|  |  |
| --- | --- |
| PET-CLINIC (Rendimiento) | GRUPO 13  Jesús Andrés Rico Catalán  Carlos Albalat Heredia |

Índice

[Introducción 2](#_Toc41821095)

# Introducción

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





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

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

 setUp(

  transportarNeg.inject(rampUsers(280) during (100 seconds)),

  transportarPos.inject(rampUsers(280) during (100 seconds))

  ).protocols(httpProtocol)

    .assertions(

        global.responseTime.max.lt(5000),

        global.responseTime.mean.lt(1000),

        global.successfulRequests.percent.gt(95)

     )

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

# Rendimiento del sistema

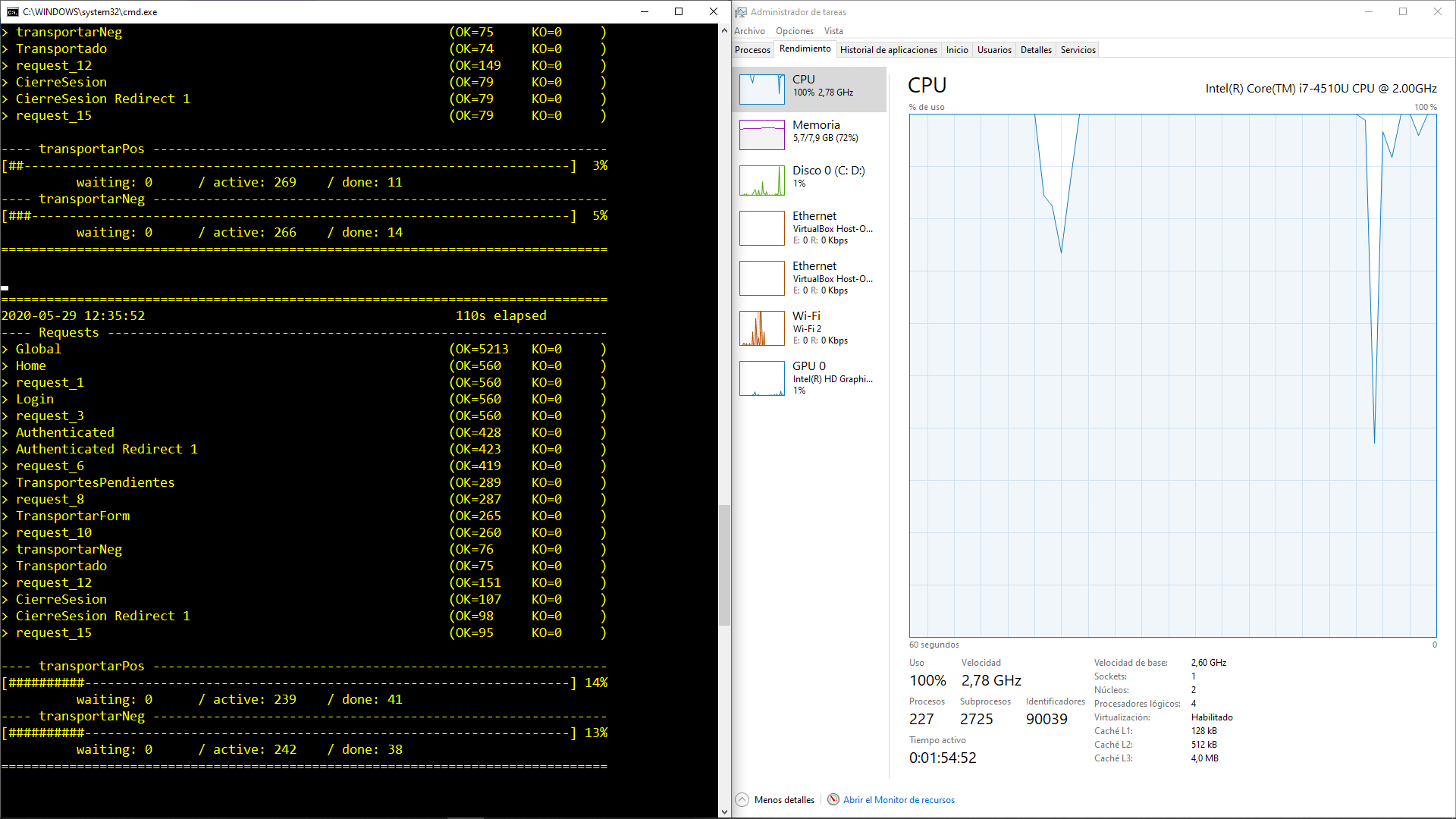
Algunos test de rendimiento incluyen varias HUs en el mismo test debido a la naturaleza de esta. Debido a que cada compañero posee un equipo con distintas especificaciones varía considerablemente el rendimiento de nuestra aplicación.

