**PERFORMANCE REPORT**

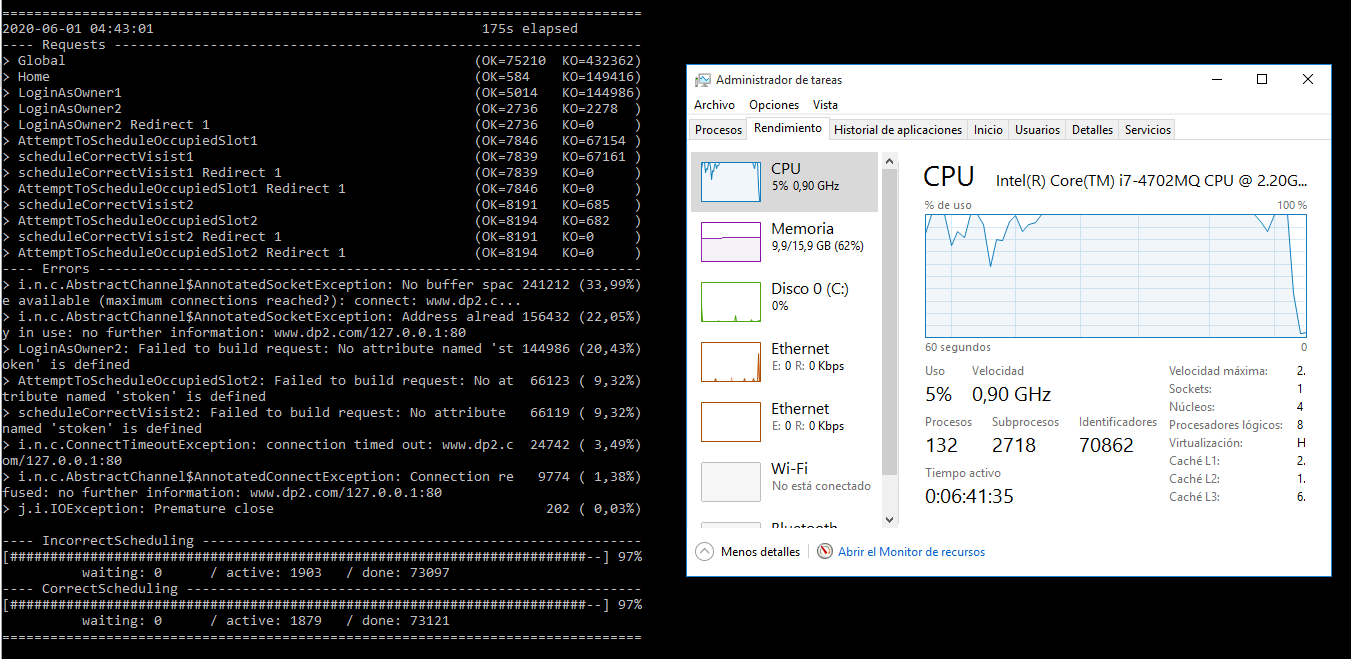
**[US1] Schedule an appointment online**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **150k**

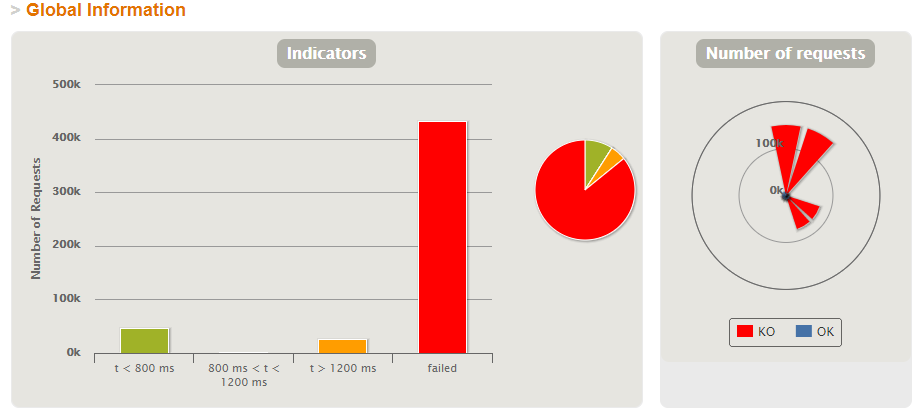
Evidences:

When performing a stress test with 150.000 (75.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

Capture of the performance monitor of our computer and the test execution console:



Capture of the Gatling report:

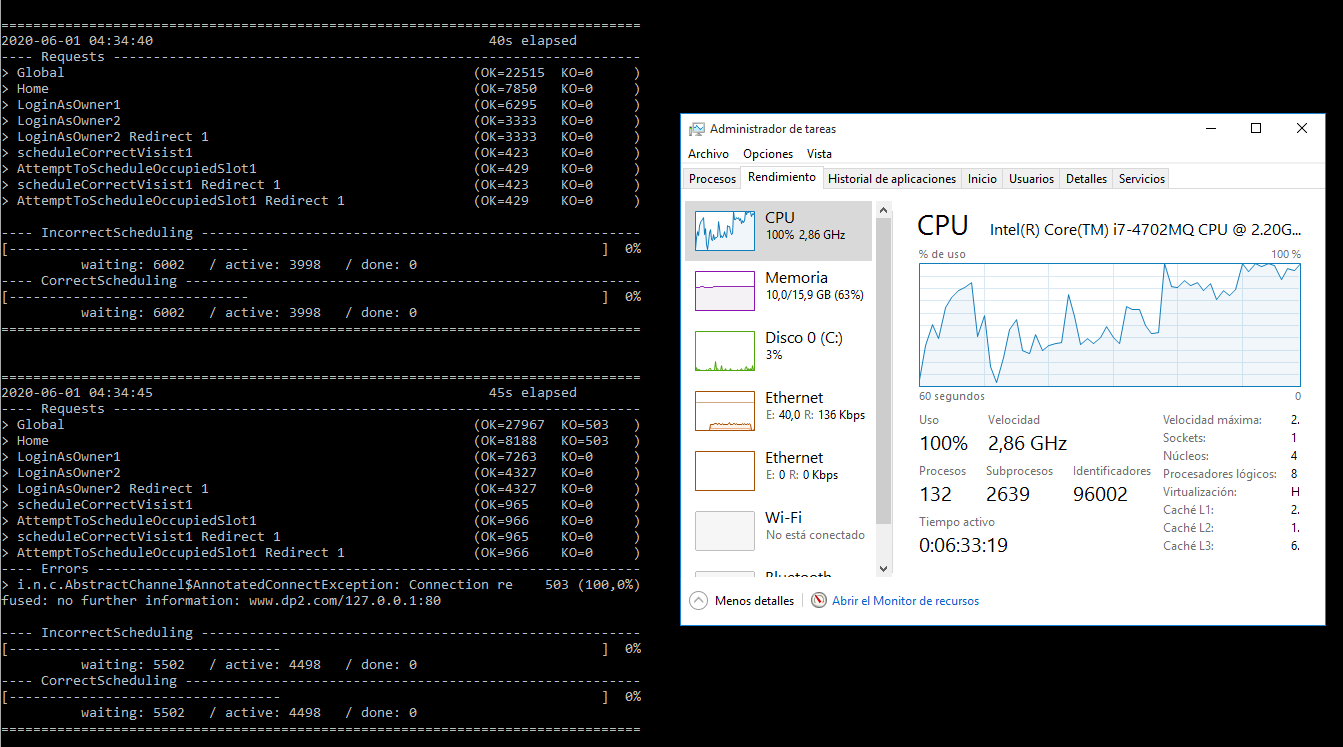


1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **8800**

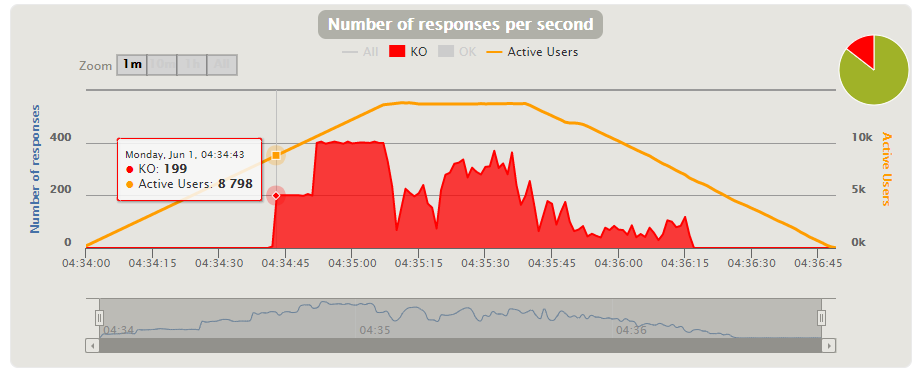
Evidences:

Performing a load test with 20.000 (10.000 users for each scenario) concurrent users, when the number of active users is 8.800 we can see that the mistakes made and failed requests begin. We can also see the bottleneck in the CPU.

Capture of the performance monitor of our computer and test execution console:



Capture of Gatling report:



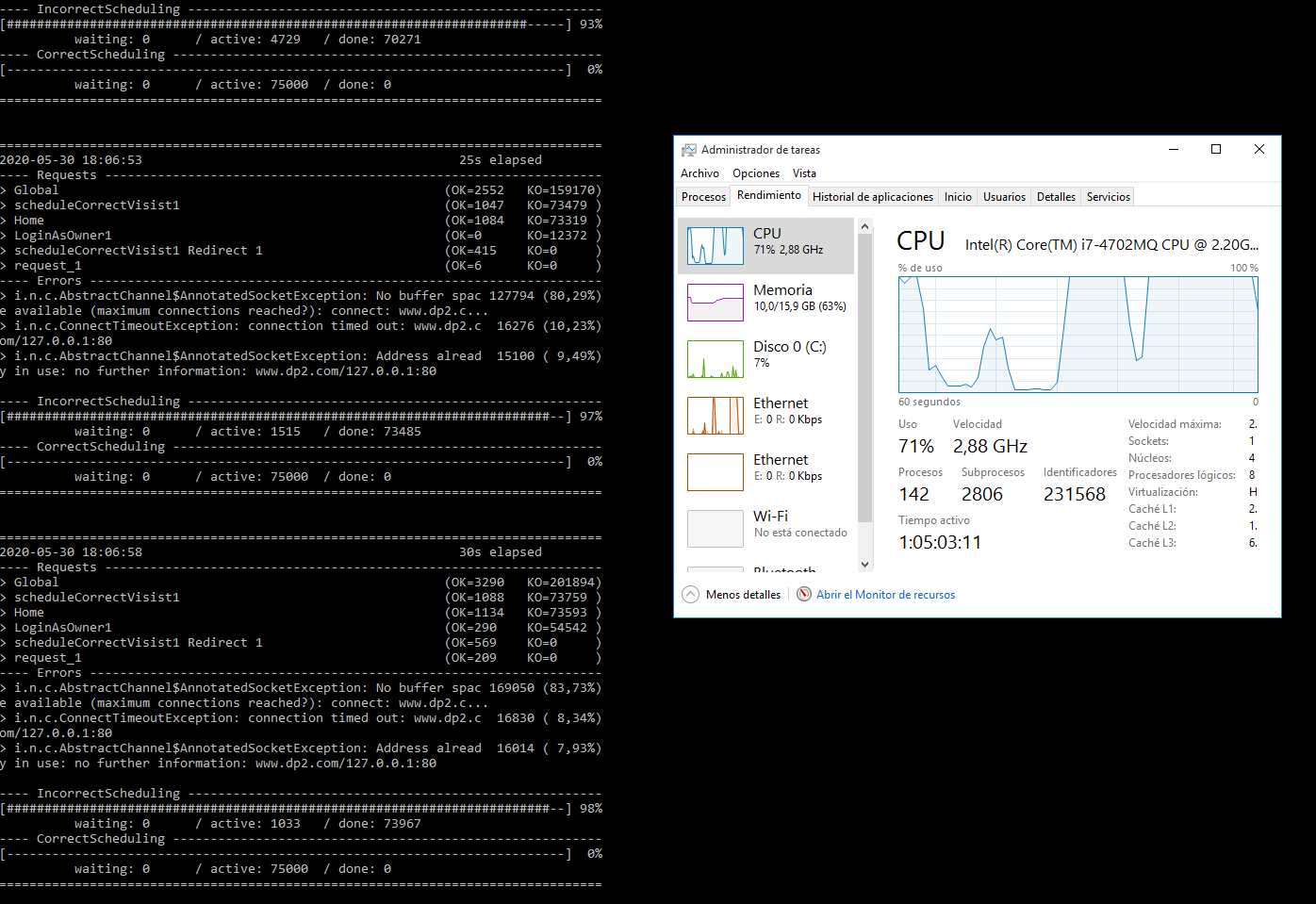
**[US2] Request a visit with a specific veterinarian**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **120k**

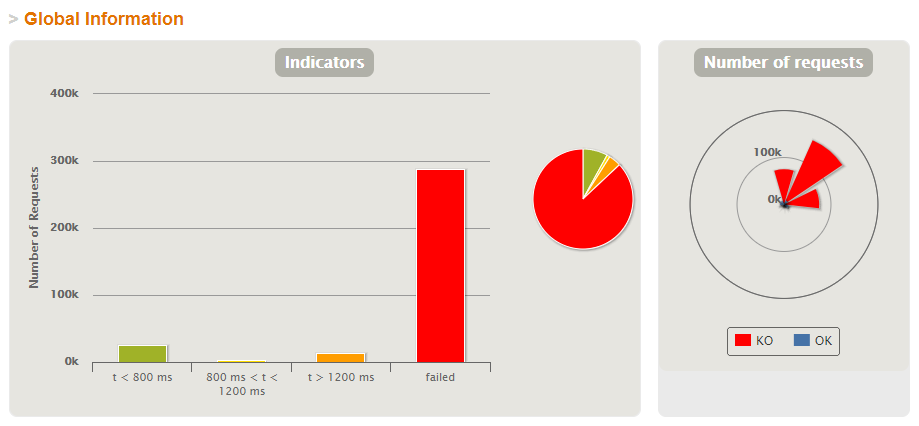
Evidences:

When performing a stress test with 120.000 (60.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

Capture of the performance monitor of our computer and the test execution console:



Capture of Gatling report:

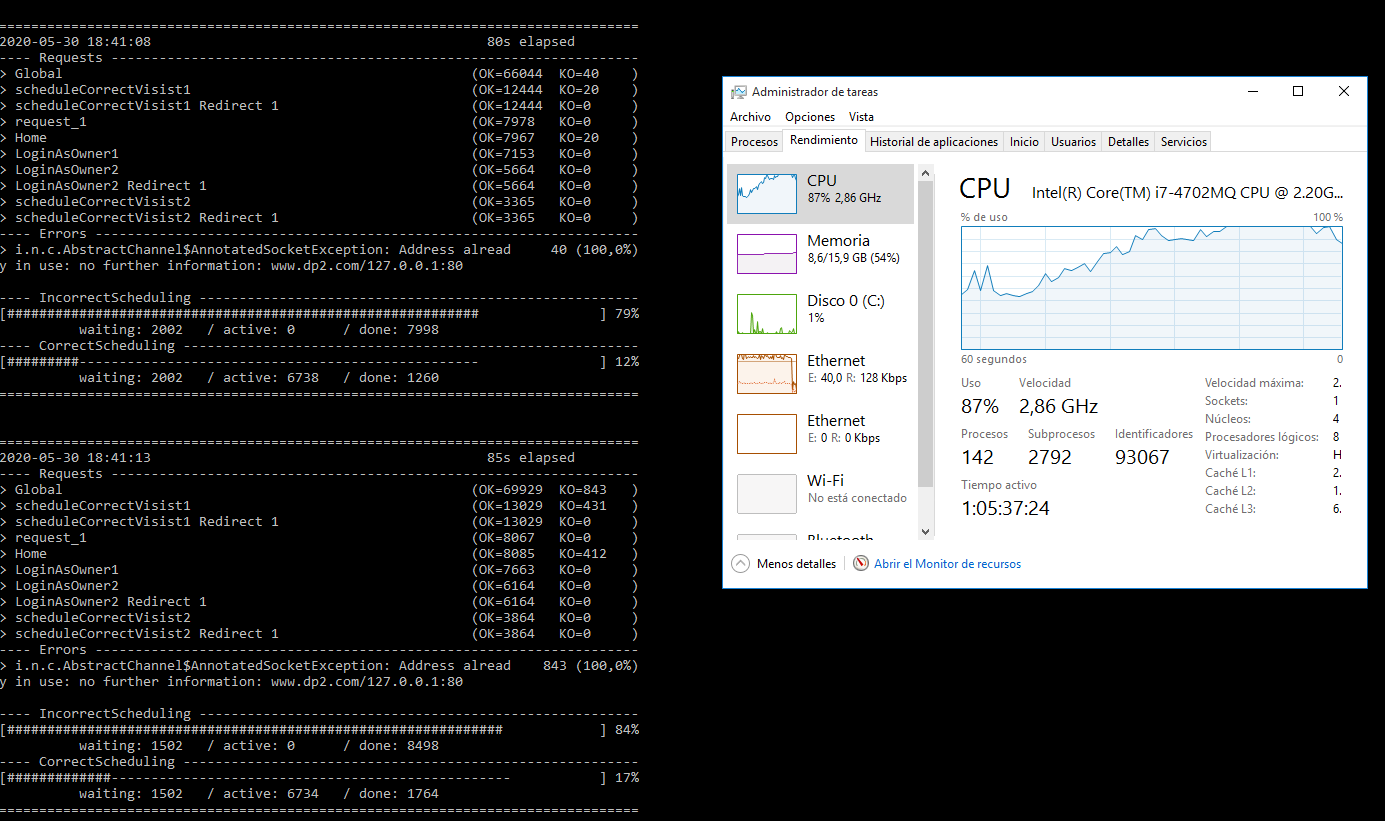


1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **6.900**

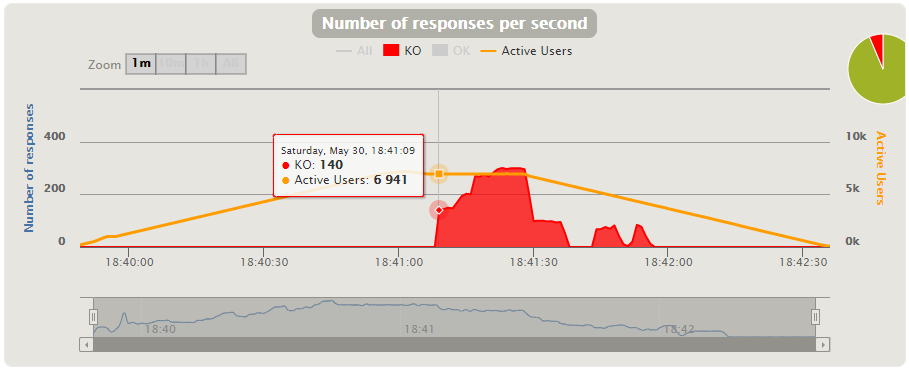
Evidences:

Performing a load test with 20.000 (10.000 users for each scenario) concurrent users, when the number of active users is 6.900 we can see that the mistakes made and failed requests begin. We can also see the bottleneck in the CPU.

