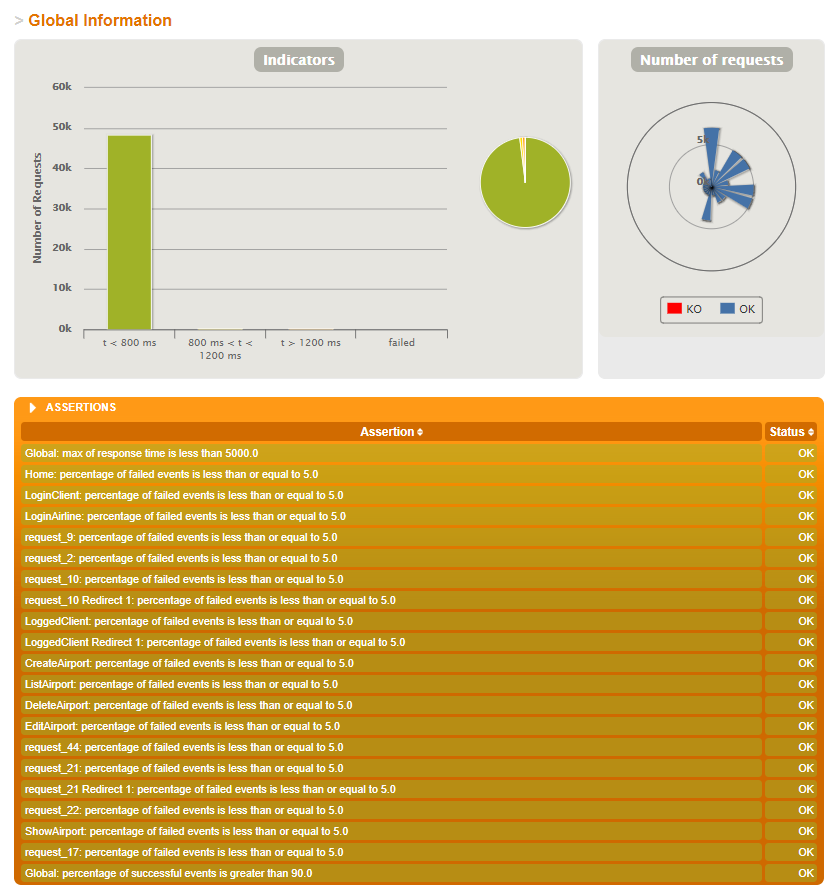


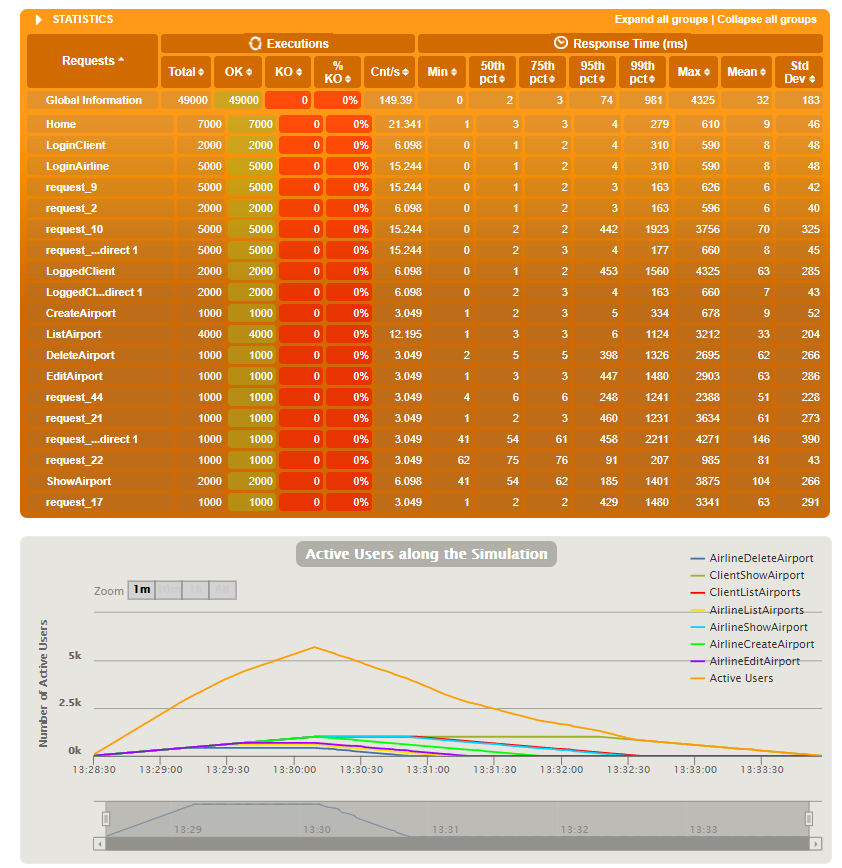




# Airport

Para los informes de rendimiento de aeropuerto las hemos realizado en un tiempo de 10 segundos con 1000, 2000, 50000 y 20000 usuarios para las historias de usuario correspondientes a airport de mostrado y listado para el caso de un cliente, y en el caso de trabajador de aerolínea la creación, borrado y edición de este sumado a las dos anteriores que también se dan para este rol de usuario. Como podemos ver la mayoría de historias de usuario están por debajo de los 800 ms de respuesta, el cuello de botella se produciría en la memoria del ordenador que, como podemos ver, está en su máxima capacidad.





# Runway

Hemos realizado pruebas de rendimiento para las historias de usuario correspondientes a runaway, en este caso el creado, editado, borrado y listado de ésta por parte de un trabajador de aerolínea, estas pruebas las hemos realizado para 2000, 5000 y 20000 usuarios en una duración de 10 segundos para cada pruebas, aquí vemos una captura de pantalla para el último caso, como podemos ver el cuello de botella se produce como en el caso anterior, el cuello de botella se produce en la memoria del computador que como podemos ver es la parte que más está siendo consumida.