## HU-1,3,5: Diagnosis

El siguiente test engloba estas HUs:

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

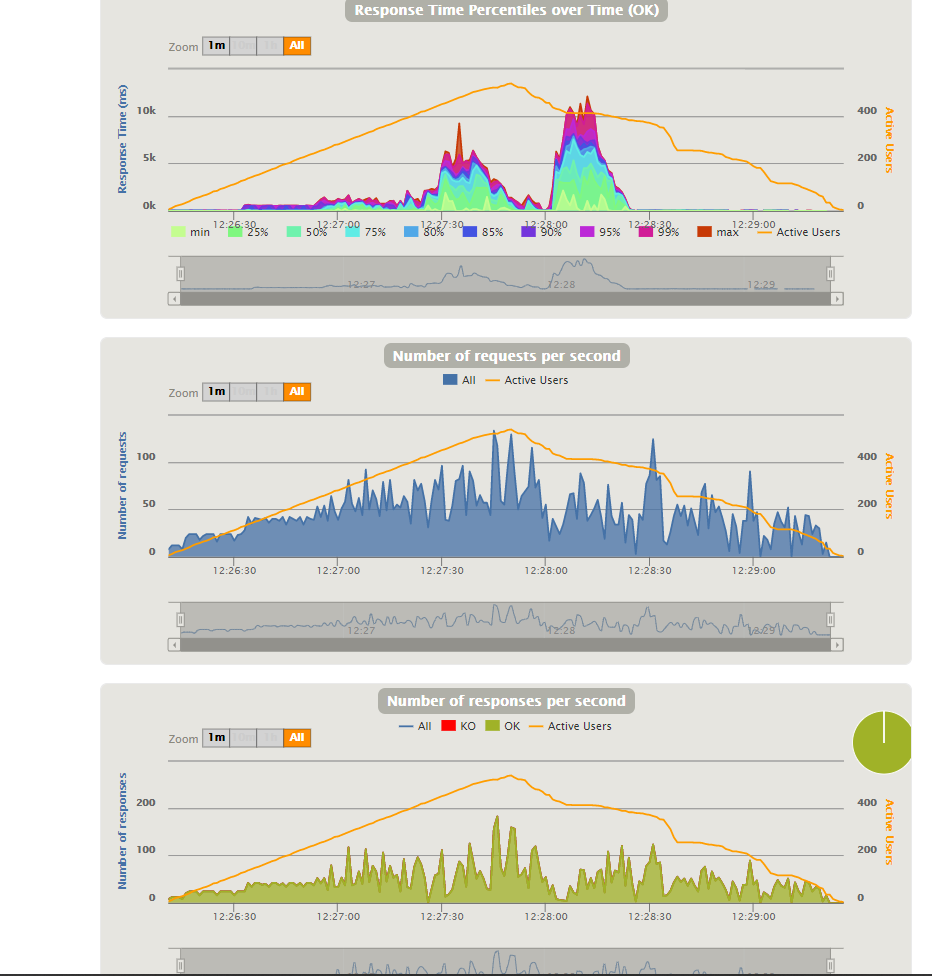
El máximo número de usuarios concurrentes es 560. Un aumento de 20 usuarios concurrente provoca una gran saturación provocando que el sistema para estas historias sea inviable usar la aplicación. La causa de que nuestro servidor deje de responder de manera óptima es debido a que la CPU no es capaz de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