Capture of the performance monitor of our computer and test execution console:



Capture of Gatling report:



**[US3] Manage appointments automatically**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **150k**

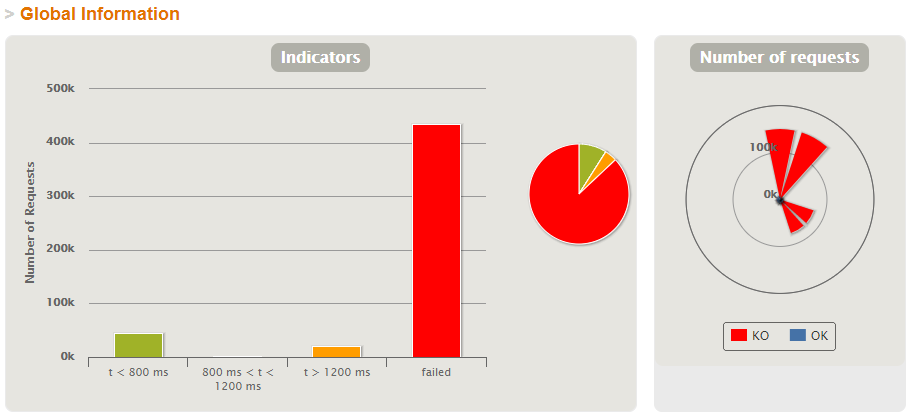
Evidences:

When performing a stress test with 150.000 (75.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

Capture of the performance monitor of our computer and the test execution console:



Capture of Gatling report:

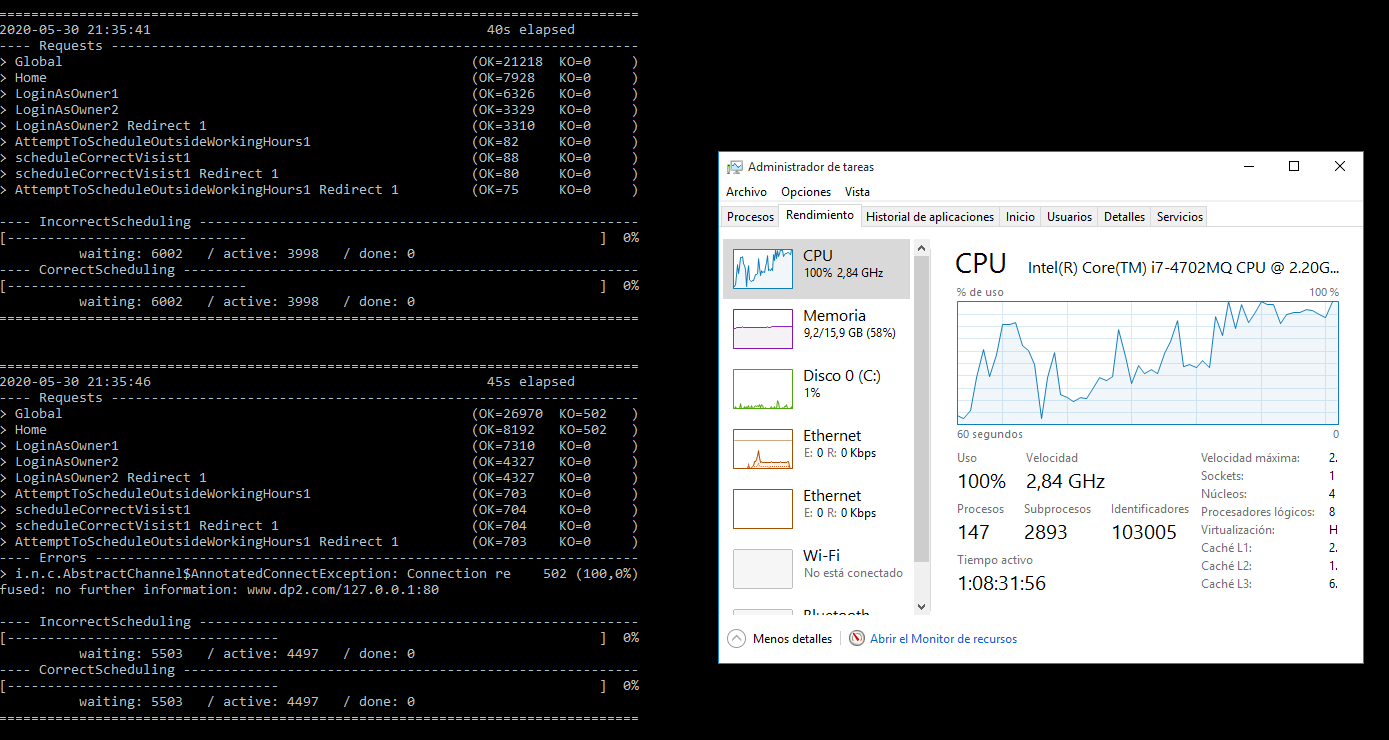


1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **8.800**

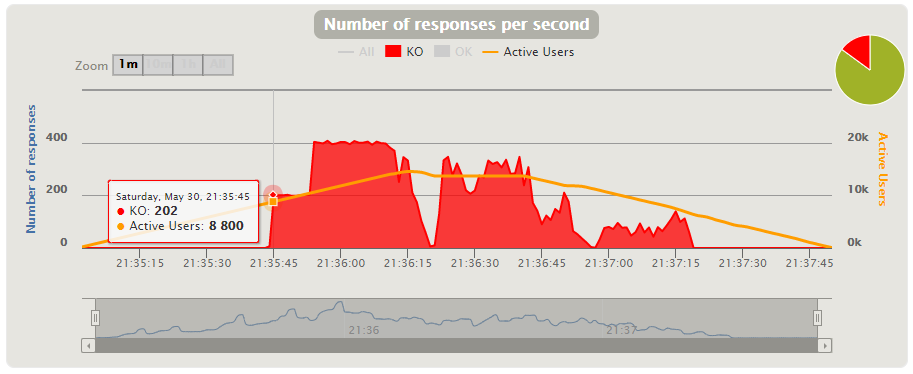
Evidences:

Performing a load test with 20.000 (10.000 users for each scenario) concurrent users, when the number of active users is 8.800 we can see that the mistakes made and failed requests begin. We can also see the bottleneck in the CPU.

Capture of the performance monitor of our computer and test execution console:



Capture of Gatling report:



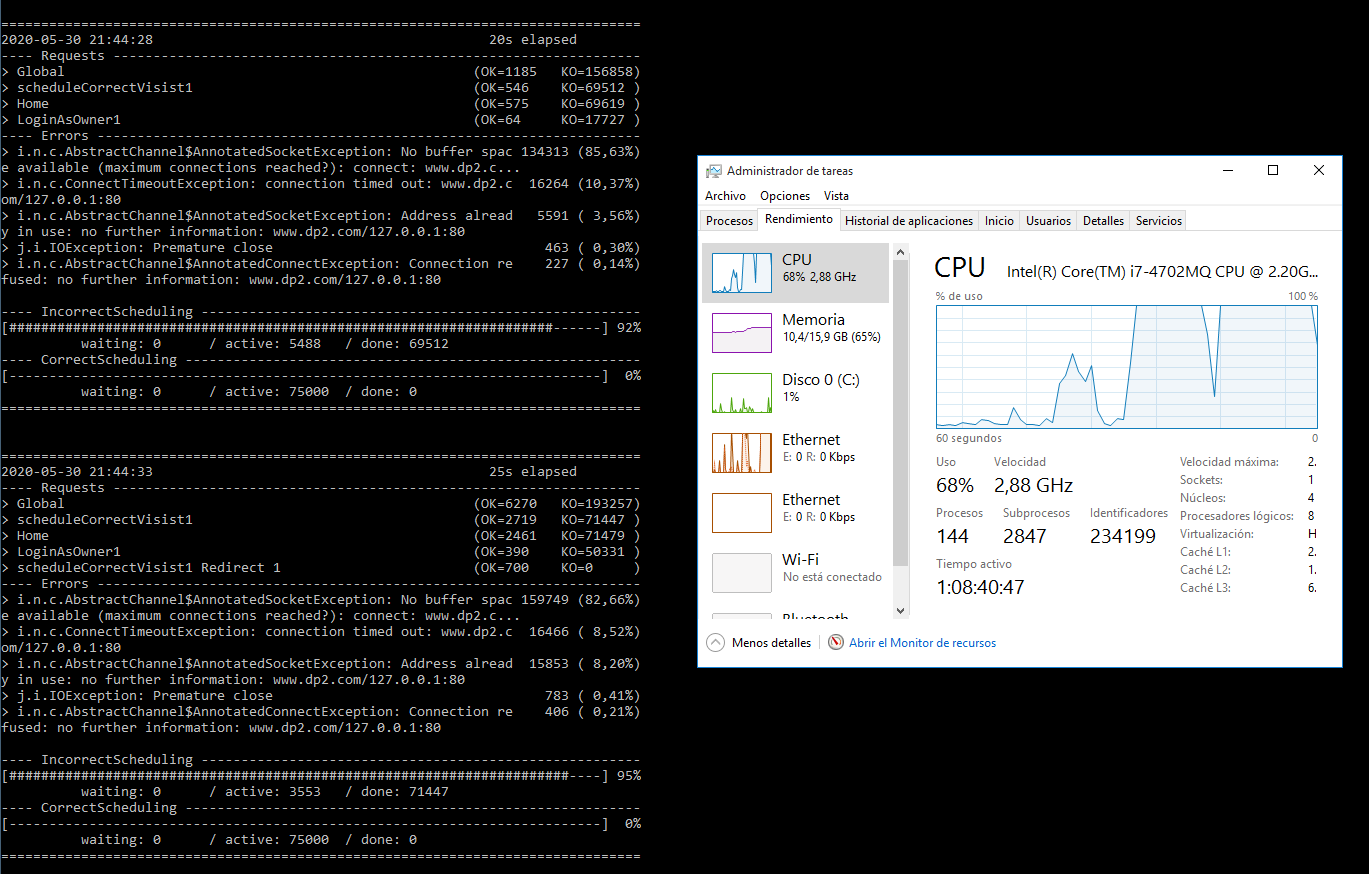
**[US4] Select a type of visit**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **150k**

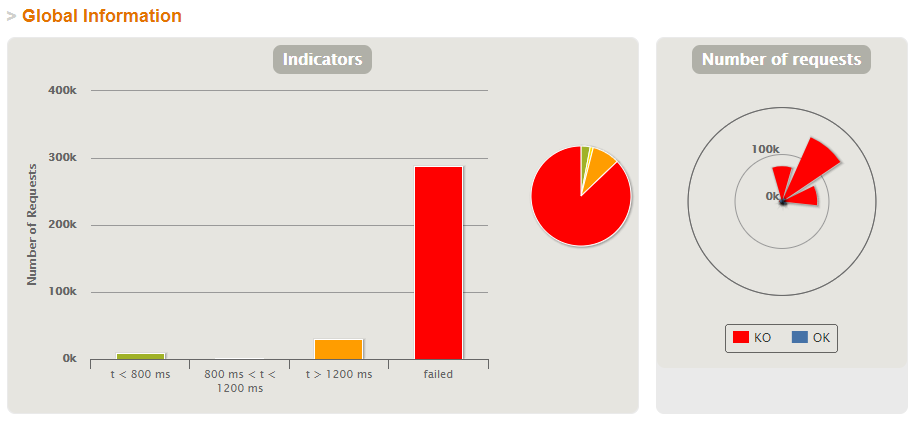
Evidences:

When performing a stress test with 150.000 (75.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

Capture of the performance monitor of our computer and the test execution console:



Capture of Gatling report:



1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **6.900**

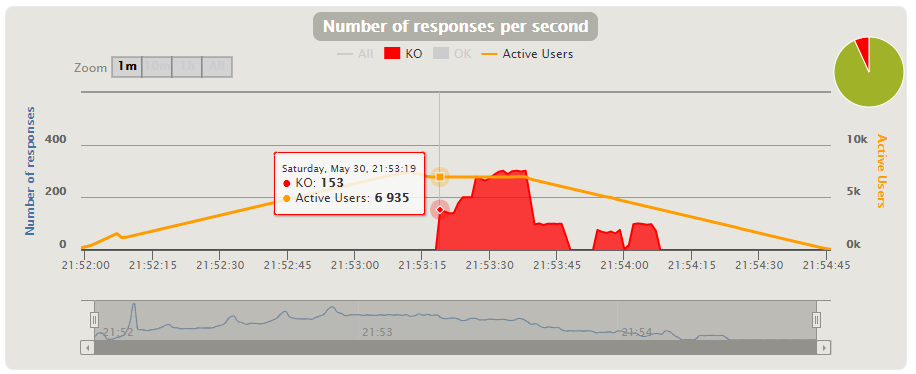
Evidences:

Performing a load test with 20.000 (10.000 users for each scenario) concurrent users, when the number of active users is 6.900 we can see that the mistakes made and failed requests begin. We can also see the bottleneck in the CPU.

Capture of the performance monitor of our computer and test execution console:



Capture of Gatling report:



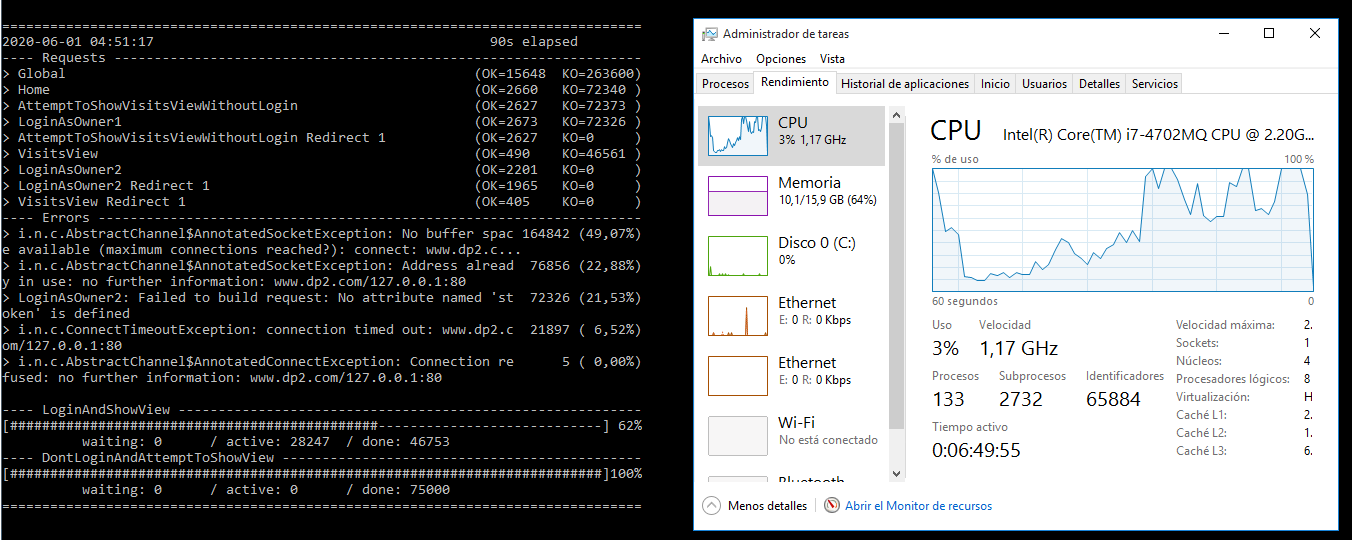
**[US5A] Upcoming visits view (pet owner)**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **150k**

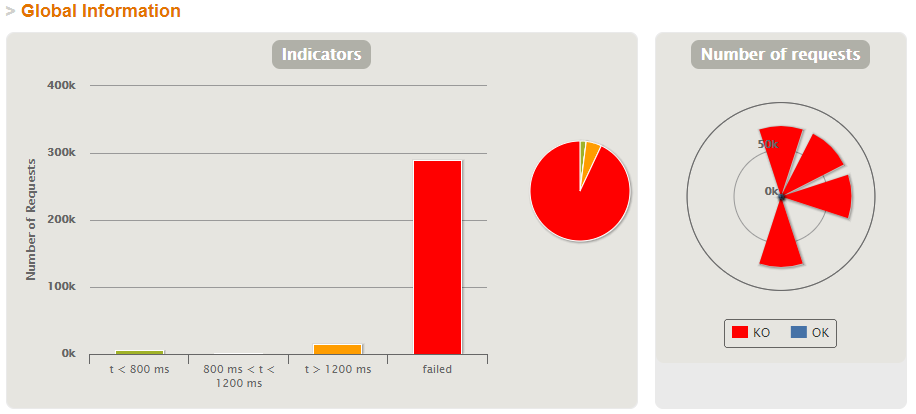
Evidences:

When performing a stress test with 150.000 (75.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

Capture of the performance monitor of our computer and the test execution console:



Capture of Gatling report:

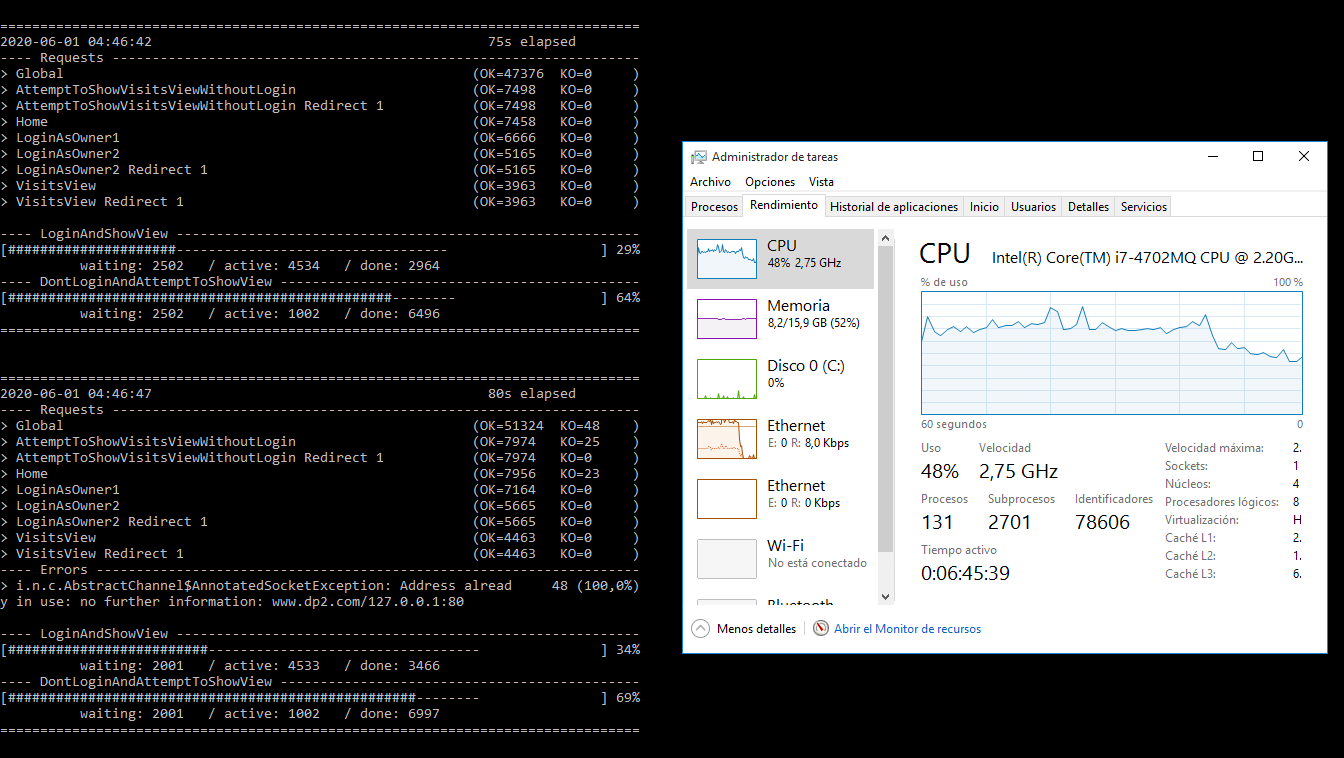


1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **5.700**

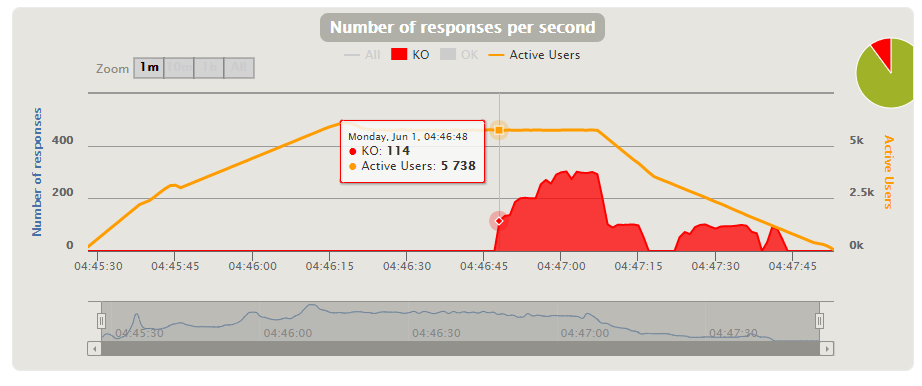
Evidences:

Performing a load test with 20.000 (10.000 users for each scenario) concurrent users, when the number of active users is 5.700 we can see that the mistakes made and failed requests begin. We can also see the bottleneck in the CPU.

