

# PERFORMANCE REPORT

First, we report the results of the stress test (minimum number of concurrent users that causes the system to fail) and the results of the load test (maximum number of concurrent users supported with good performance) for each of the 22 user stories. At the end of the document we mention the maximum total system performance.

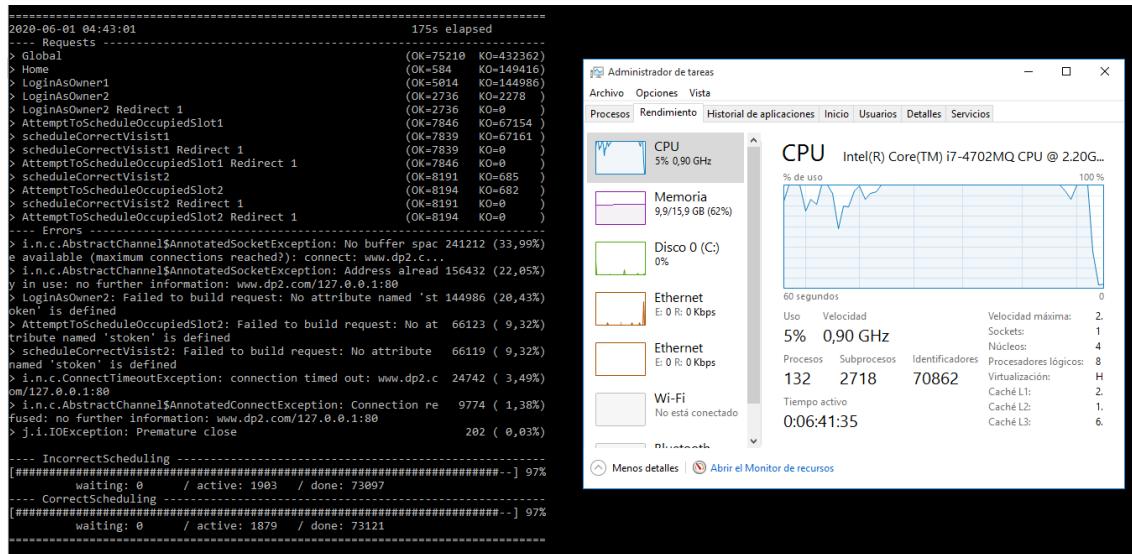
## [US1] Schedule an appointment online

A) Stress Test → Minimum number that is not supported by our system → **150k**

### Evidences:

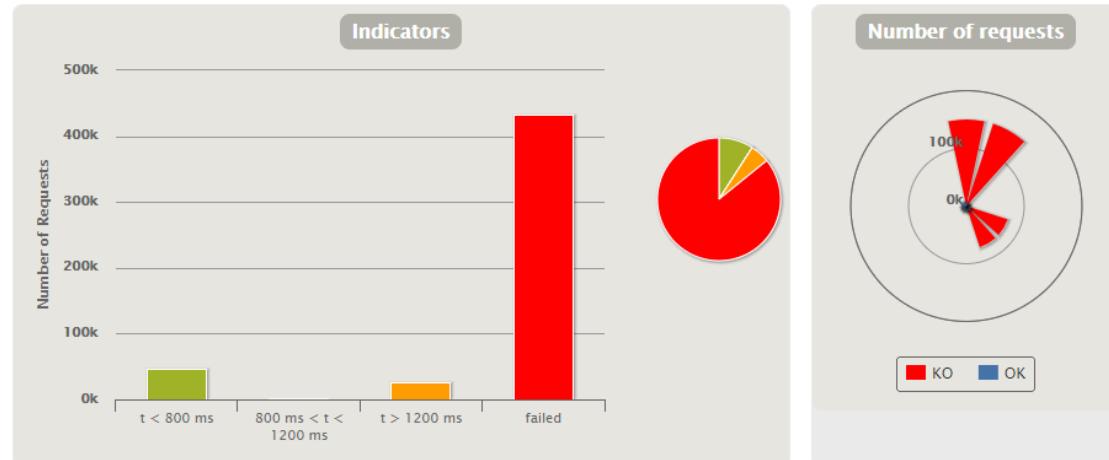
When performing a stress test with 150.000 (75.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