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

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





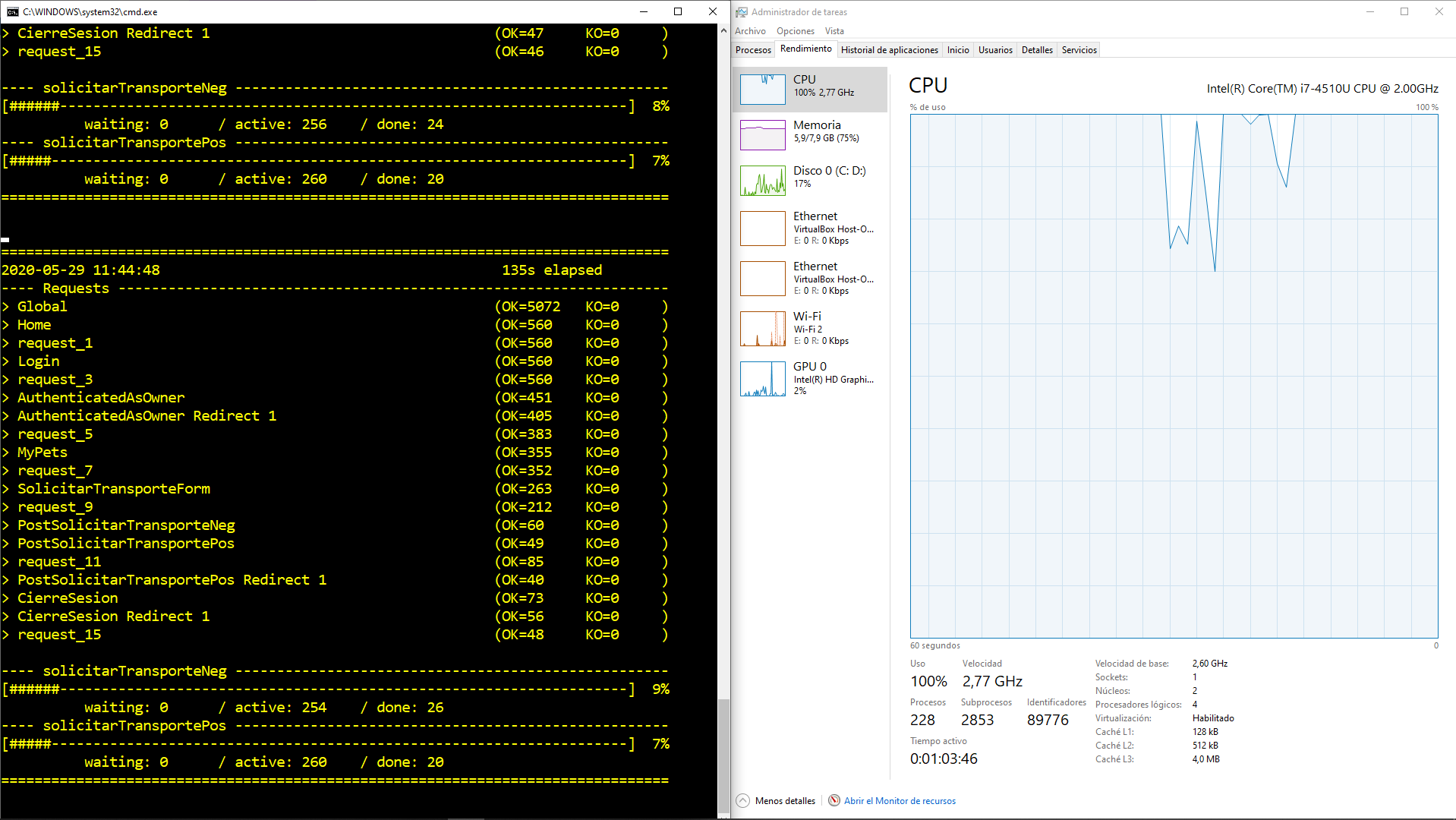
En el report generado por Gatling podemos comprobar que el sistema satisface como mínimo el 95% de peticiones http que genera el test. También se verifica que la media de las peticiones es menor a un segundo aunque se pueden observar que al someter a cierto estrés hay una serie de peticiones que necesitan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

## HU2: Diagnosis

El siguiente test engloba esta HU:

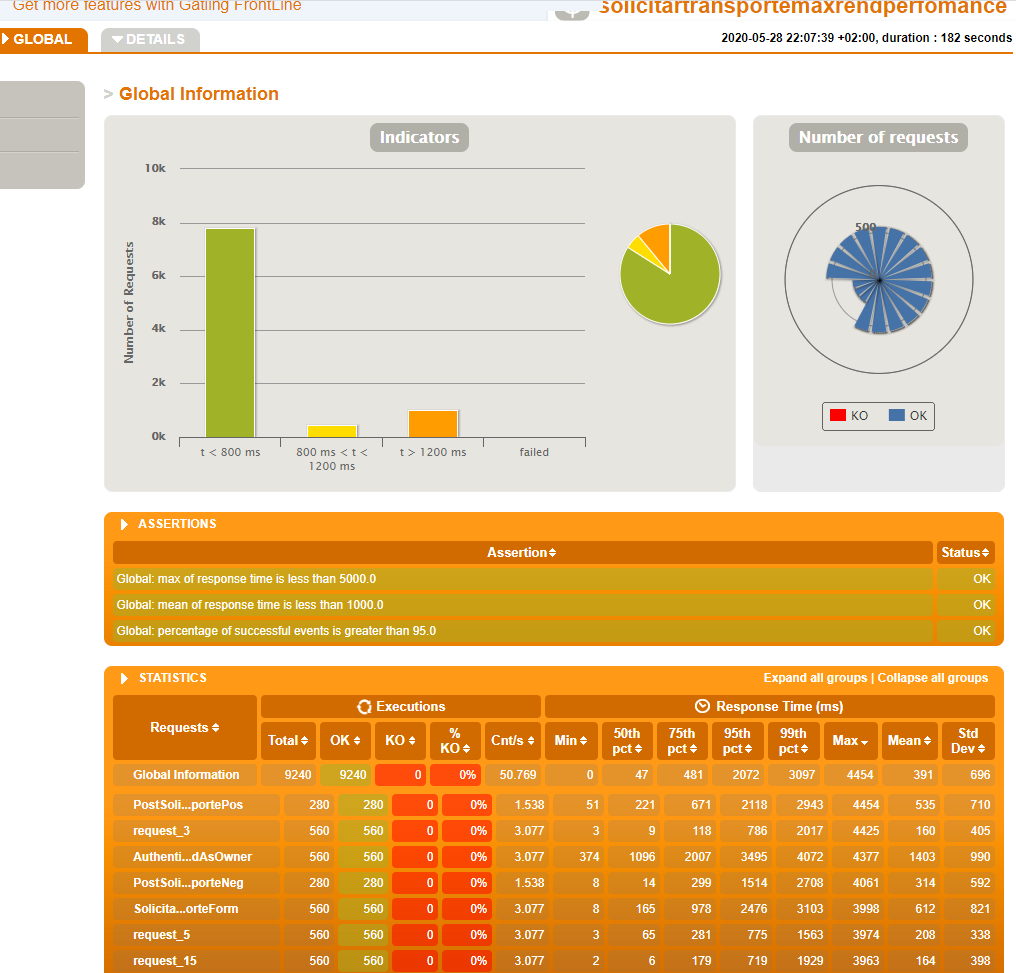
* El propietario puede solicitar transportar a su animal.

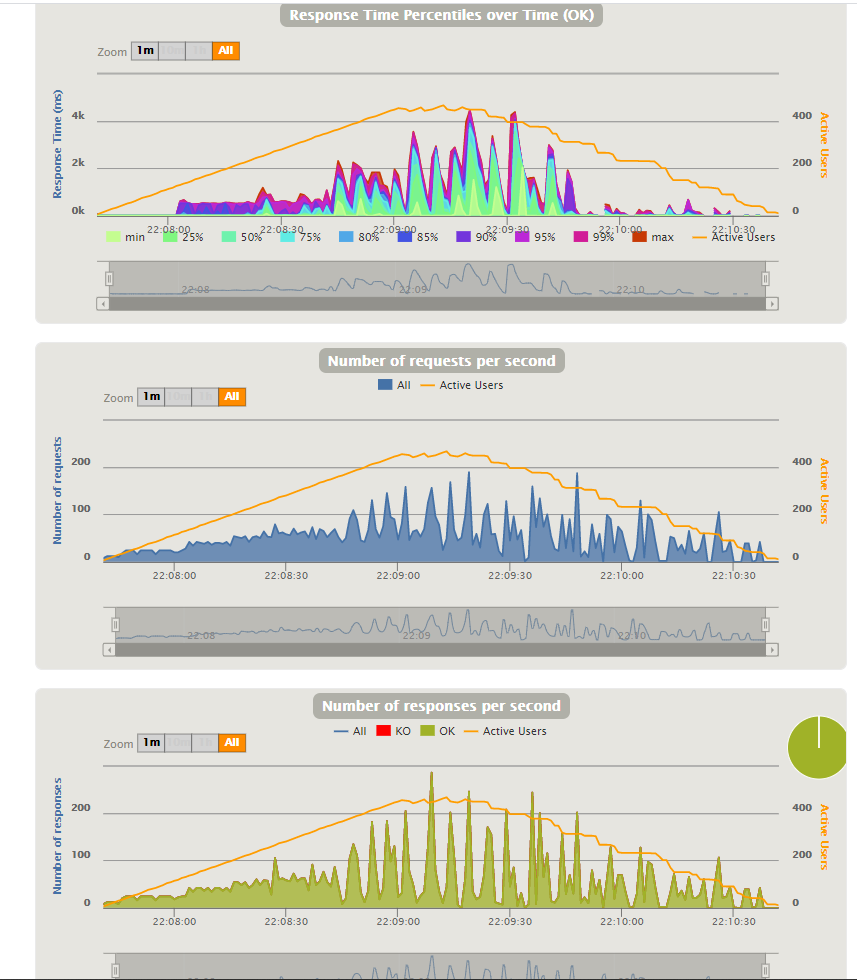
El máximo número de usuarios concurrentes es 560. Un aumento de 35 usuarios concurrente provoca una gran saturación provocando que el sistema para esta historia sea inviable usar la aplicación. La causa de que nuestro servidor deje de responder de manera óptima es debido a que la CPU no es capaz de procesar tantas peticiones concurrentemente propiciando cuellos de botellas y por lo tanto muchos TimeOuts como se puede ver en la siguiente captura:

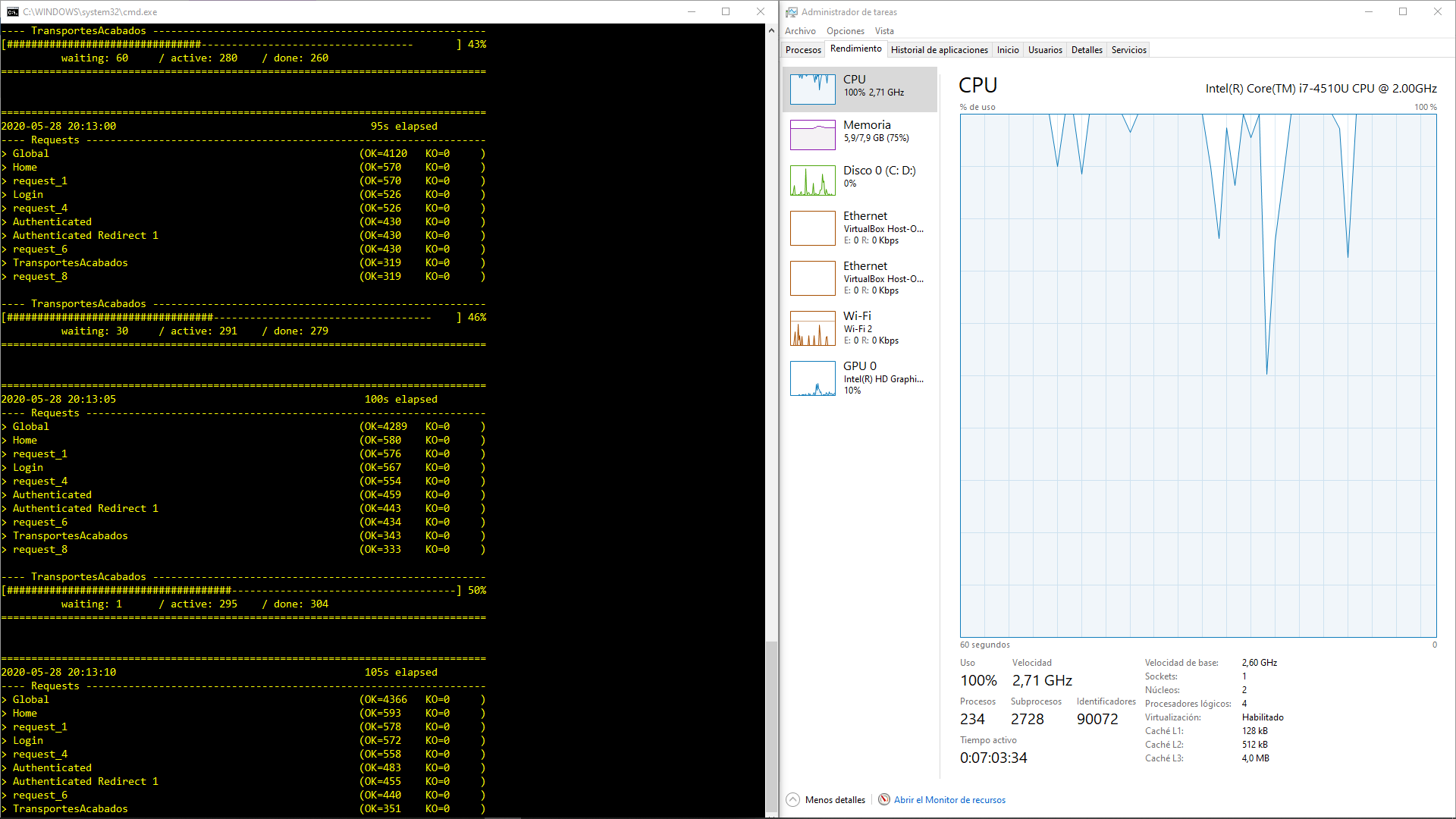


\*En este test se enfrenta un escenario positivo de transportar y otro negativo.

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





En el report generado por Gatling podemos comprobar que el sistema satisface como mínimo el 95% de peticiones http que genera el test. También se verifica que la media de las