Capture of the performance monitor of our computer and test execution console:



Capture of Gatling report:



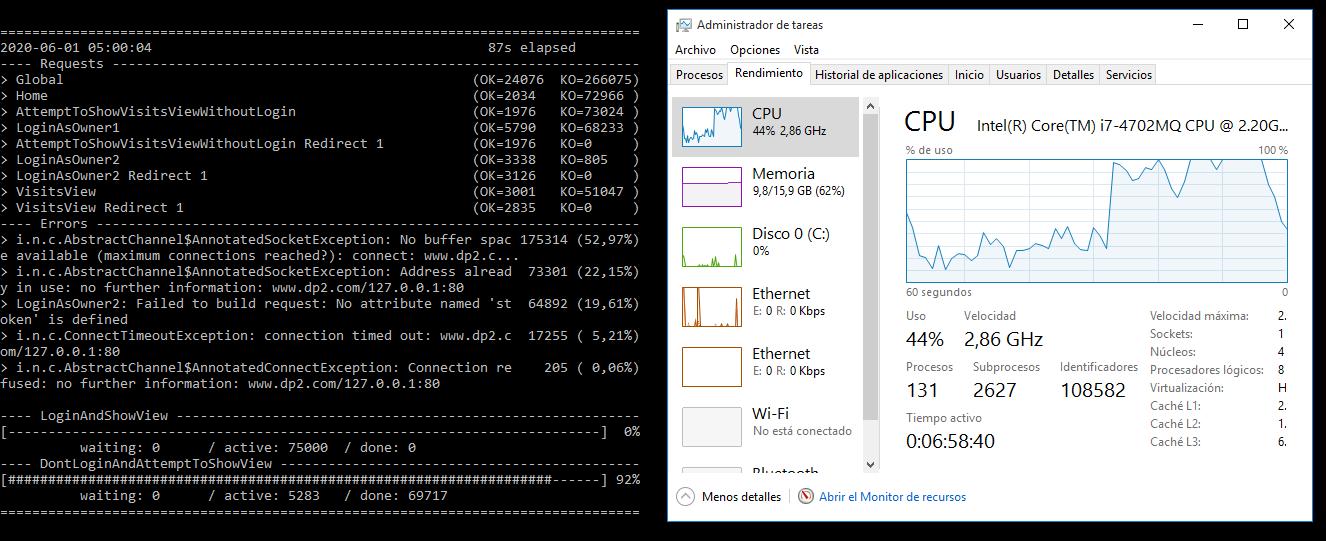
**[US5B] Past visits view (pet owner)**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **150k**

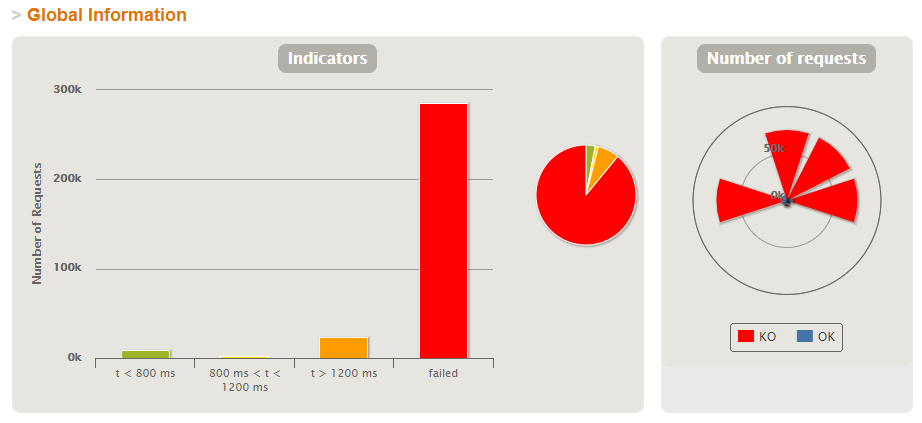
Evidences:

When performing a stress test with 150.000 (75.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

Capture of the performance monitor of our computer and the test execution console:



Capture of Gatling report:

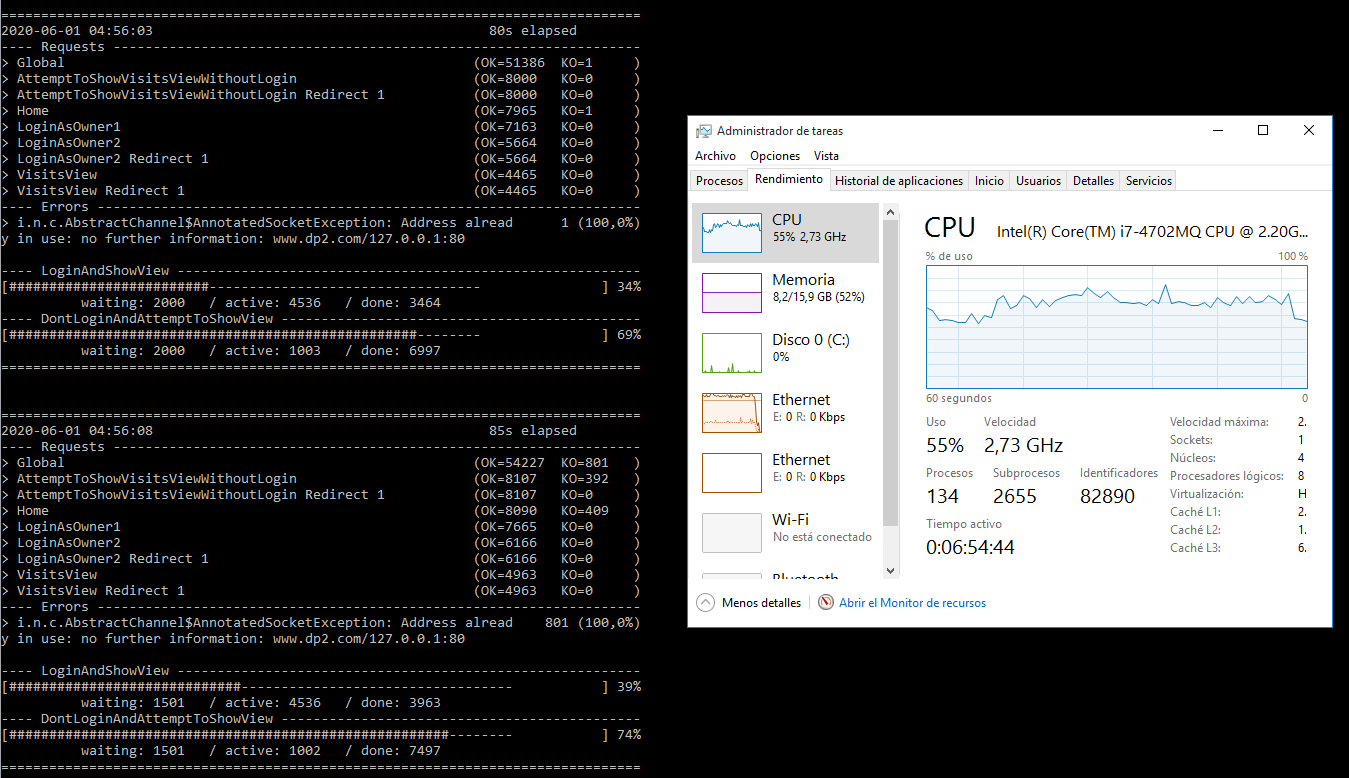


1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **5.700**

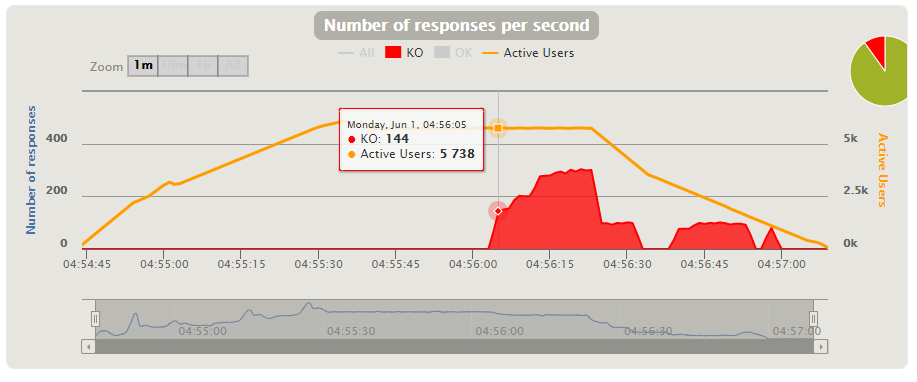
Evidences:

Performing a load test with 20.000 (10.000 users for each scenario) concurrent users, when the number of active users is 5.700 we can see that the mistakes made and failed requests begin. We can also see the bottleneck in the CPU.

Capture of the performance monitor of our computer and test execution console:



Capture of Gatling report:



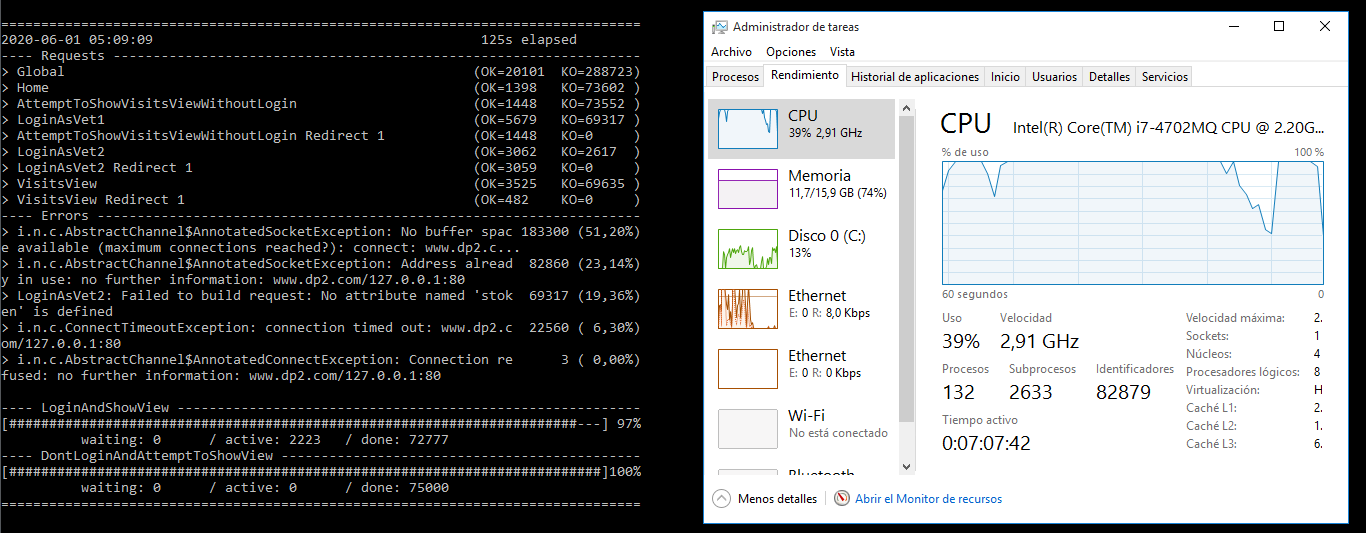
**[US6] Upcoming visits view (vet)**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **150k**

Evidences:

When performing a stress test with 150.000 (75.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

Capture of the performance monitor of our computer and the test execution console:



Capture of Gatling report:

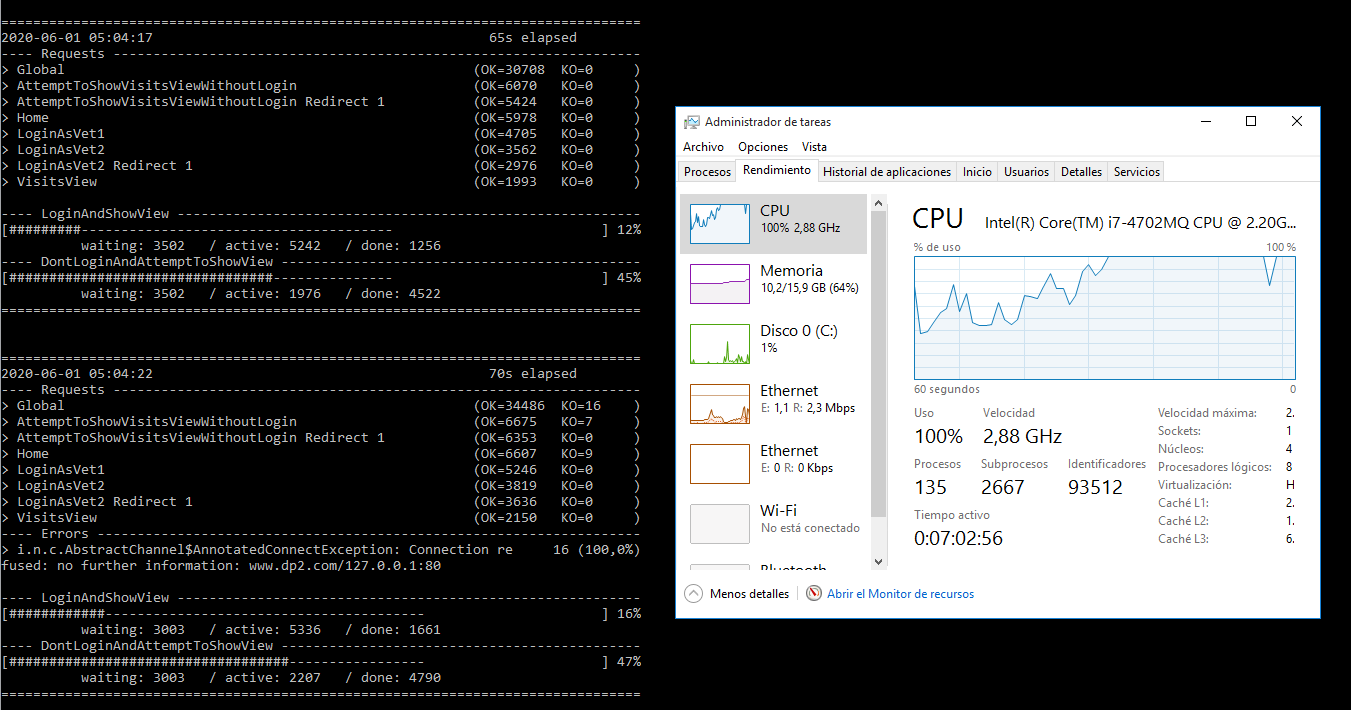


1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **7.500**

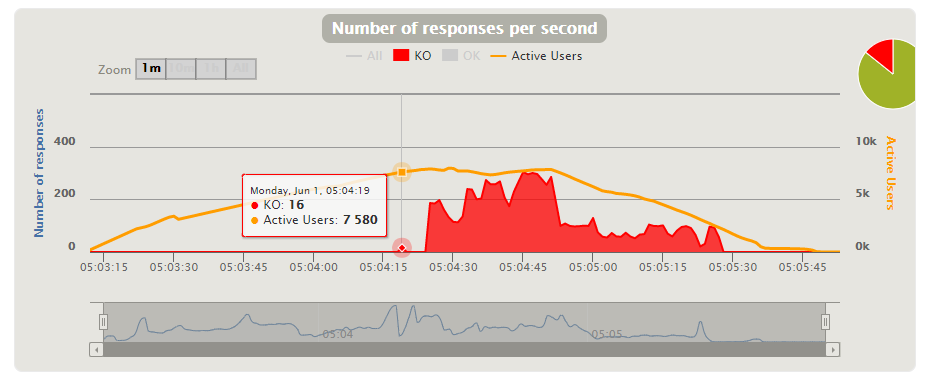
Evidences:

Performing a load test with 20.000 (10.000 users for each scenario) concurrent users, when the number of active users is 7.500 we can see that the mistakes made and failed requests begin. We can also see the bottleneck in the CPU.