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



Capture of the Gatling report:

## > Global Information

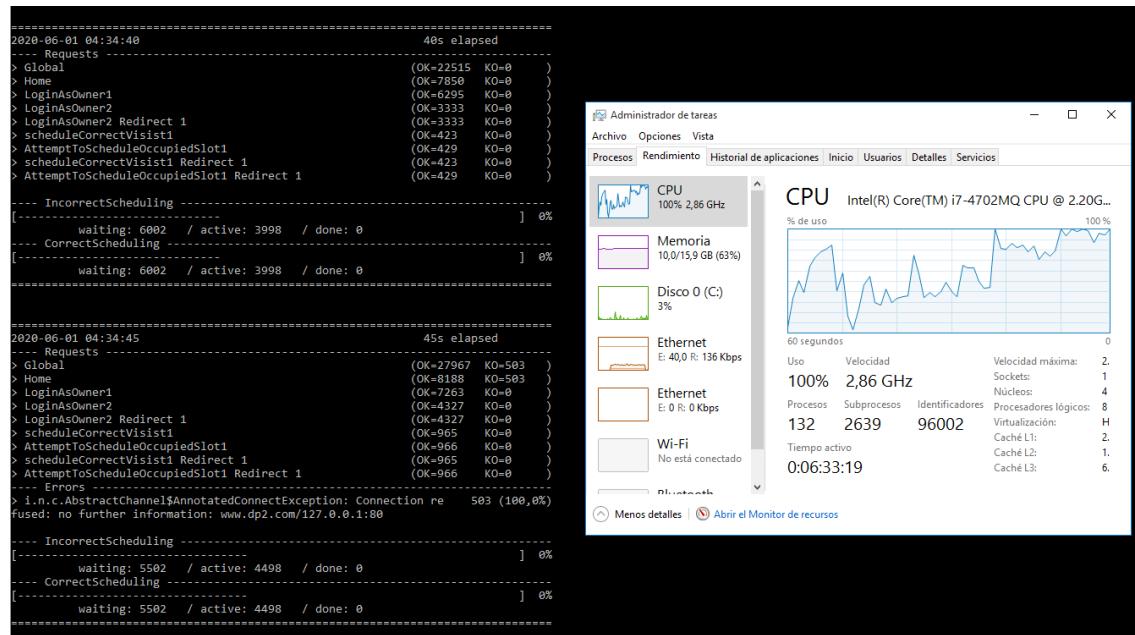


B) Load Test → Maximum number that is supported with good performance → **8800**

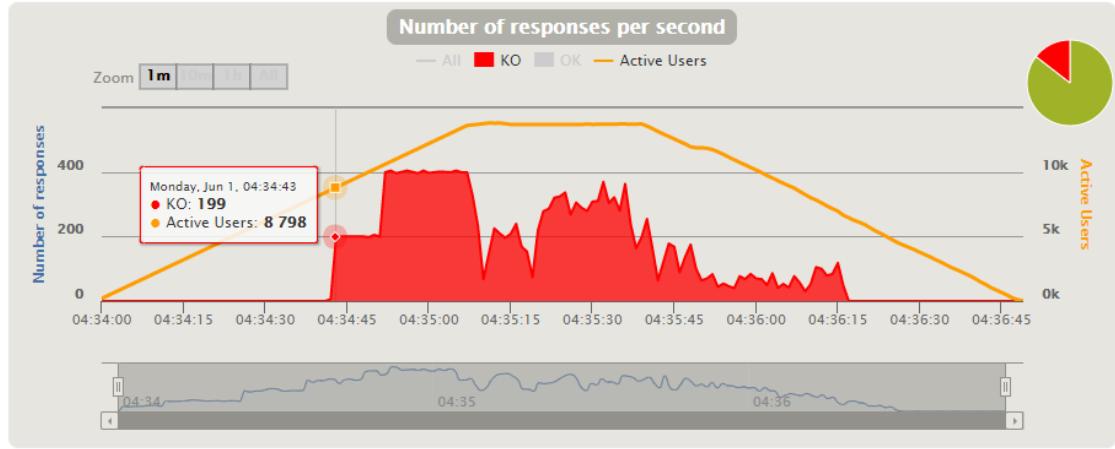
### Evidences:

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

A) 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:

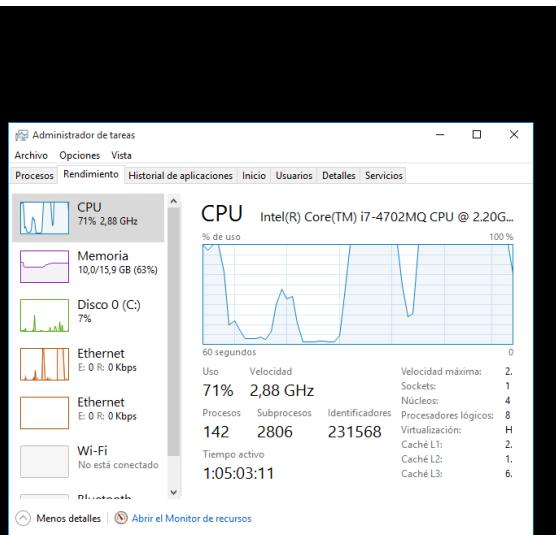
```
---- IncorrectScheduling -----
[##### waiting: 0 / active: 4729 / done: 70271 ] 93%
---- CorrectScheduling -----
[##### waiting: 0 / active: 75000 / done: 0 ] 0%
=====

2020-05-30 18:06:53 25s elapsed
---- Requests -----
> Global (OK=2552 KO=159170)
> scheduleCorrectVisist1 (OK=1047 KO=73479 )
> Home (OK=1084 KO=73319 )
> LoginAsOwner1 (OK=0 KO=12372 )
> scheduleCorrectVisist1 Redirect 1 (OK=415 KO=0 )
> request_1 (OK=6 KO=0 )
> Errors
> i.n.c.AbstractChannel$AnnotatedSocketException: No buffer spac 127794 (89,29%)
available (maximum connections reached?): connect: www.dp2...
> i.n.c.ConnectTimeoutException: connection timed out: www.dp2.c 16276 (10,23%)
on/127.0.0.1:80
> i.n.c.AbstractChannel$AnnotatedSocketException: Address alread 15100 ( 9,49%)
y in use: no further information: www.dp2.com/127.0.0.1:80

---- IncorrectScheduling -----
[##### waiting: 0 / active: 1515 / done: 73485 ] 97%
---- CorrectScheduling -----
[##### waiting: 0 / active: 75000 / done: 0 ] 0%
=====

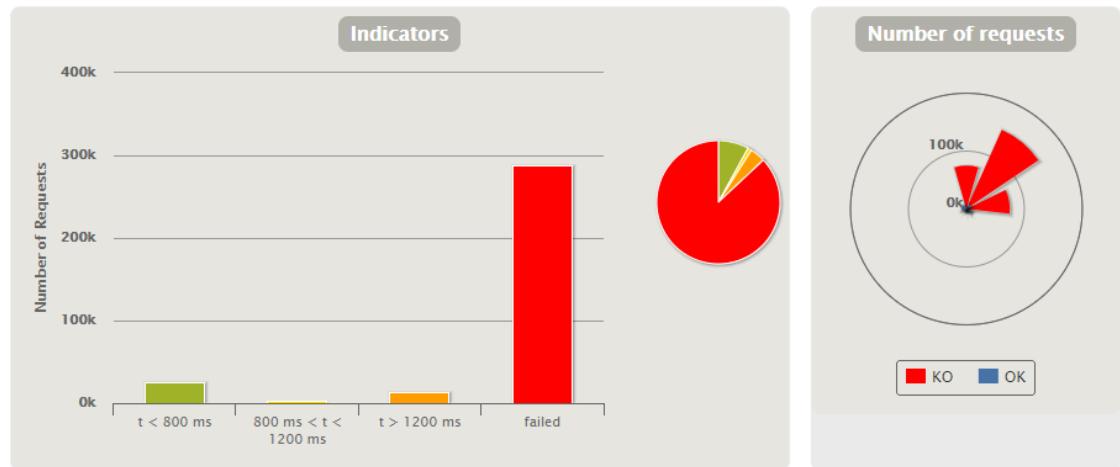
2020-05-30 18:06:58 30s elapsed
---- Requests -----
> Global (OK=3298 KO=201894)
> scheduleCorrectVisist1 (OK=1088 KO=73759 )
> Home (OK=1134 KO=73593 )
> LoginAsOwner1 (OK=298 KO=54542 )
> scheduleCorrectVisist1 Redirect 1 (OK=569 KO=0 )
> request_1 (OK=209 KO=0 )
> Errors
> i.n.c.AbstractChannel$AnnotatedSocketException: No buffer spac 169050 (83,73%)
available (maximum connections reached?): connect: www.dp2...
> i.n.c.ConnectTimeoutException: connection timed out: www.dp2.c 16830 ( 8,34%)
on/127.0.0.1:80
> i.n.c.AbstractChannel$AnnotatedSocketException: Address alread 16014 ( 7,93%)
y in use: no further information: www.dp2.com/127.0.0.1:80

---- IncorrectScheduling -----
[##### waiting: 0 / active: 1033 / done: 73967 ] 98%
---- CorrectScheduling -----
[##### waiting: 0 / active: 75000 / done: 0 ] 0%
```