peticiones es menor a un segundo y que ninguna petición supera un tiempo de respuesta mayor a 5 segundos. Por lo tanto, se podría decir que el comportamiento es idóneo ya que apenas se producen cuellos de botella.

## HU3:

## HU4:

## HU5:

## HU6:

## HU7:

## HU8:

## HU9:

## HU10:

## HU11:

## HU12:

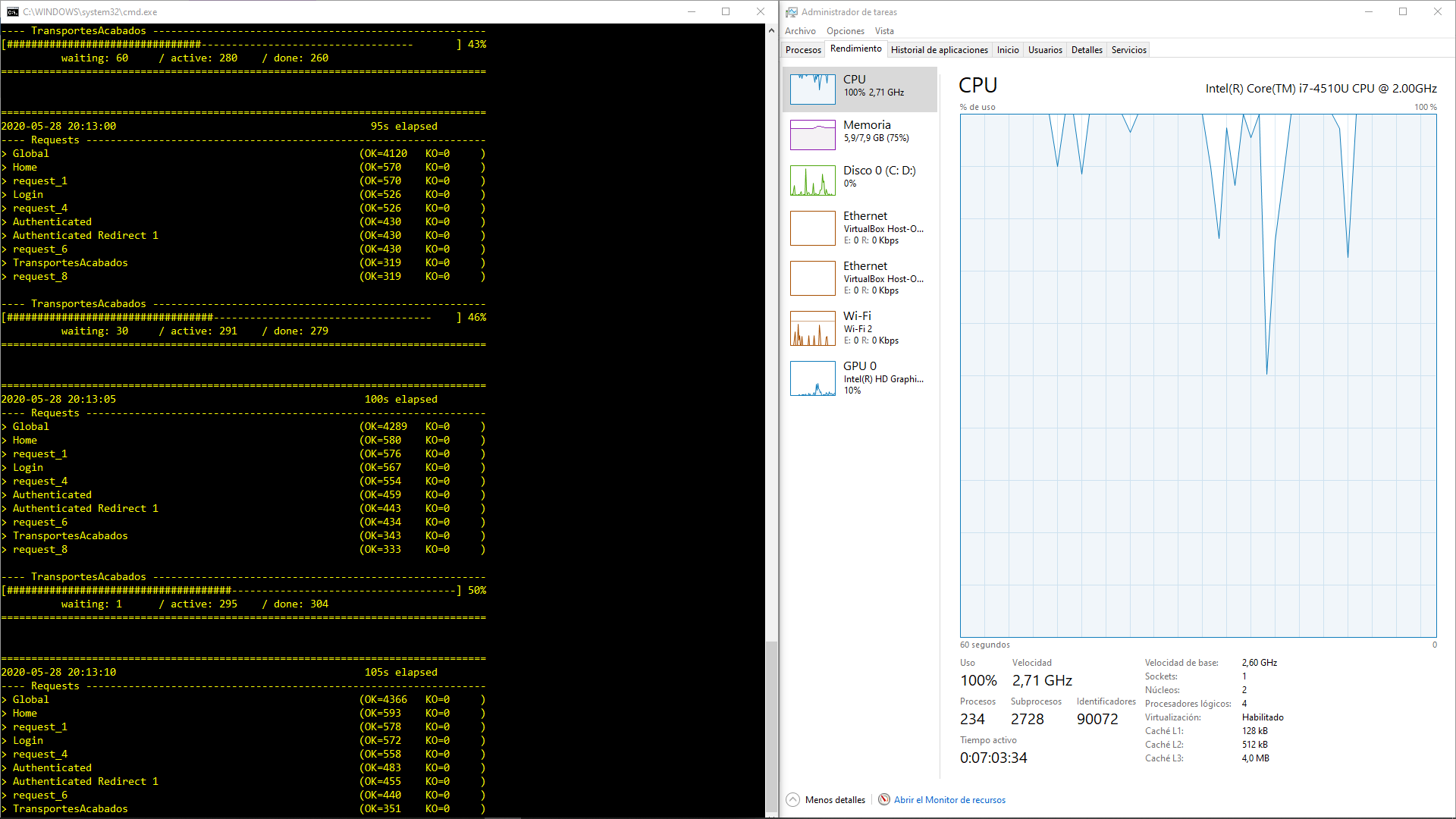
## HU13:

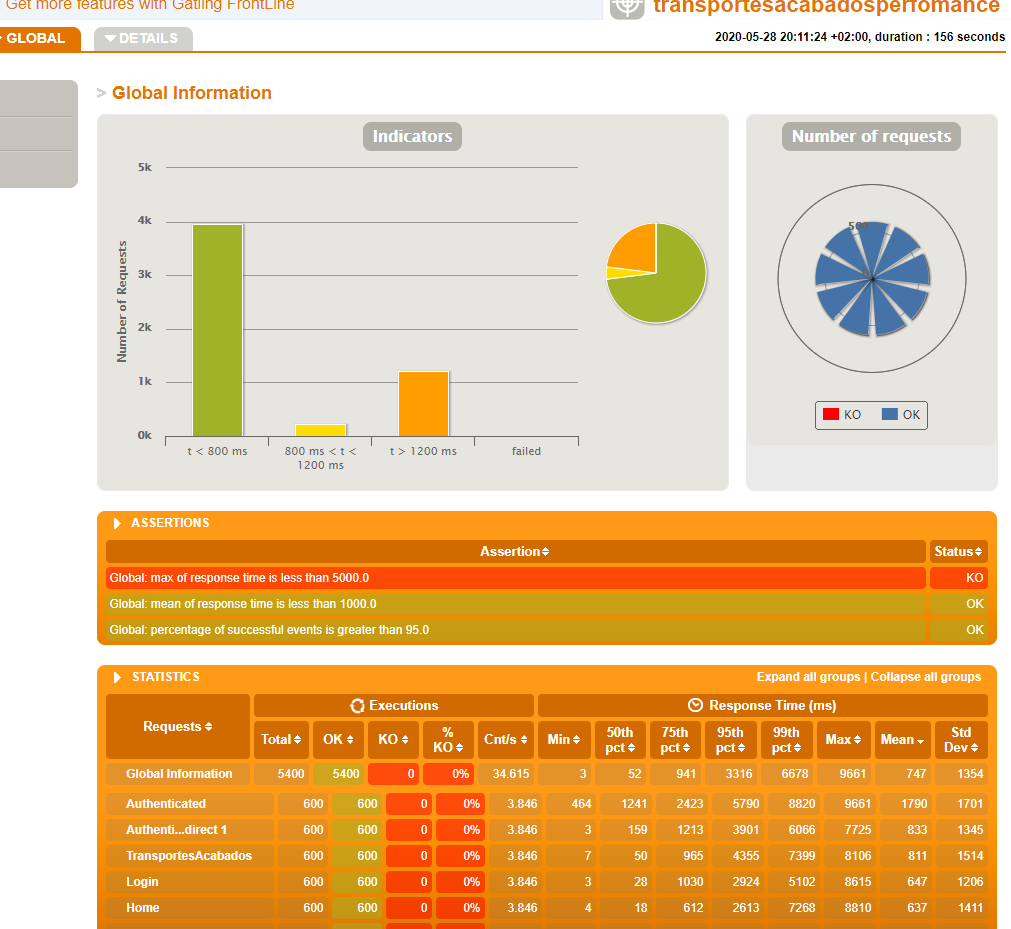
## HU14:

El siguiente test engloba esta HU:

* El propietario/secretario debe poder ver todos los transportes acabados

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





# 

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

# Conclusión

TODO PONER CUAL ES EL RENDIMIENTO DEL SISTEMA MAXIMO SIN PETAR QUE ES EL MIN USERS SOPORTADOS

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