Capture of the performance monitor of our computer and test execution console:



Capture of Gatling report:

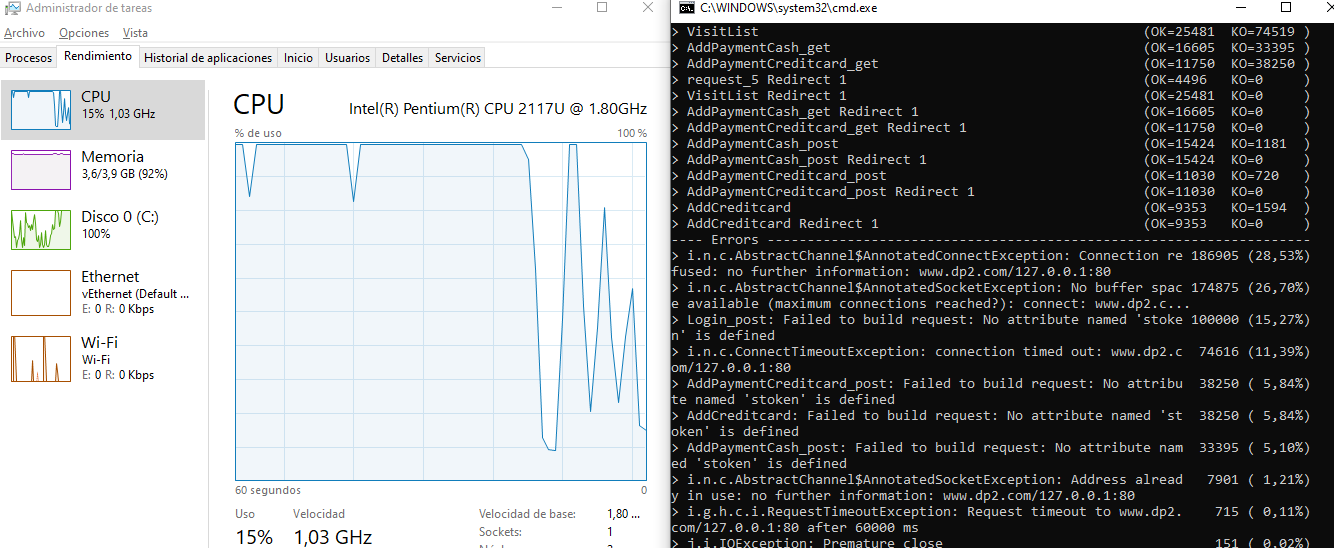


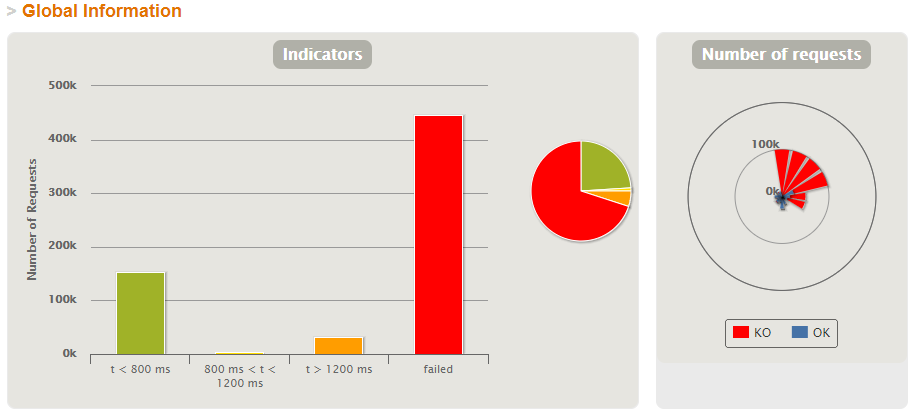
**[US7] Register a payment with credit card or cash**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **100.000 in 10**

Evidences:

When doing a stress test with 100.000 (50.000 users for each scenario) current users, we can see that most requests are failed and the CPU has a bottleneck. So this is the minimum number that my system not supported because with this users there is bad performance.

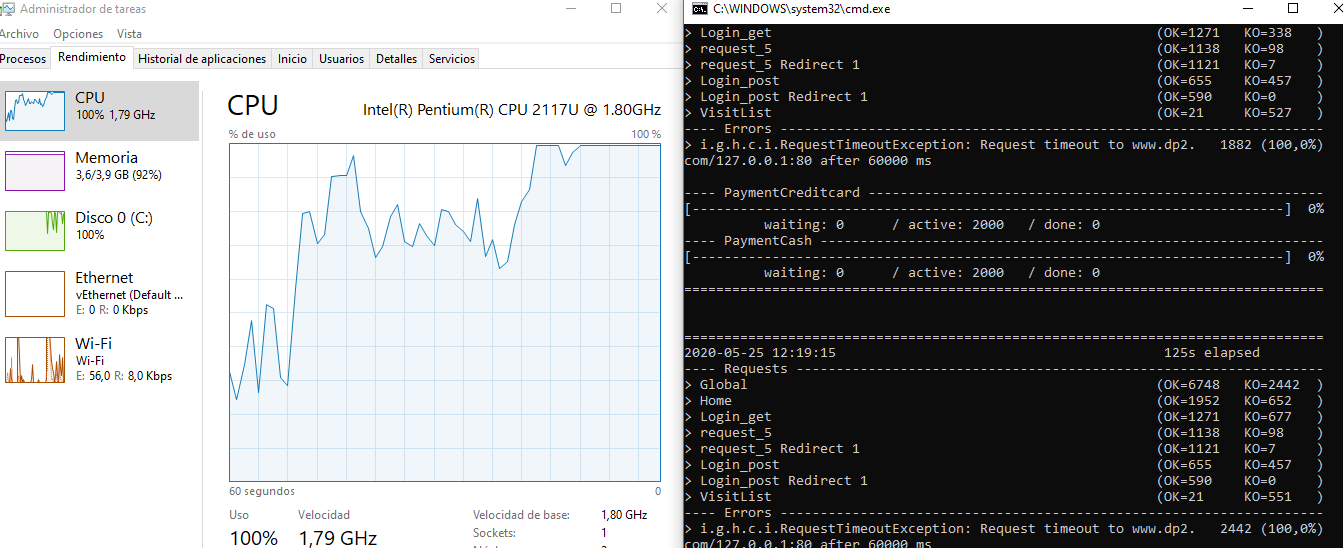
* Capture of the performance monitor of my computer and test execution console:
* Capture of Gatling report:

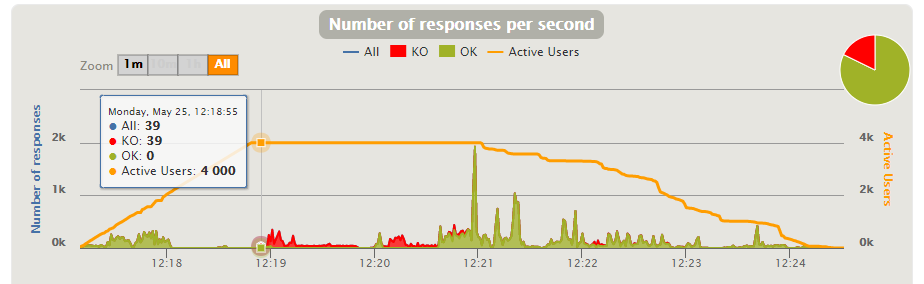


1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **4.000 in 100**

Evidences:

Doing a load test with 4.000 (2.000 users for each scenario) current users, when the number of active users is 4.000 we can see that the mistakes made and failed requests begin. Also can see the bottleneck in the CPU.

* Capture of the performance monitor of my computer and test execution console:
* Capture of Gatling report:

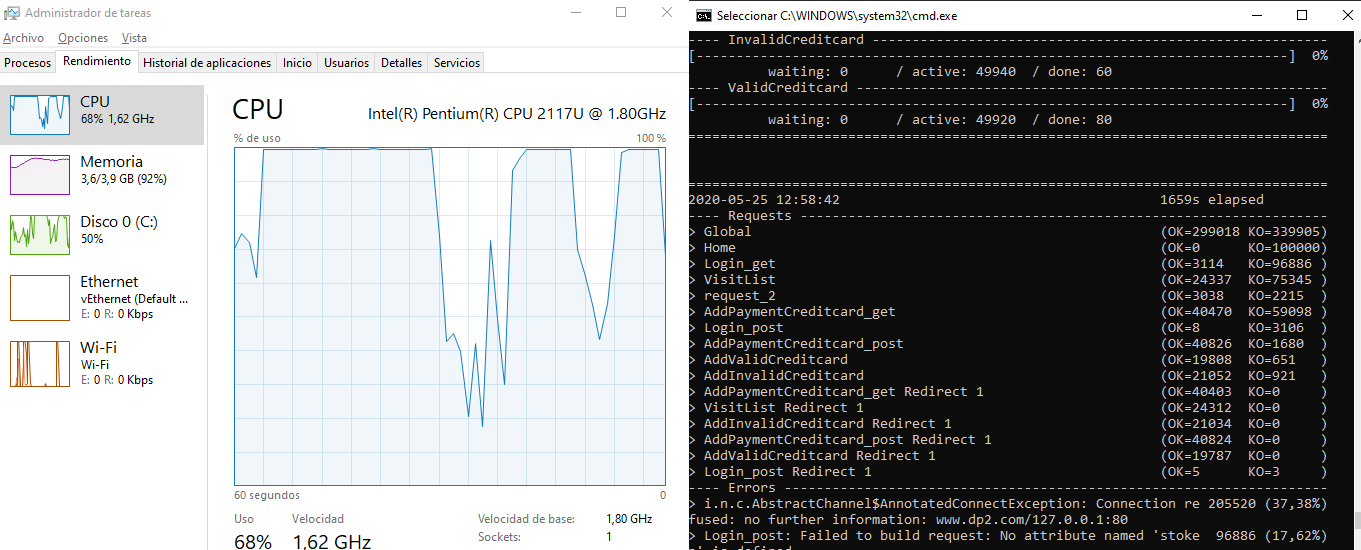


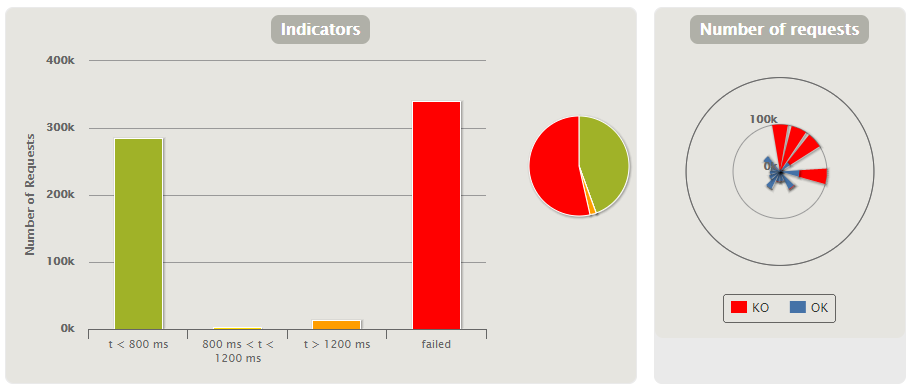
**[US8] Validate credit card**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **100.000 in 10**

Evidences:

When doing a stress test with 100.000 (50.000 users for each scenario) current users, we can see that most requests are failed and the CPU has a bottleneck. So this is the minimum number that my system not supported because with this users there is bad performance.

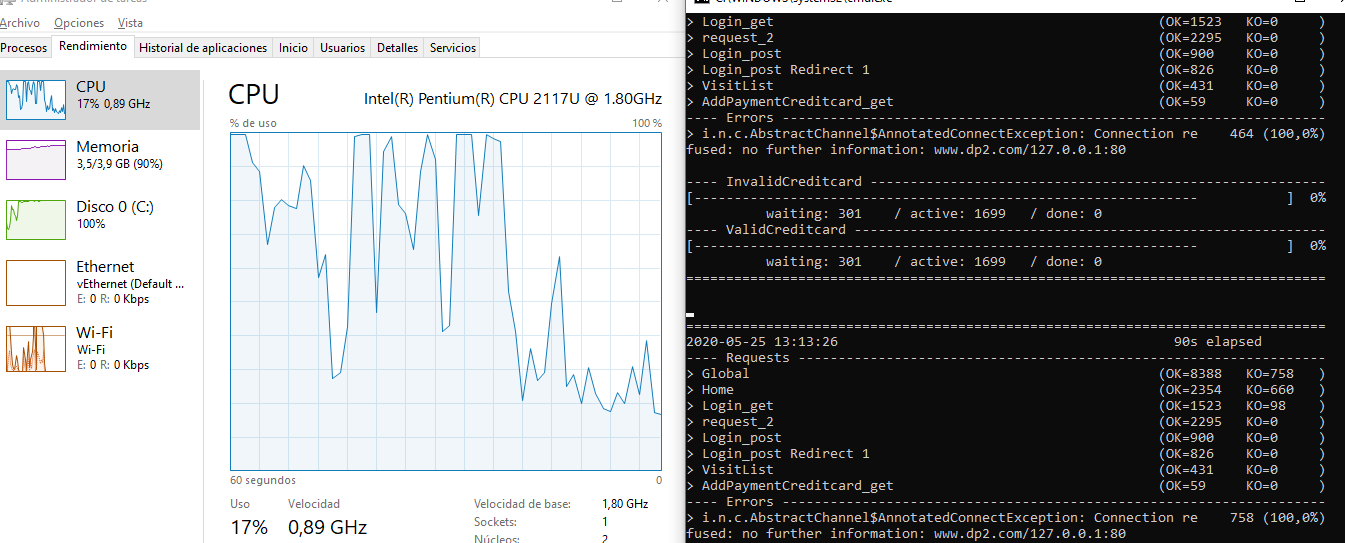
* Capture of the performance monitor of my computer and test execution console:
* Capture of Gatling report:

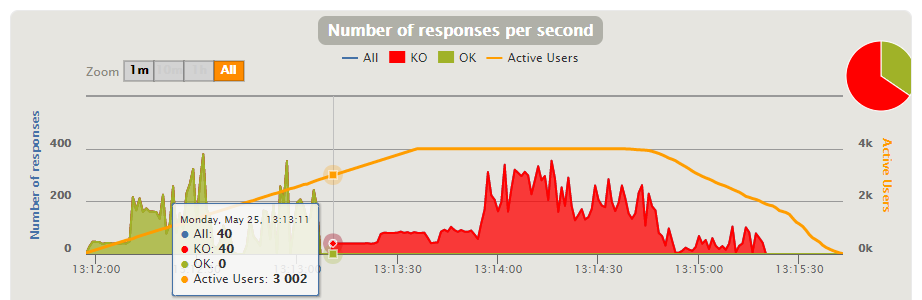


1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **3.000 in 100**

Evidences:

Doing a load test with 4.000 (2.000 users for each scenario) current users, when the number of active users is 3.000 we can see that the mistakes made and failed requests begin. Also can see the bottleneck in the CPU.

* Capture of the performance monitor of my computer and test execution console:
* Capture of Gatling report:

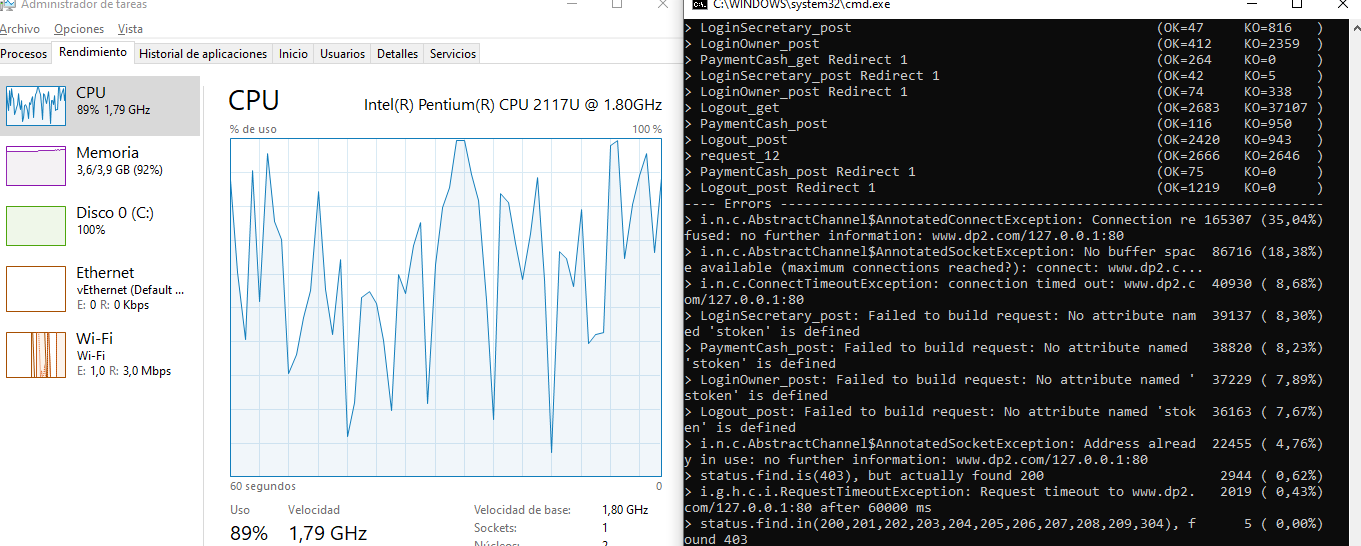


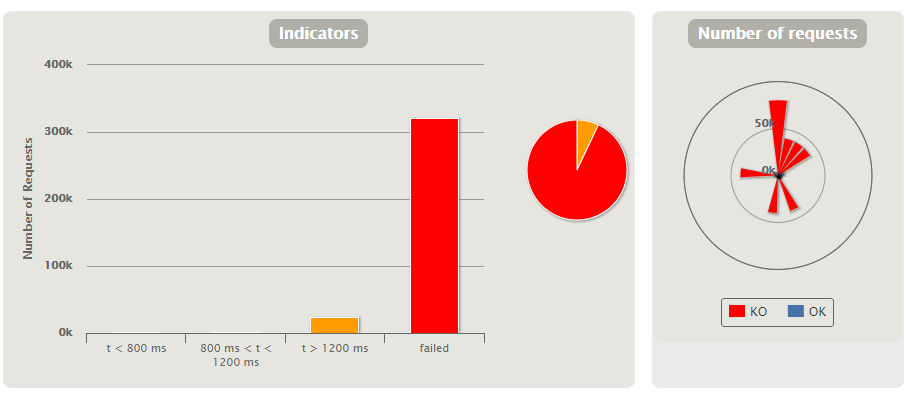
**[US9] Store who registered a payment**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **80.000 in 10**

Evidences:

When doing a stress test with 80.000 (40.000 users for each scenario) current users, we can see that most requests are failed and the CPU has a bottleneck. So this is the minimum number that my system not supported because with this users there is bad performance.

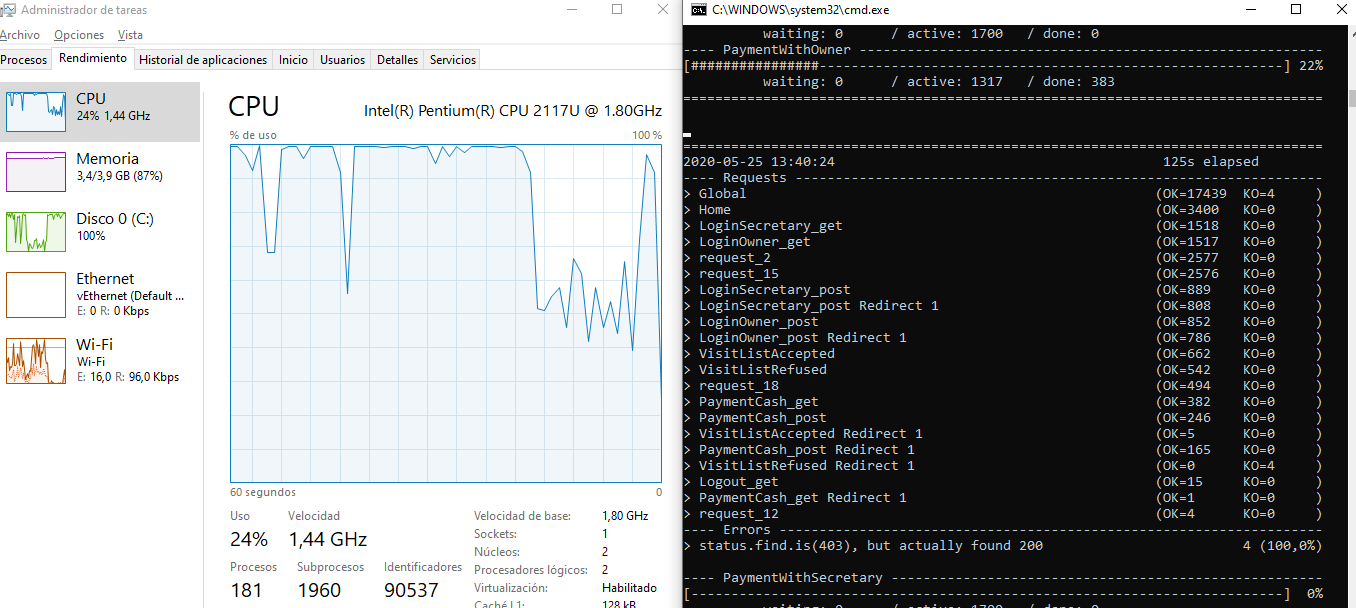
* Capture of the performance monitor of my computer and test execution console:
* Capture of Gatling report:

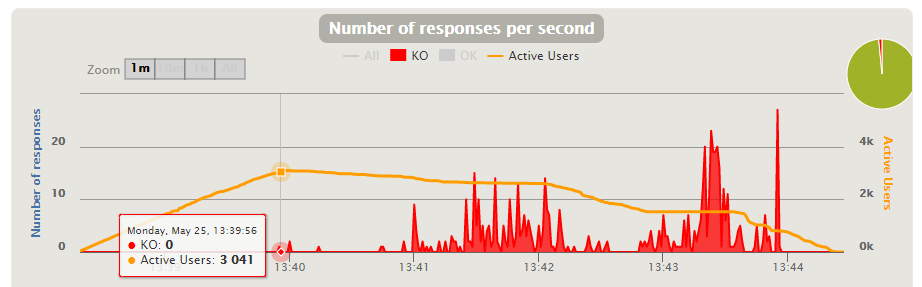


1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **3.000 in 100**

Evidences:

Doing a load test with 3.400 (1.700 users for each scenario) current users, when the number of active users is 3.000 we can see that the mistakes made and failed requests begin. Also can see the bottleneck in the CPU.