### Capture of Gatling report:

## > Global Information

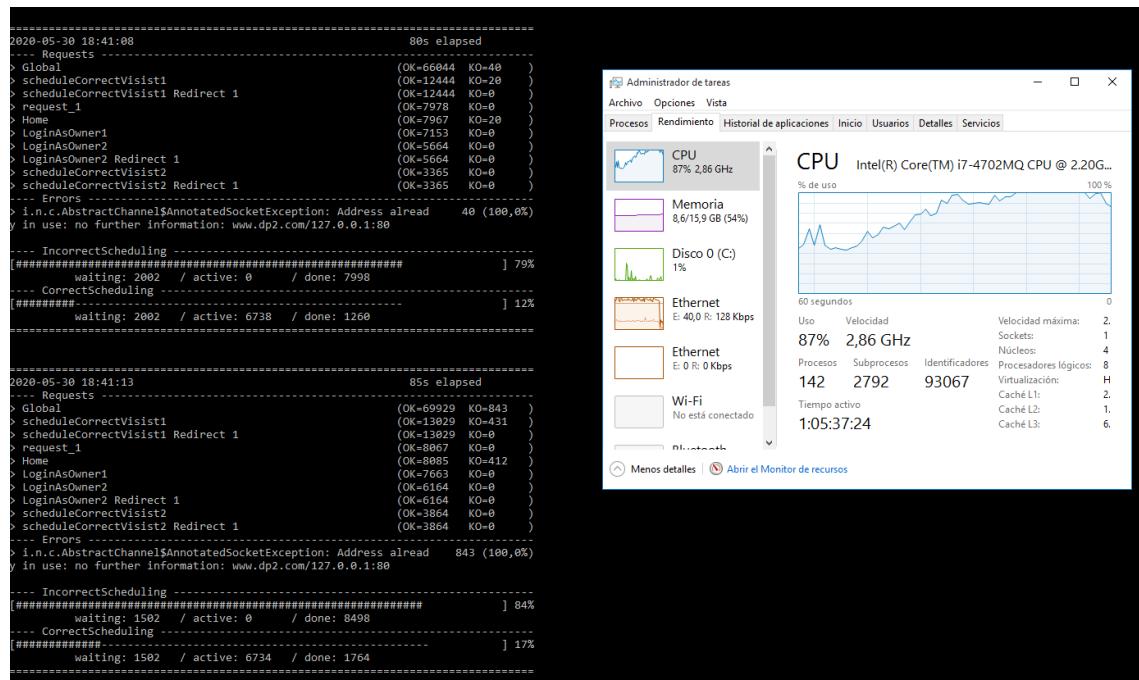


## B) Load Test → Maximum number that is supported with good performance → 6.900

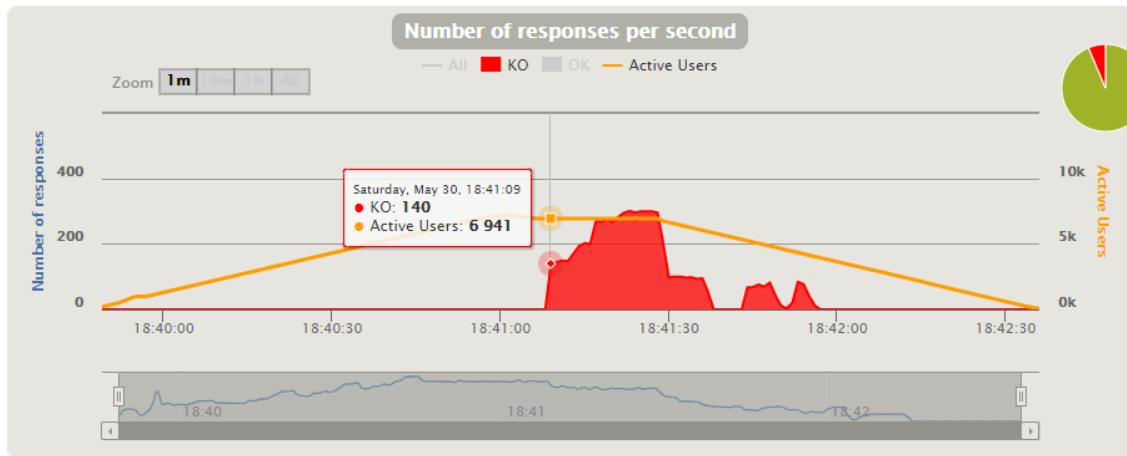