* Capture of the performance monitor of my computer and test execution console:
* Capture of Gatling report:

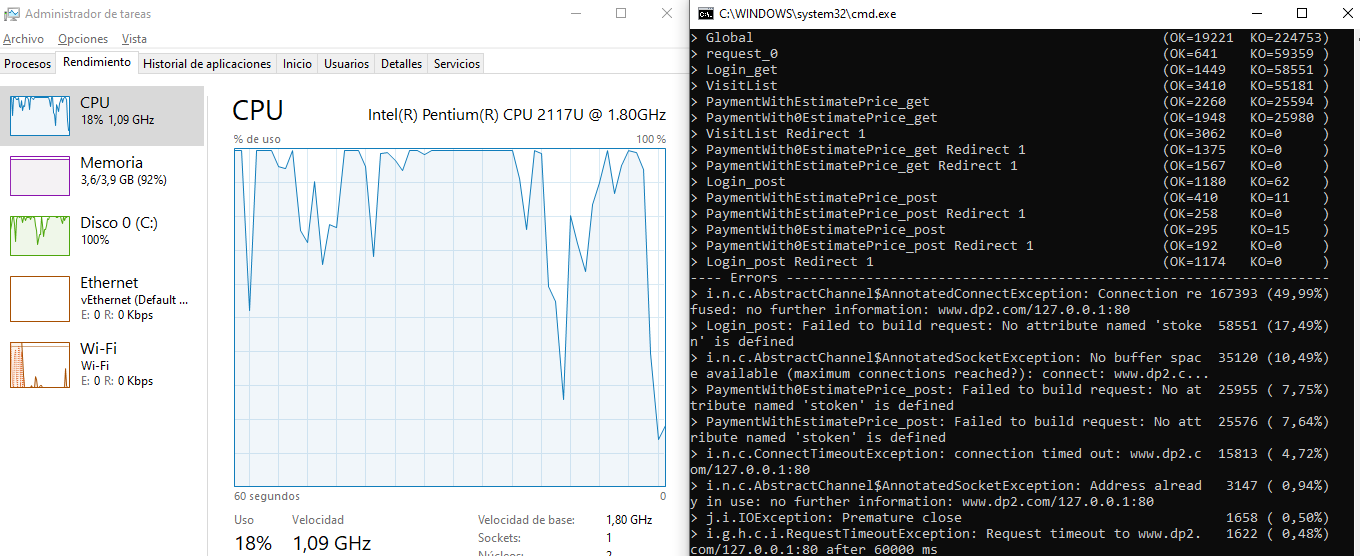


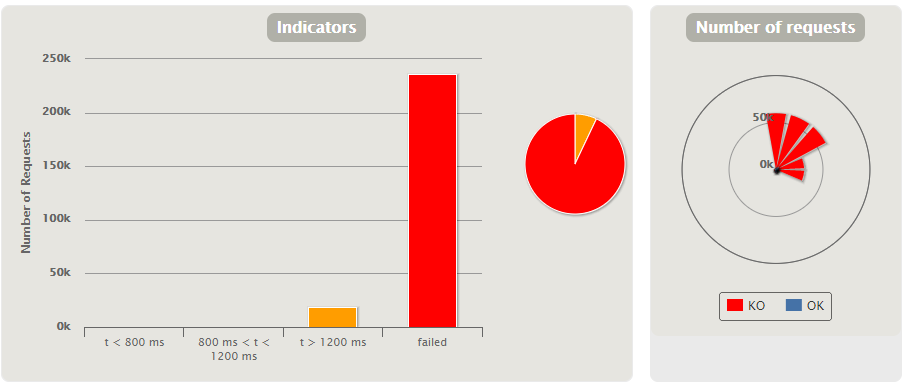
**[US10] Suggest price for a visit based on its type**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **60.000 in 10**

Evidences:

When doing a stress test with 60.000 (30.000 users for each scenario) current users, we can see that most requests are failed and the CPU has a bottleneck. So this is the minimum number that my system not supported because with this users there is bad performance.

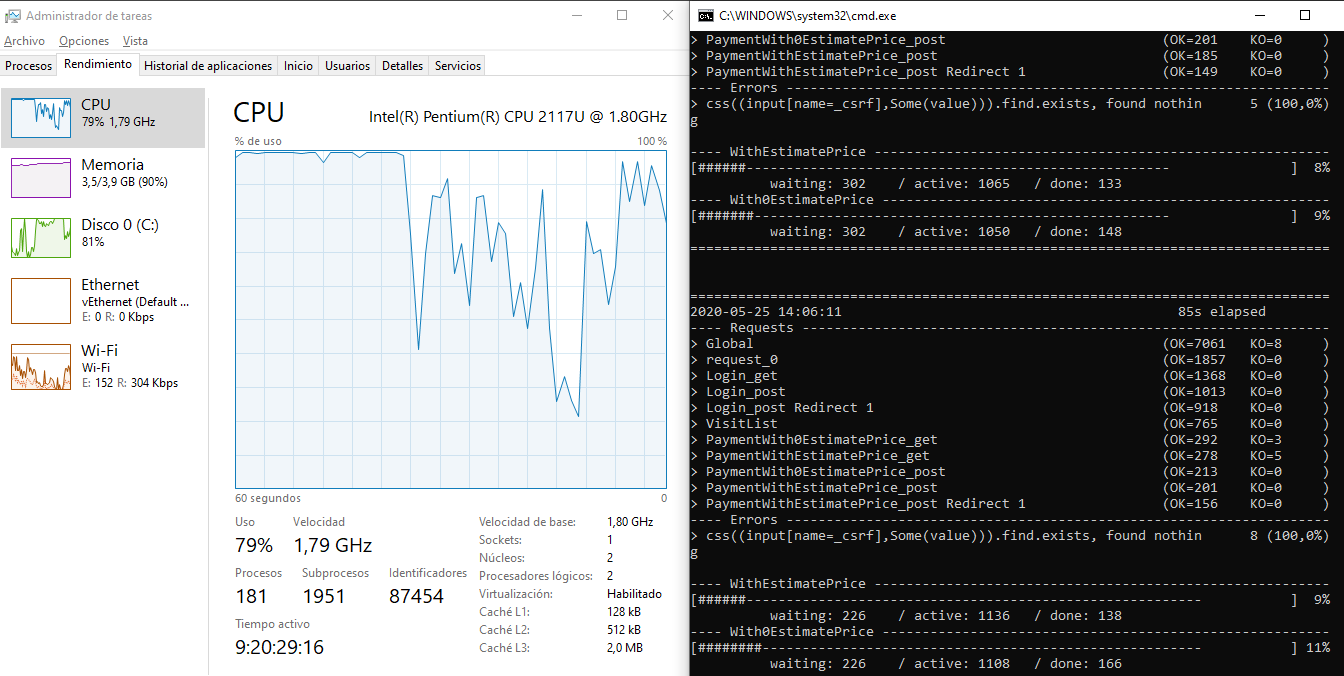
* Capture of the performance monitor of my computer and test execution console:
* Capture of Gatling report:

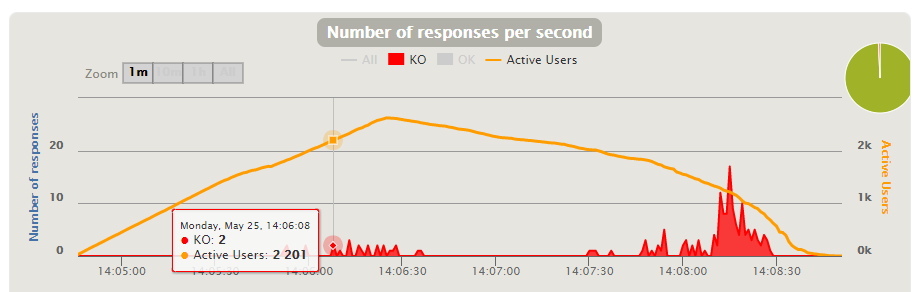


1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **2.200 in 100**

Evidences:

Doing a load test with 3.000 (1.500 users for each scenario) current users, when the number of active users is 2.200 we can see that the mistakes made and failed requests begin. Also can see the bottleneck in the CPU.

* Capture of the performance monitor of my computer and test execution console:
* Capture of Gatling report:

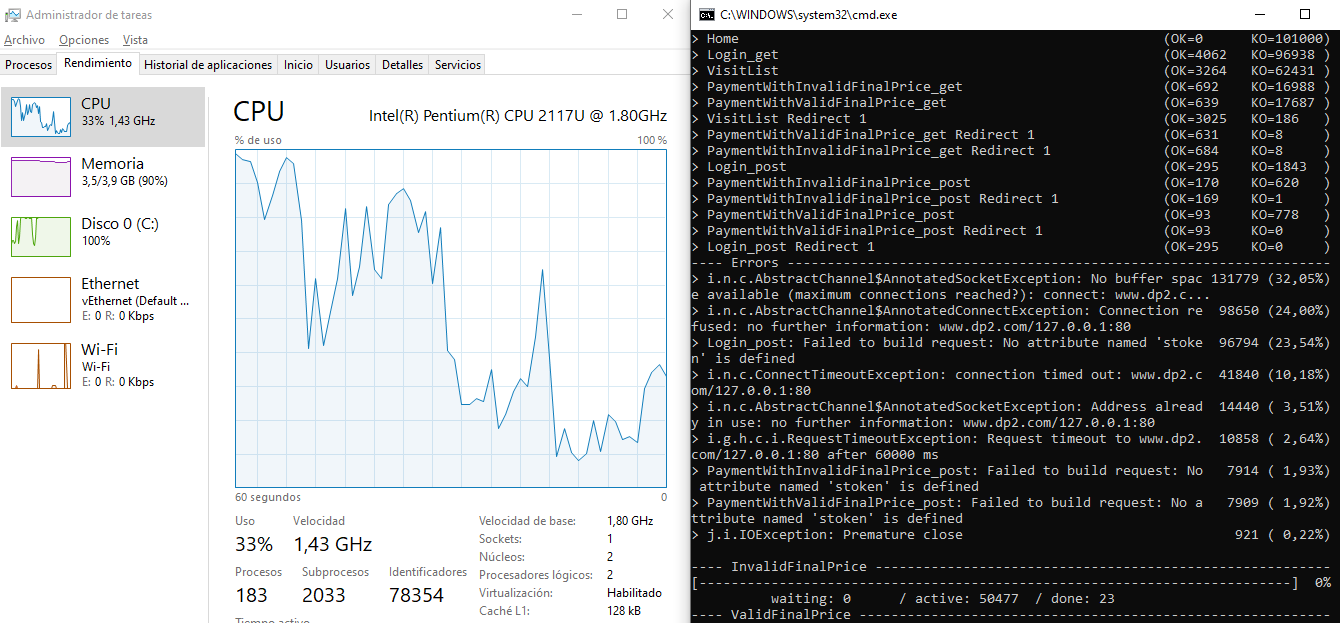


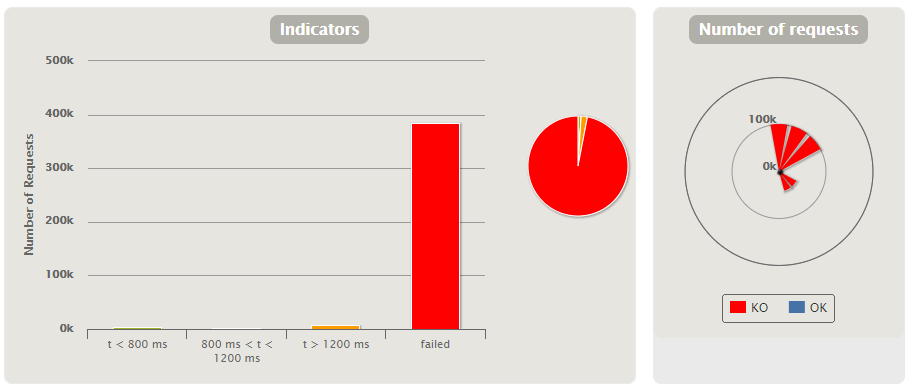
**[US11] Freely assign price to a visit**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **101.000 in 10**

Evidences:

When doing a stress test with 101.000 (50.500 users for each scenario) current users, we can see that most requests are failed and the CPU has a bottleneck. So this is the minimum number that my system not supported because with this users there is bad performance.

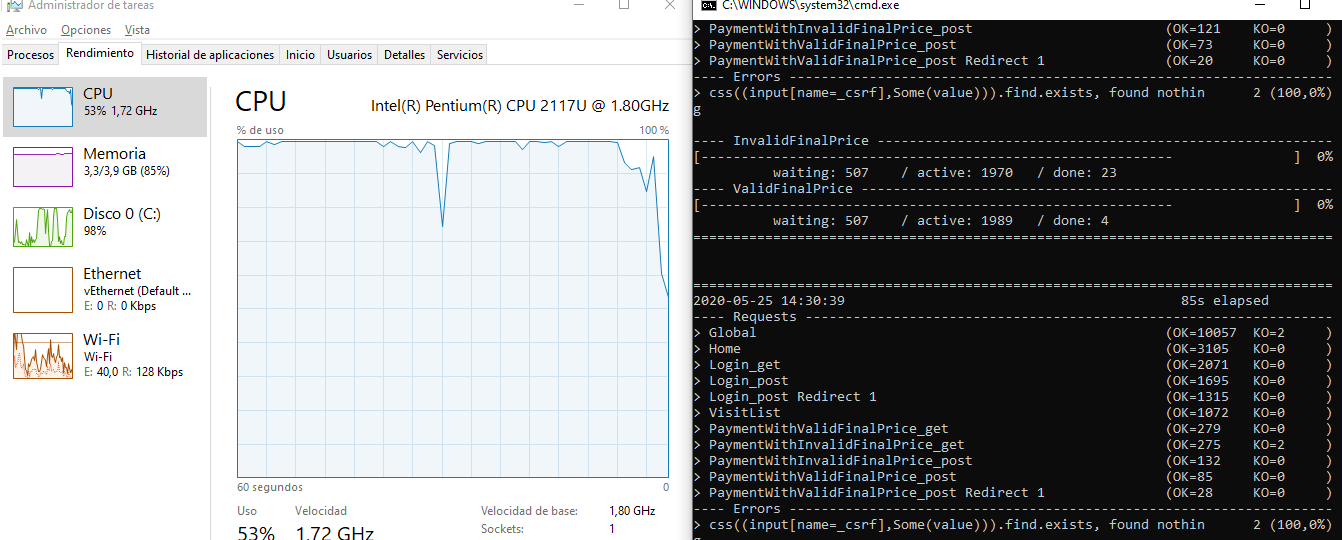
* Capture of the performance monitor of my computer and test execution console:
* Capture of Gatling report:

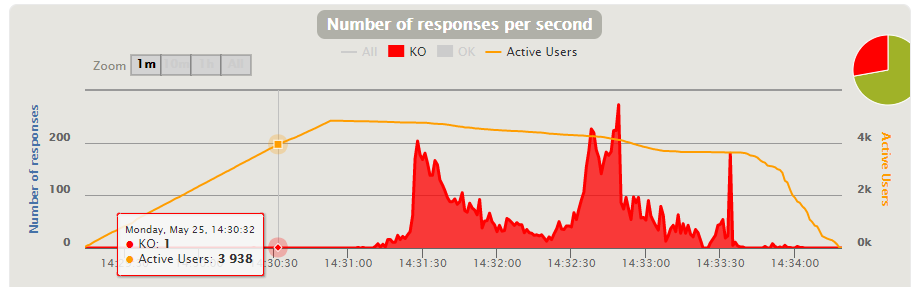


1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **4.000 in 100**

Evidences:

Doing a load test with 5.000 (2.500 users for each scenario) current users, when the number of active users is 4.000 we can see that the mistakes made and failed requests begin. Also can see the bottleneck in the CPU.

* Capture of the performance monitor of my computer and test execution console:
* Capture of Gatling report:

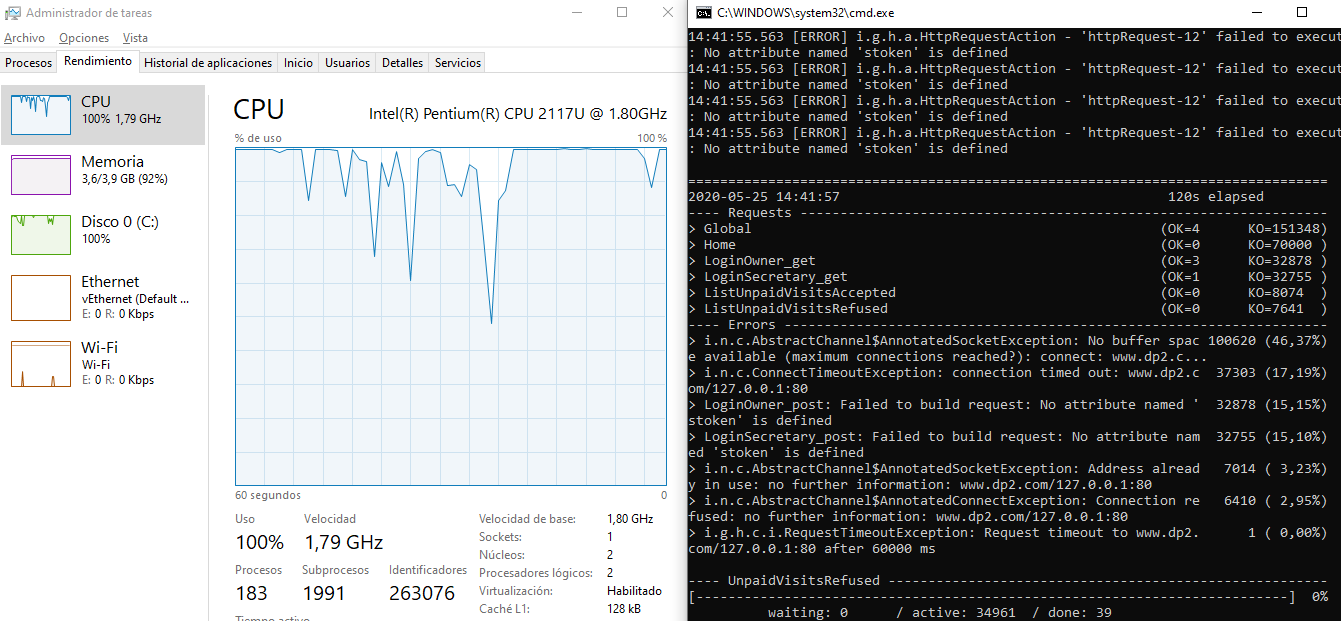


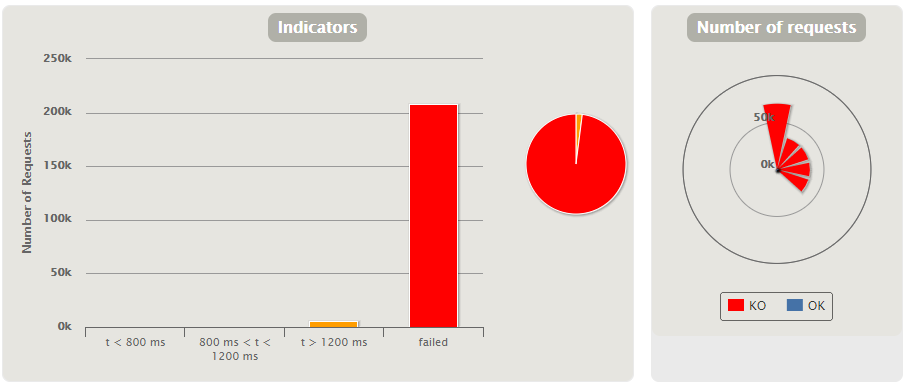
**[US12] View all unpaid visits**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **70.000 in 10**

Evidences:

When doing a stress test with 70.000 (35.000 users for each scenario) current users, we can see that most requests are failed and the CPU has a bottleneck. So this is the minimum number that my system not supported because with this users there is bad performance.

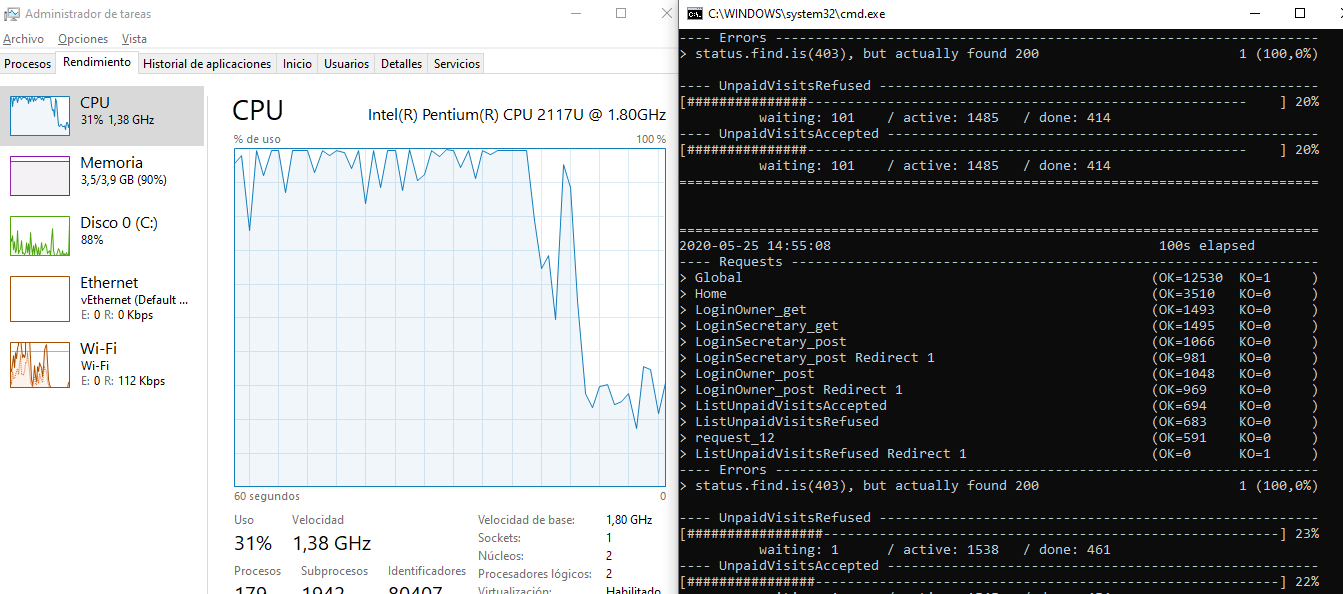
* Capture of the performance monitor of my computer and test execution console:
* Capture of Gatling report:

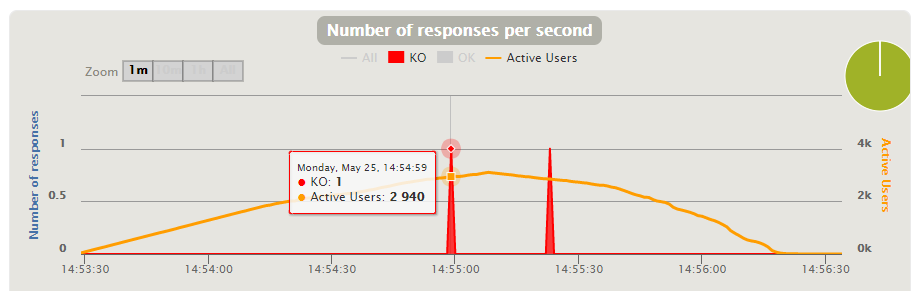


1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **3.000 in 100**

Evidences:

Doing a load test with 4.000 (2.000 users for each scenario) current users, when the number of active users is 3.000 we can see that the mistakes made and failed requests begin. Also can see the bottleneck in the CPU.

* Capture of the performance monitor of my computer and test execution console:
* Capture of Gatling report:



**[US13] Add diagnosis to a visit**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **120k**

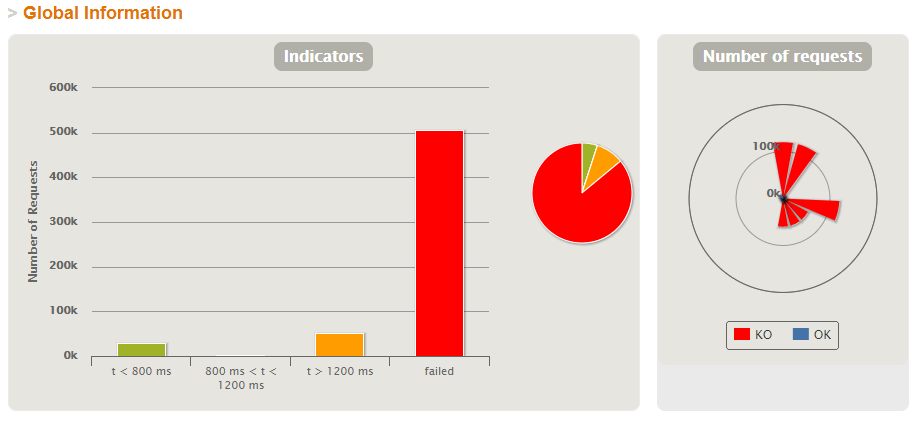
Evidences:

When performing a stress test with 120.000 (60.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

Capture of the performance monitor of our computer and the test execution console:



Capture of Gatling report:

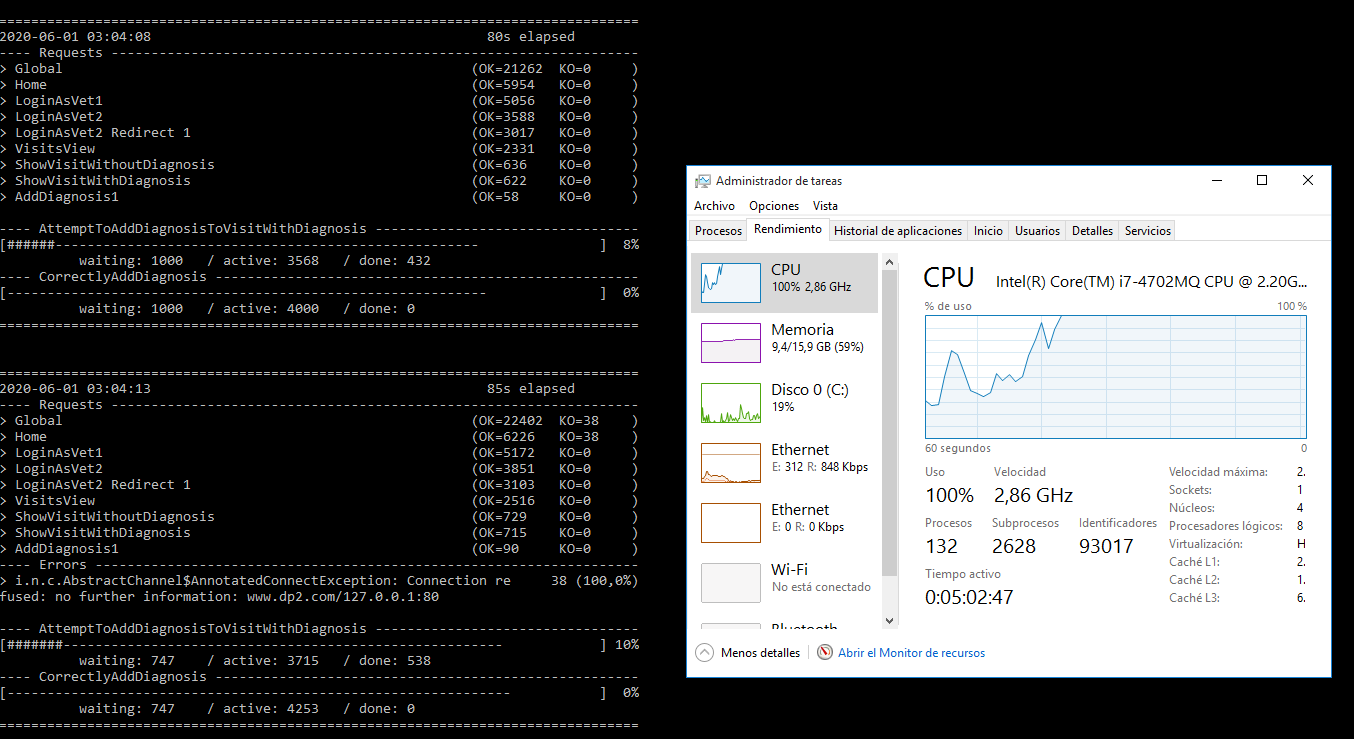


1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **8.100**

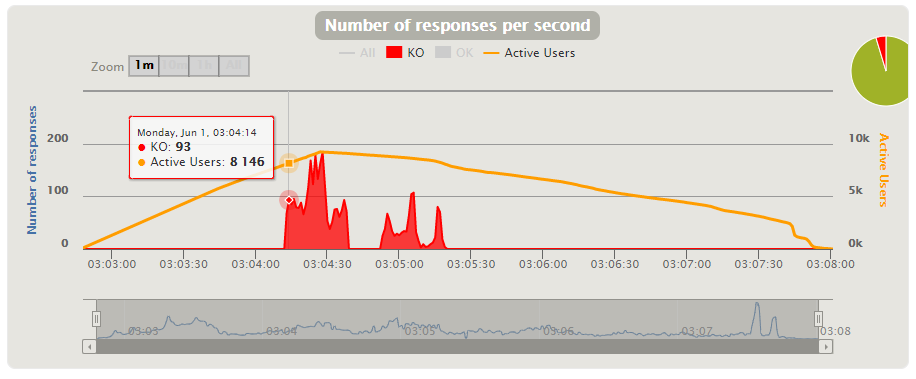
Evidences:

Performing a load test with 10.000 (5.000 users for each scenario) concurrent users, when the number of active users is 8.100 we can see that the mistakes made and failed requests begin. We can also see the bottleneck in the CPU.

Capture of the performance monitor of our computer and test execution console:



Capture of Gatling report:



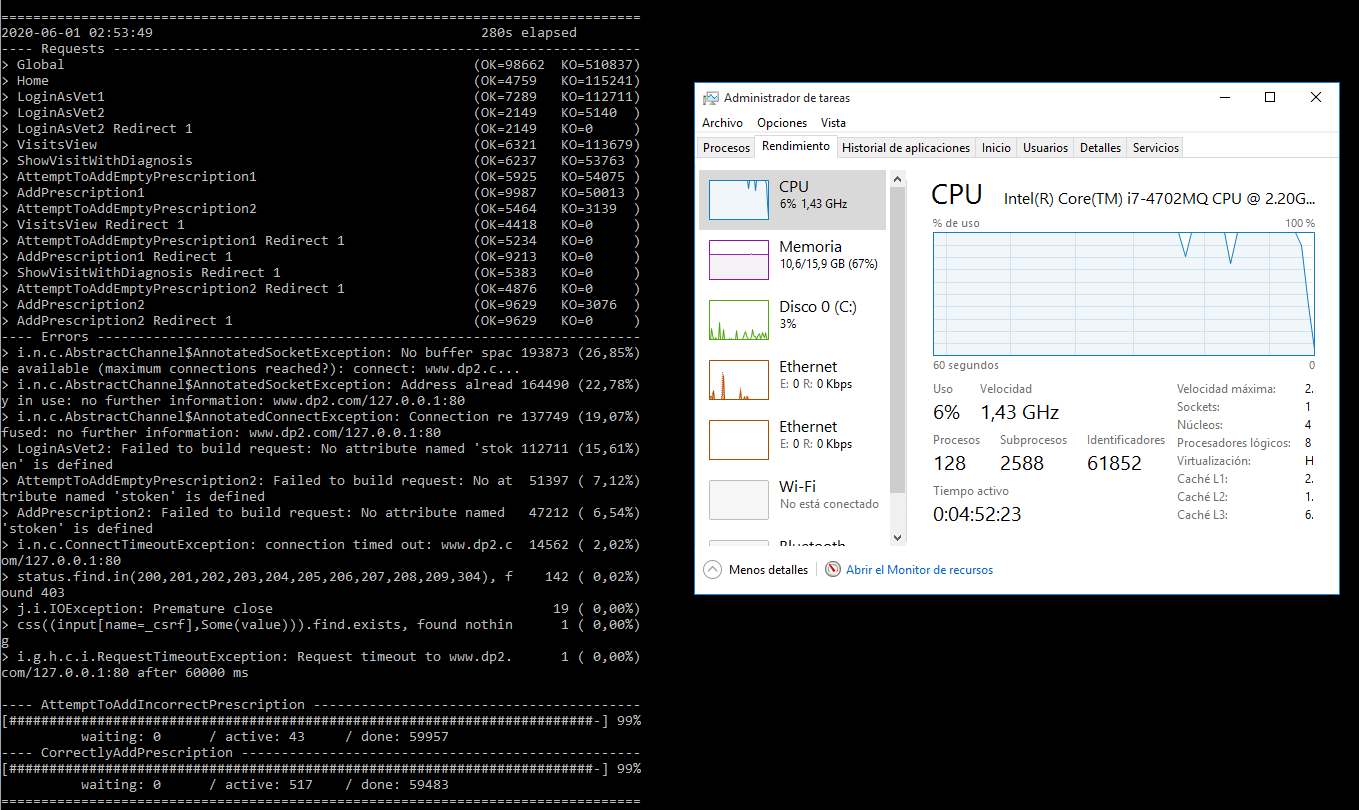
**[US14] Add prescriptions to a diagnosis**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **120k**

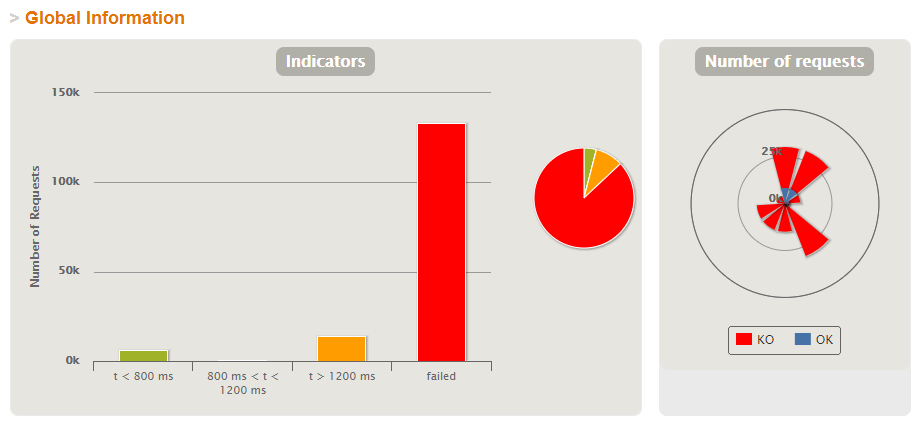
Evidences:

When performing a stress test with 120.000 (60.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

Capture of the performance monitor of our computer and the test execution console:



Capture of Gatling report:

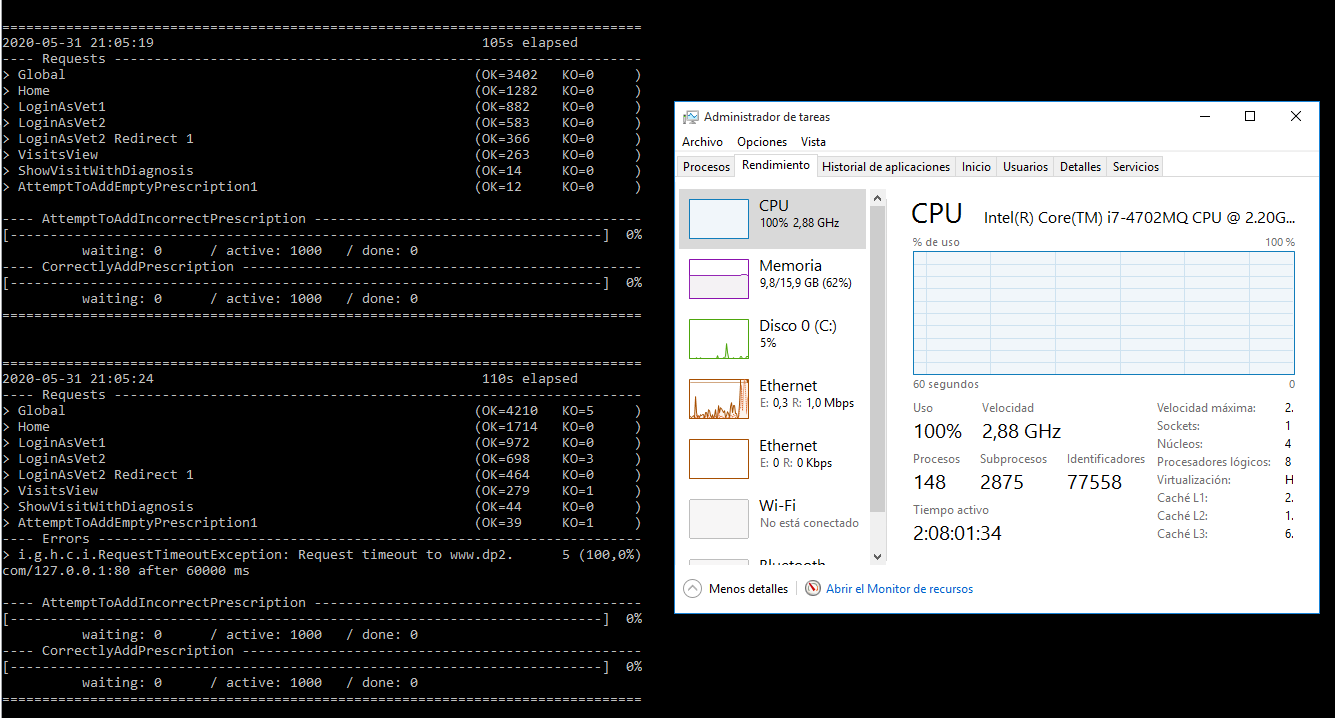


1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **1.900**

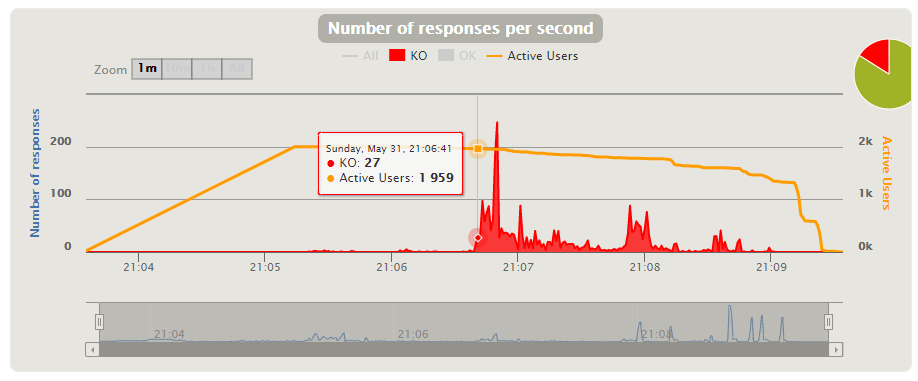
Evidences:

Performing a load test with 2000 (1000 users for each scenario) concurrent users, when the number of active users is 1900 we can see that the mistakes made and failed requests begin. We can also see the bottleneck in the CPU.

Capture of the performance monitor of our computer and test execution console:



Capture of Gatling report:



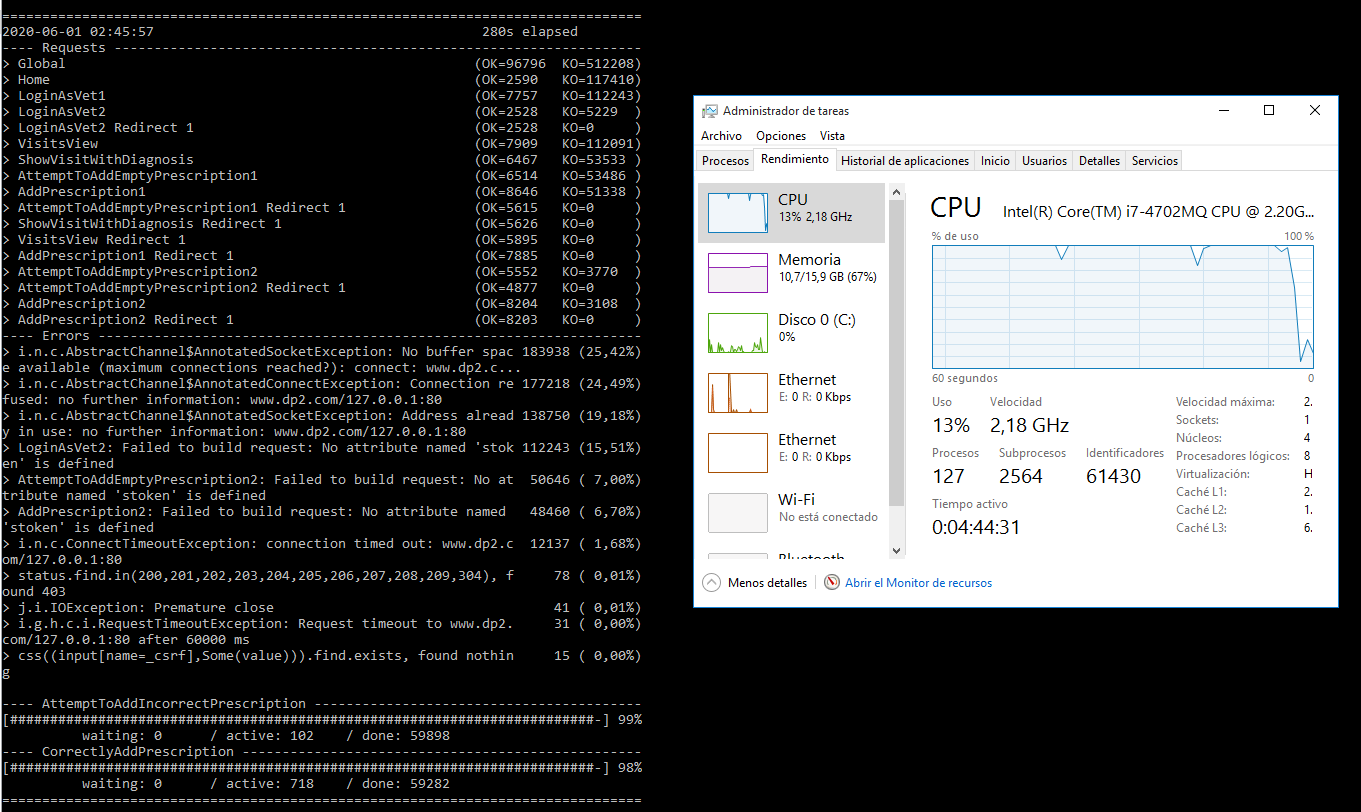
**[US15] Select medicine from database**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **120k**

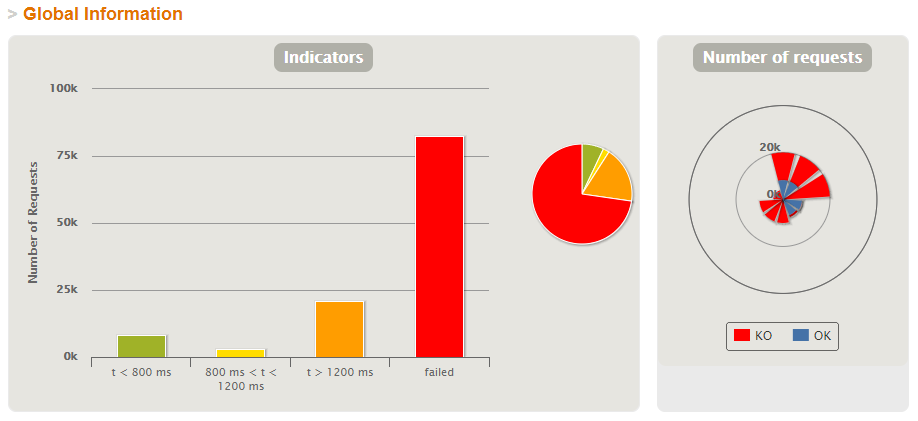
Evidences:

When performing a stress test with 120.000 (60.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

Capture of the performance monitor of our computer and the test execution console:



Capture of Gatling report:

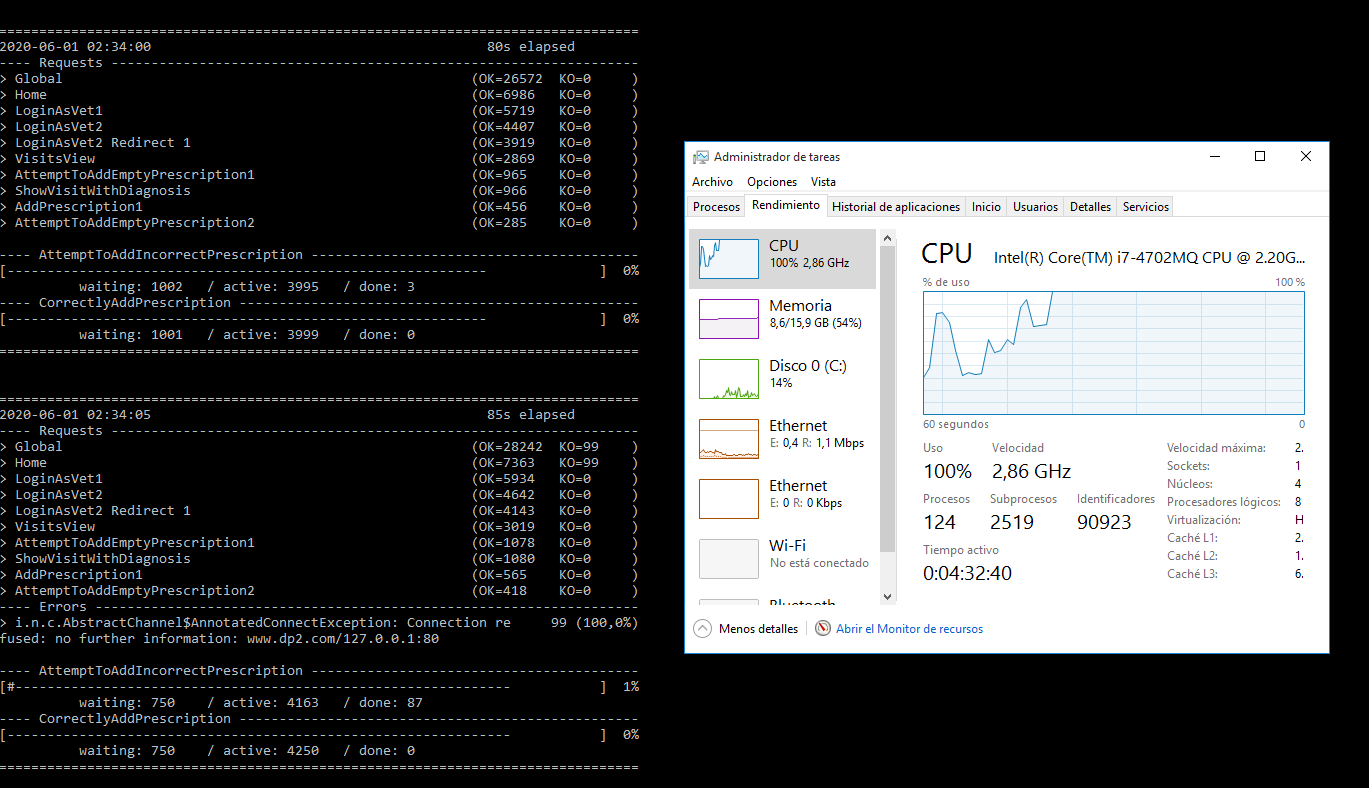


1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **8.400**

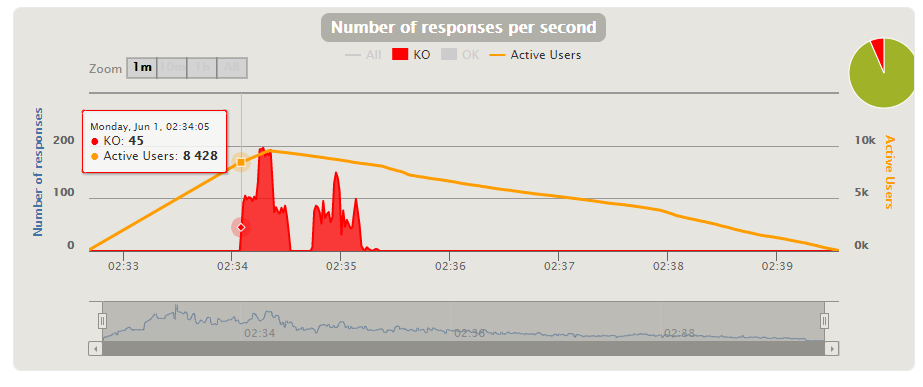
Evidences:

Performing a load test with 10.000 (8.400 users for each scenario) concurrent users, when the number of active users is 8.100 we can see that the mistakes made and failed requests begin. We can also see the bottleneck in the CPU.

Capture of the performance monitor of our computer and test execution console:



Capture of Gatling report:



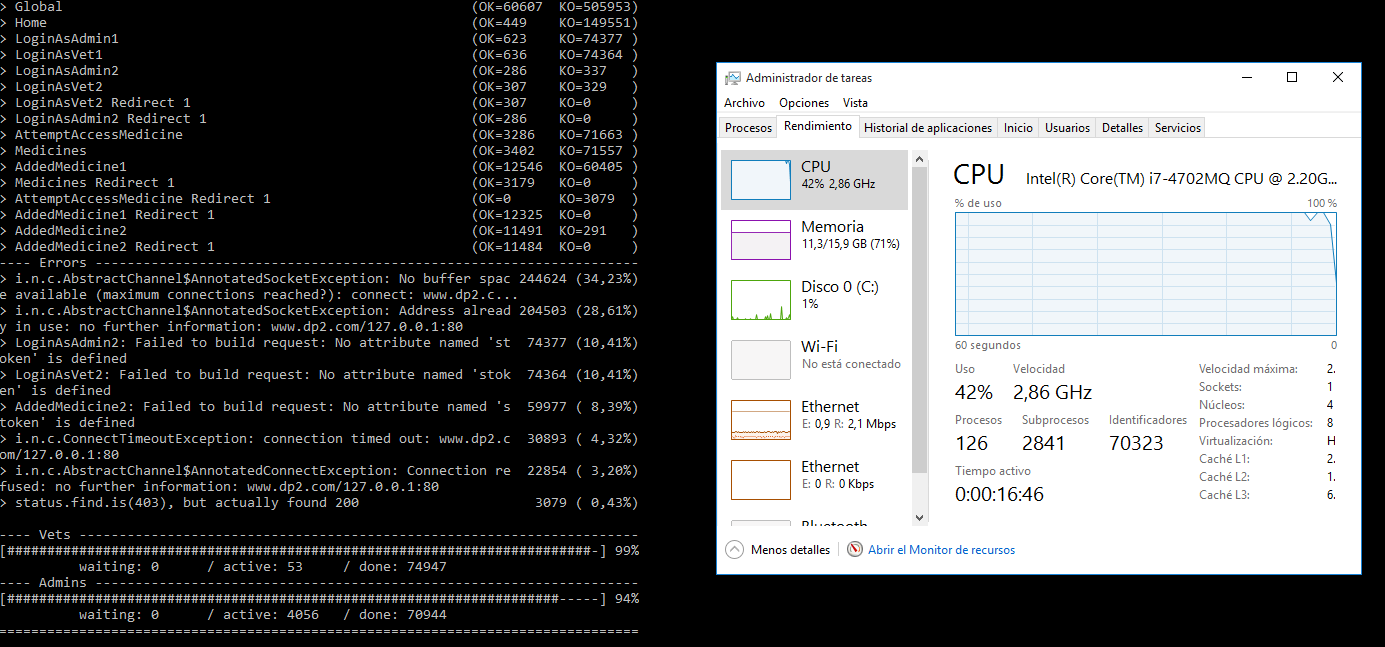
**[US16] Add new medicine the system**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **150k**

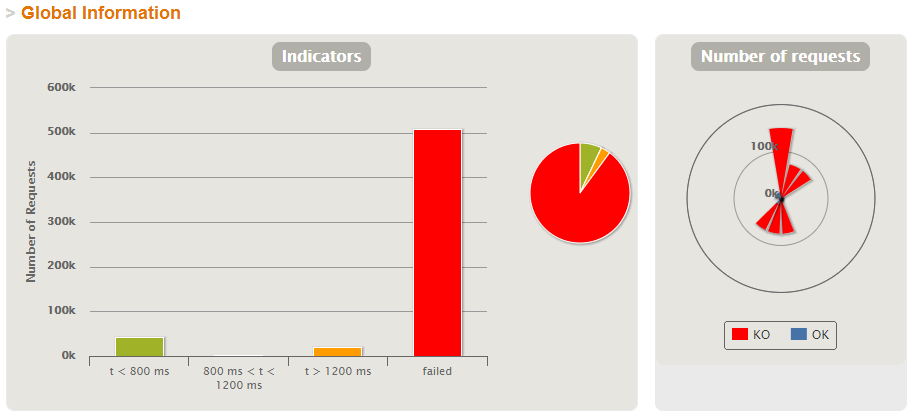
Evidences:

When performing a stress test with 150.000 (75.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

Capture of the performance monitor of our computer and the test execution console:



Capture of Gatling report:

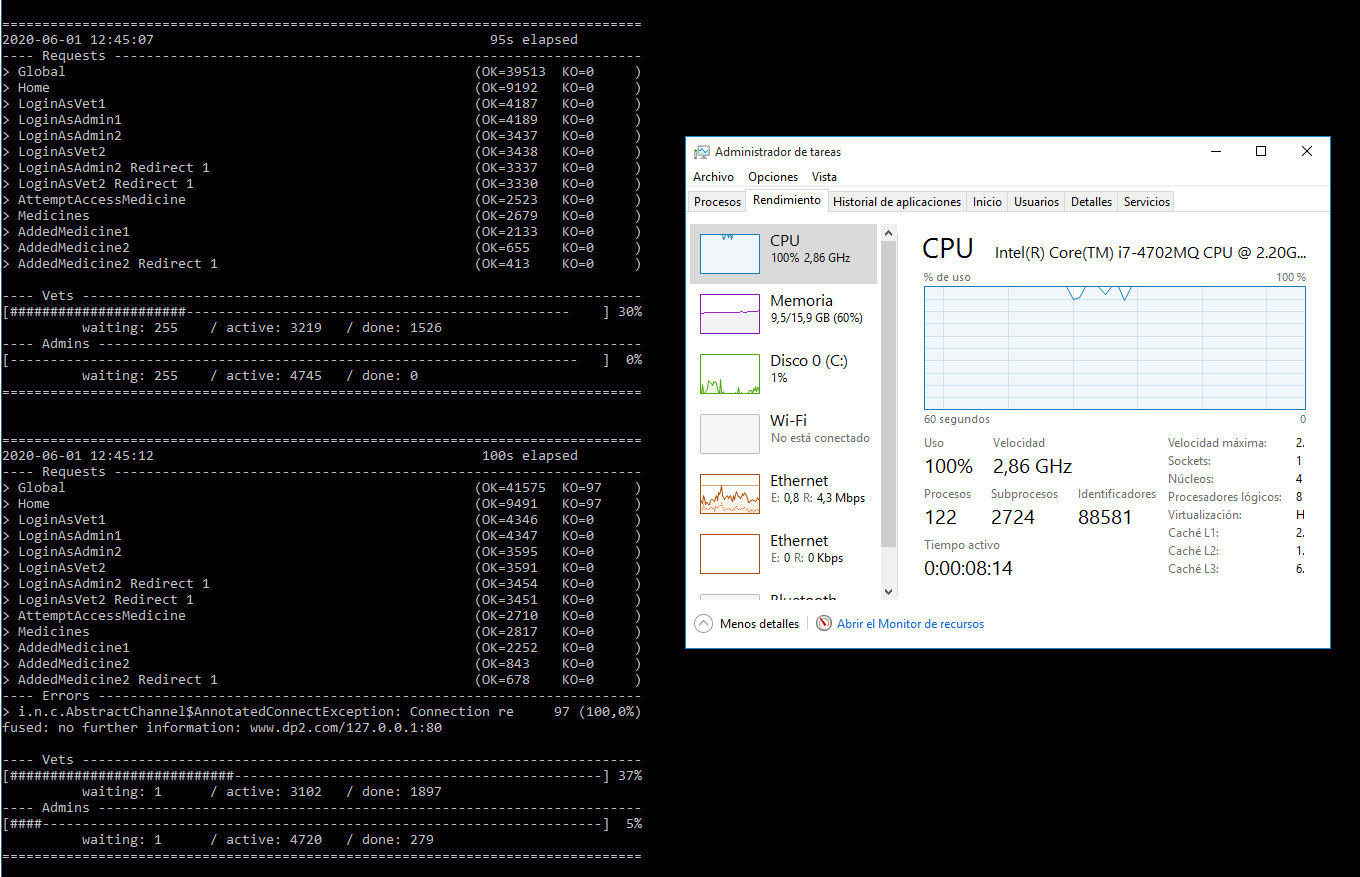


1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **8.000**

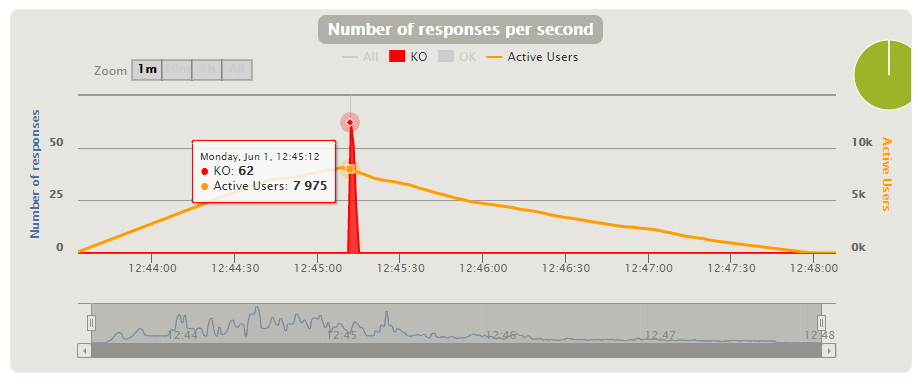
Evidences:

Performing a load test with 10.000 (5.000 users for each scenario) concurrent users, when the number of active users is 8.000 we can see that the mistakes made and failed requests begin. We can also see the bottleneck in the CPU.

Capture of the performance monitor of our computer and test execution console:



Capture of Gatling report:



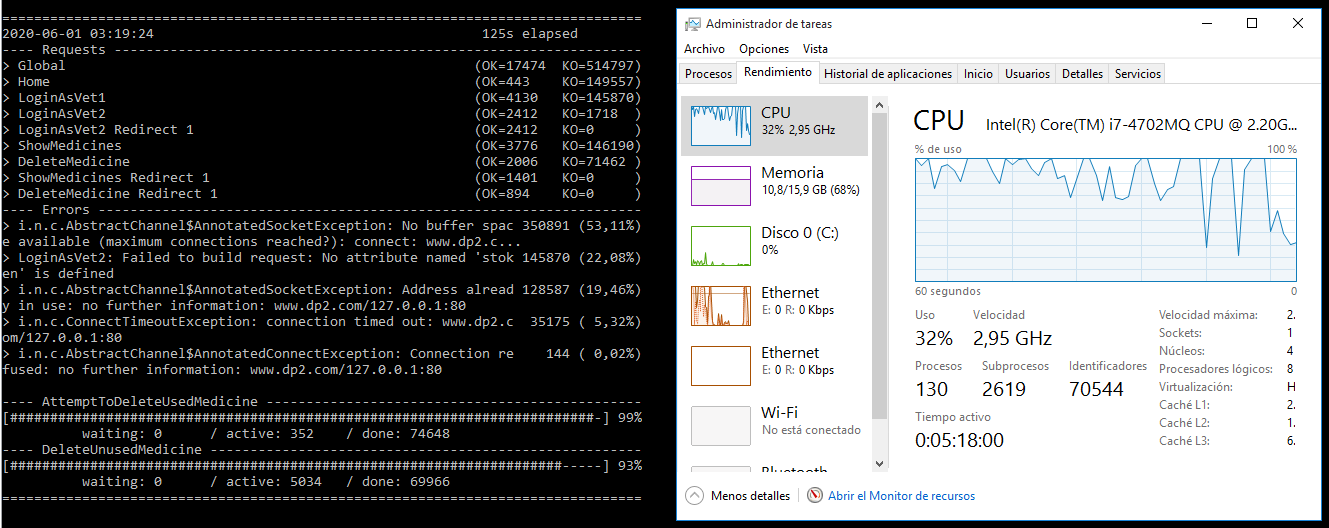
**[US17] Edit or delete the medicines in the system**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **150k**

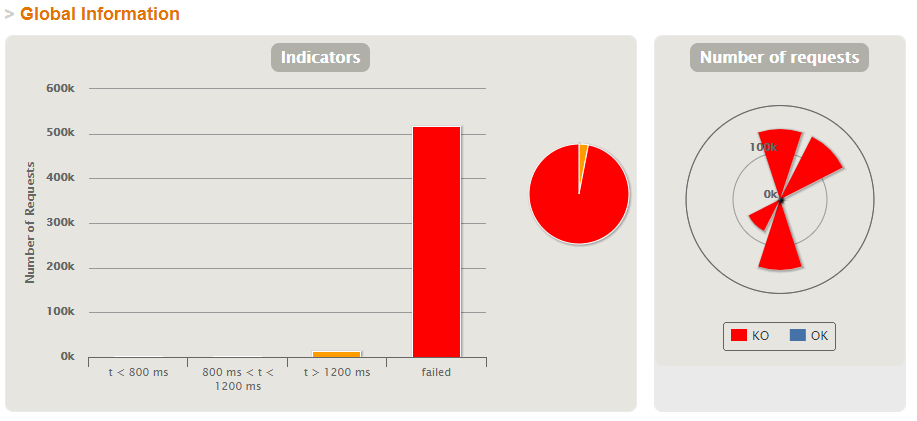
Evidences:

When performing a stress test with 150.000 (75.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

Capture of the performance monitor of our computer and the test execution console:



Capture of Gatling report:

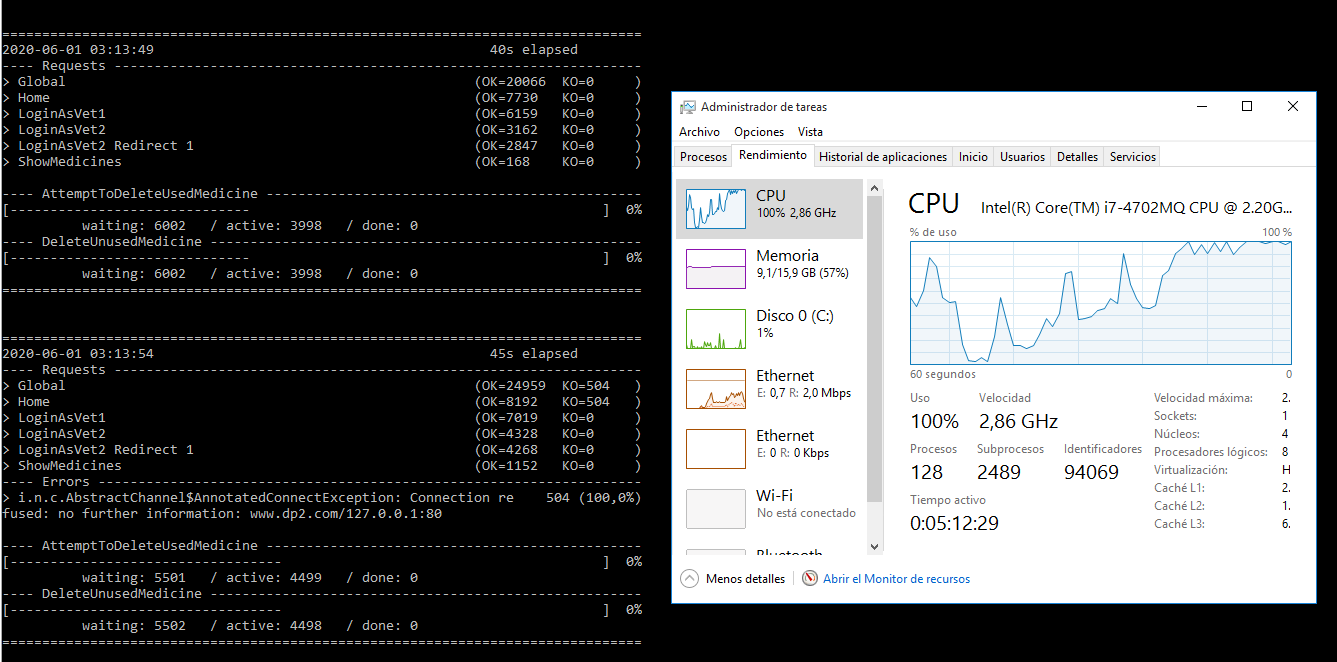


1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **8.600**

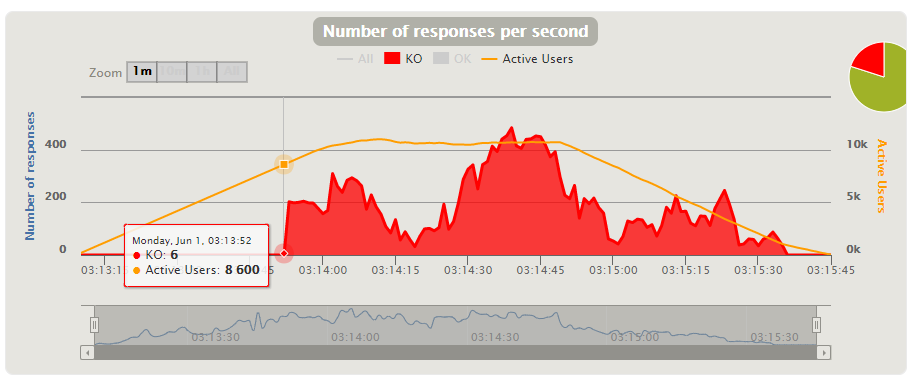
Evidences:

Performing a load test with 10.000 (5.000 users for each scenario) concurrent users, when the number of active users is 8.600 we can see that the mistakes made and failed requests begin. We can also see the bottleneck in the CPU.

Capture of the performance monitor of our computer and test execution console:



Capture of Gatling report:

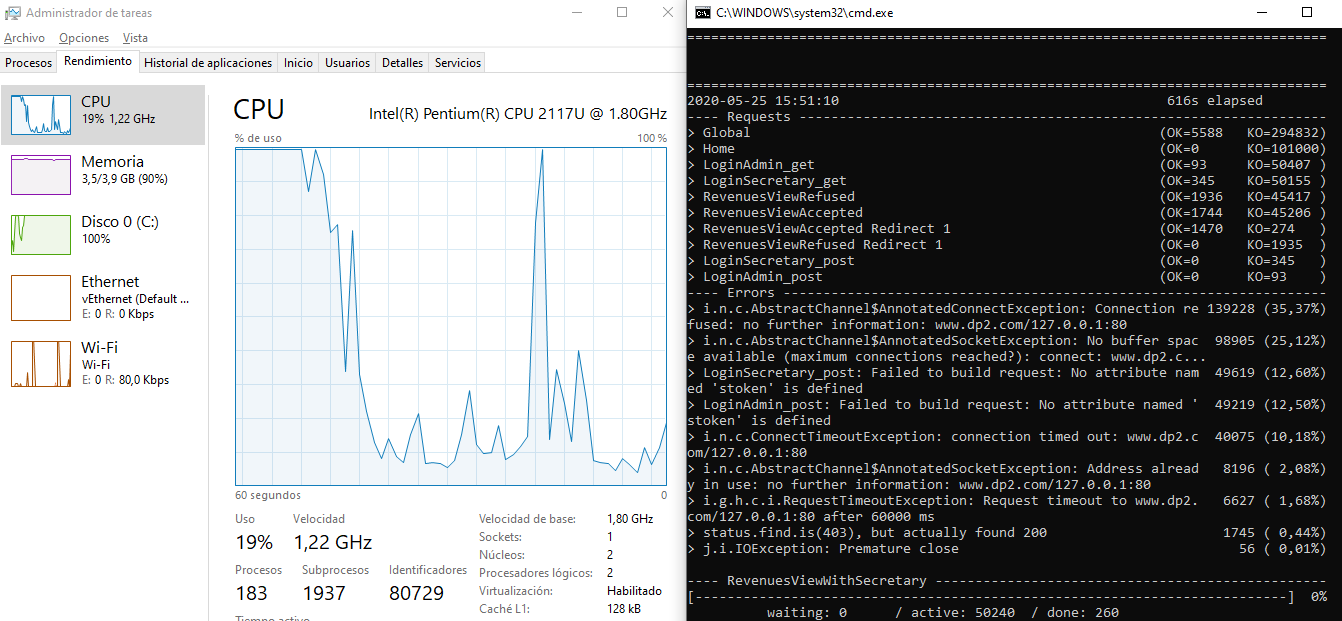


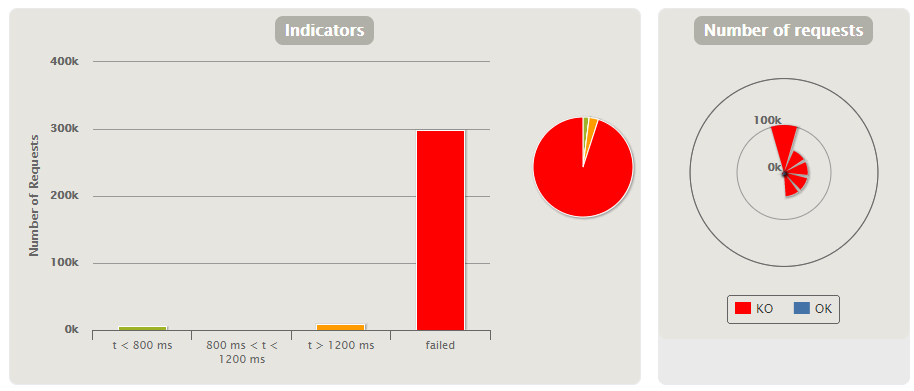
**[US18] View revenue by month**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **101.000 in 10**

Evidences:

When doing a stress test with 101.000 (50.500 users for each scenario) current users, we can see that most requests are failed and the CPU has a bottleneck. So this is the minimum number that my system not supported because with this users there is bad performance.

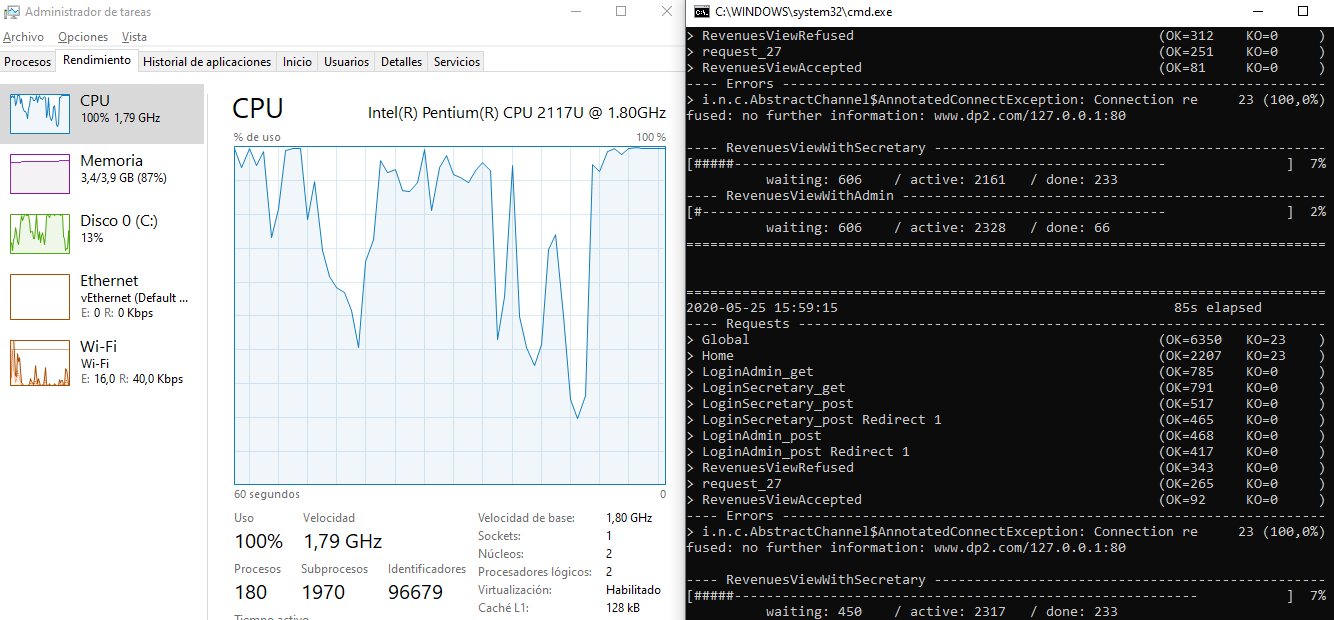
* Capture of the performance monitor of my computer and test execution console:
* Capture of Gatling report:

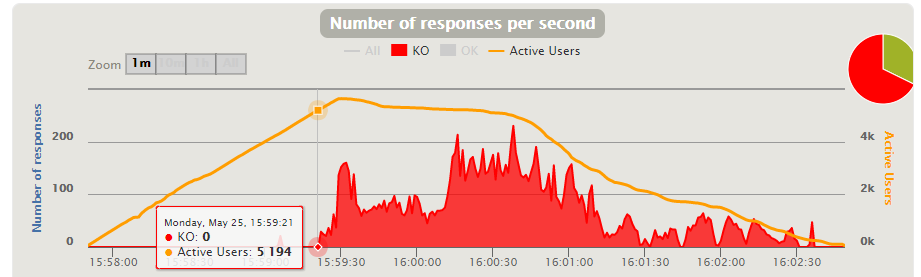


1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **4.800 in 100**

Evidences:

Doing a load test with 6.000 (3.000 users for each scenario) current users, when the number of active users is 4.800 we can see that the mistakes made and failed requests begin. Also can see the bottleneck in the CPU.

* Capture of the performance monitor of my computer and test execution console:
* Capture of Gatling report:



**[US19] See all the characteristics of visits already made**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **X**

Evidences:

aa

Capture of the performance monitor of my computer and test execution console:

Capture of Gatling report:

1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **X**

Evidences:

aa

Capture of the performance monitor of my computer and test execution console:

Capture of Gatling report:

**[US20] Add new types of visit**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **X**

Evidences:

aa

Capture of the performance monitor of my computer and test execution console:

Capture of Gatling report:

1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **X**

Evidences:

aa

Capture of the performance monitor of my computer and test execution console:

Capture of Gatling report:

**[US21] Edit types of visit**

1. **Stress Test** 🡪 Minimum number that is not supported by our system 🡪 **X**

Evidences:

aa

Capture of the performance monitor of my computer and test execution console:

Capture of Gatling report:

1. **Load Test** 🡪 Maximum number that is supported with good performance 🡪 **X**

Evidences:

aa

Capture of the performance monitor of my computer and test execution console:

Capture of Gatling report:

**MAXIMUM TOTAL SYSTEM PERFORMANCE**:

X