### Evidences:

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:



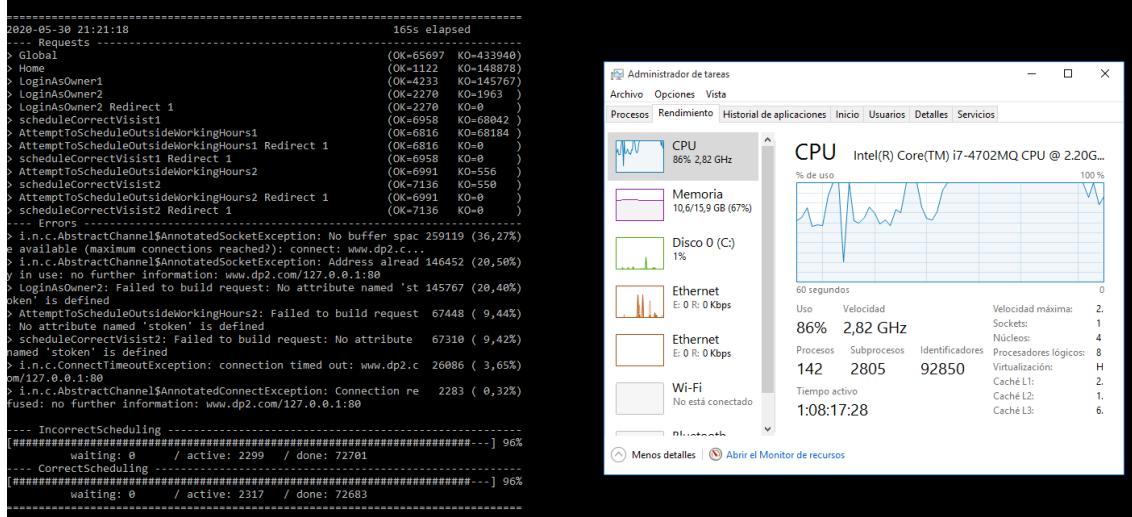
### [US3] Manage appointments automatically

A) Stress Test → Minimum number that is not supported by our system → 150k

Evidences:

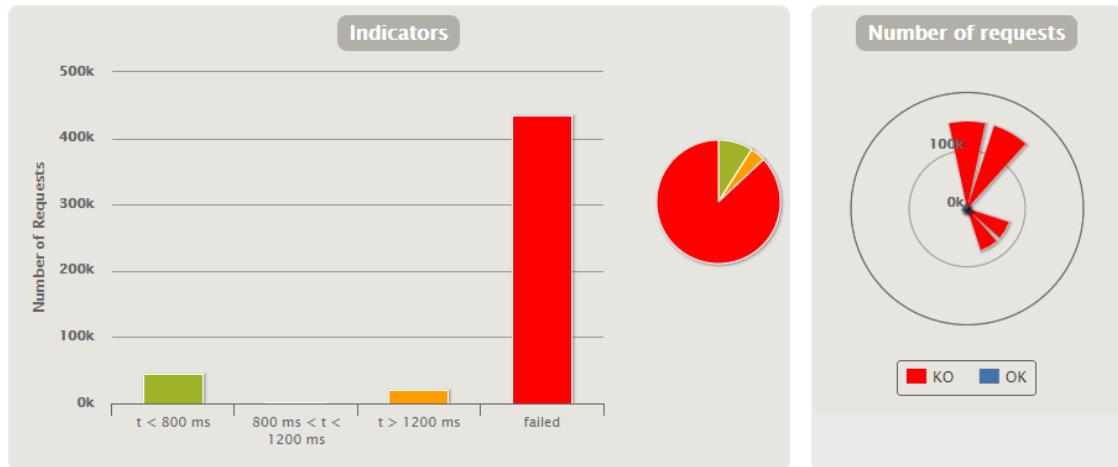
When performing a stress test with 150.000 (75.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

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



Capture of Gatling report:

## > Global Information

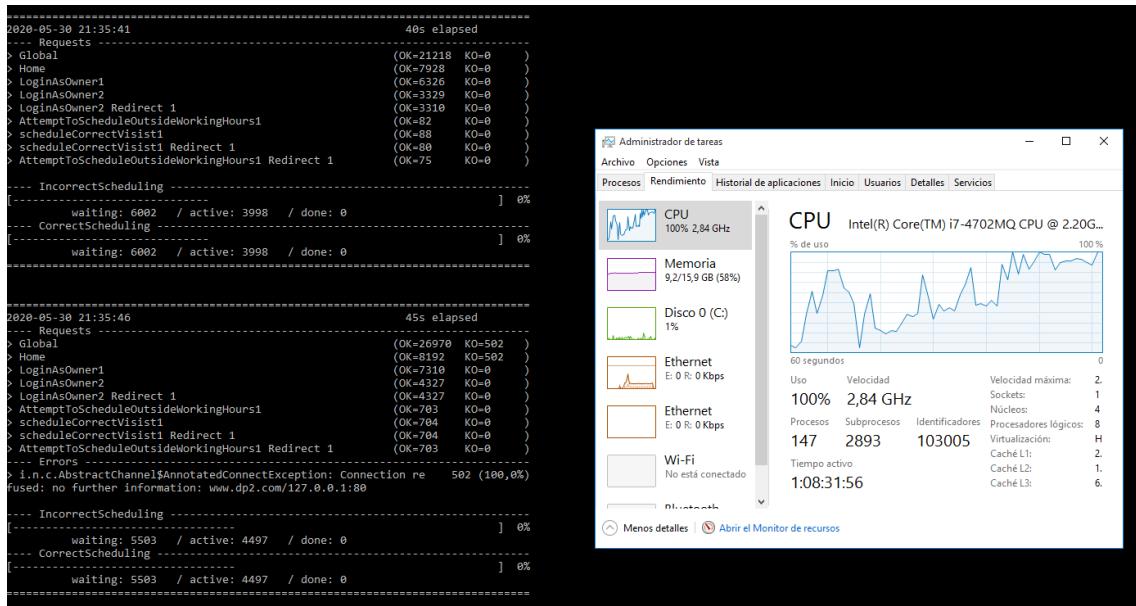


## B) Load Test → Maximum number that is supported with good performance → 8.800

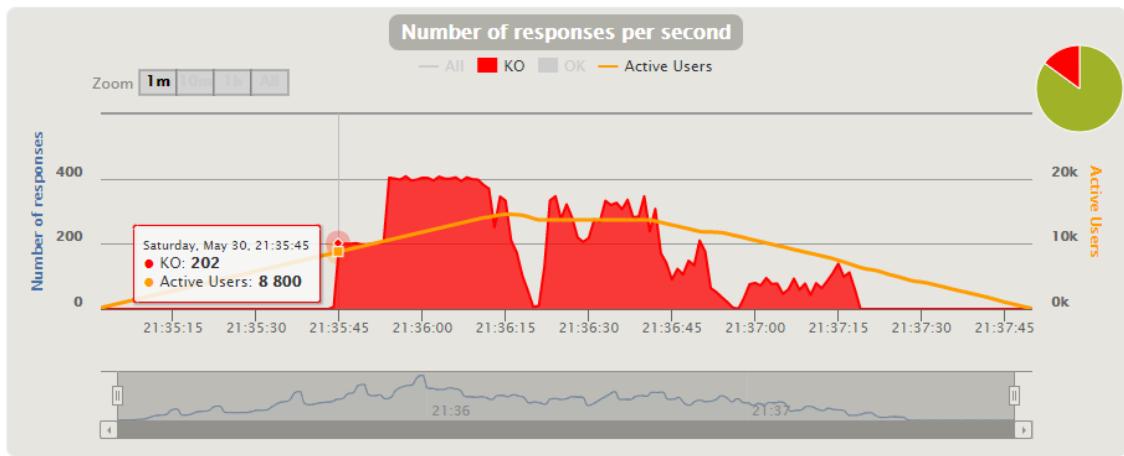
### Evidences:

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:



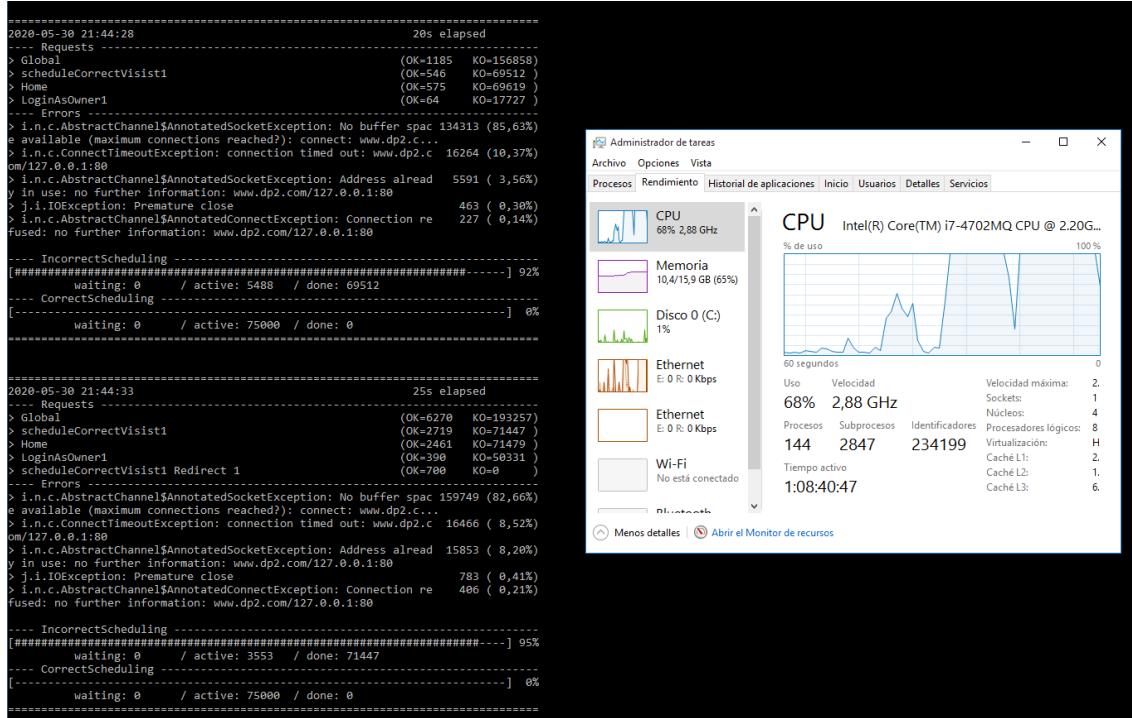
## [US4] Select a type of visit

A) Stress Test → Minimum number that is not supported by our system → 150k

### Evidences:

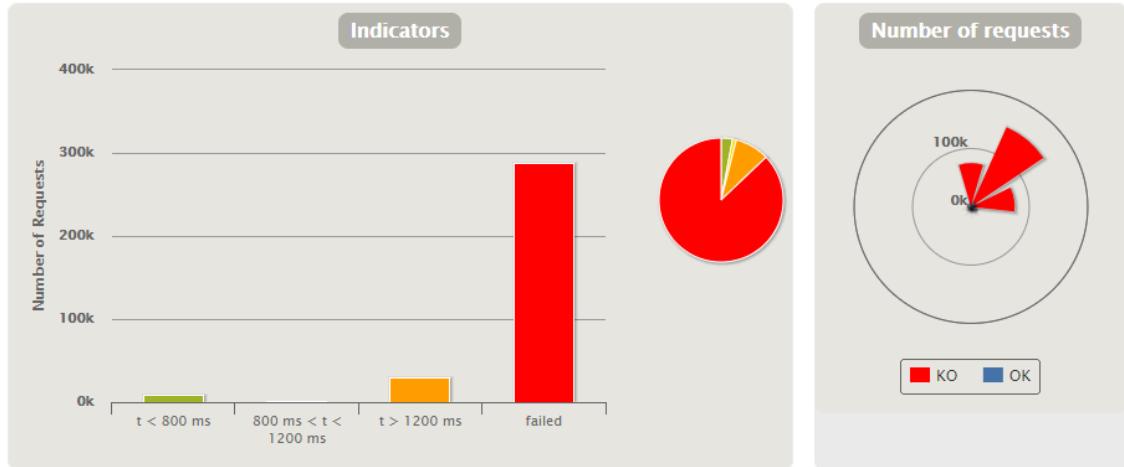
When performing a stress test with 150.000 (75.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

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



Capture of Gatling report:

## > Global Information

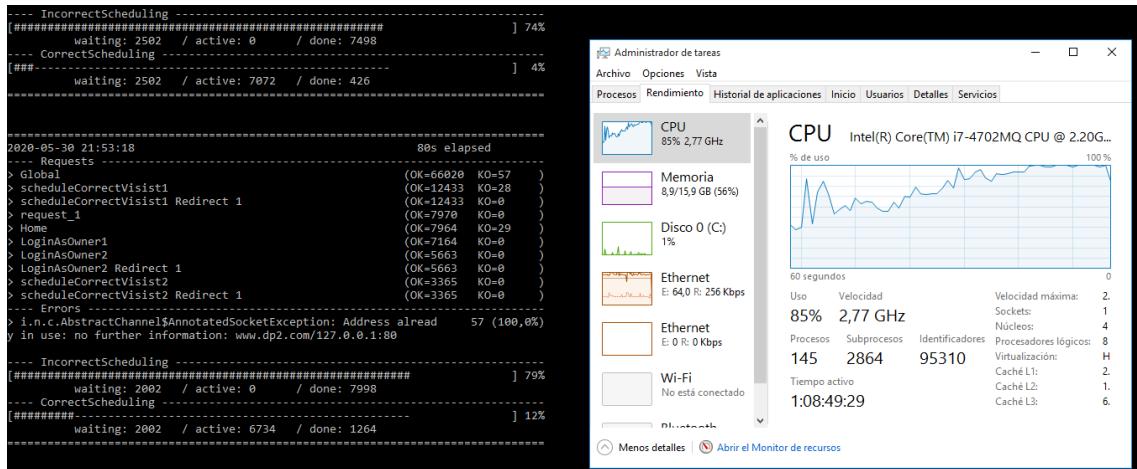


**B) Load Test → Maximum number that is supported with good performance → 6.900**

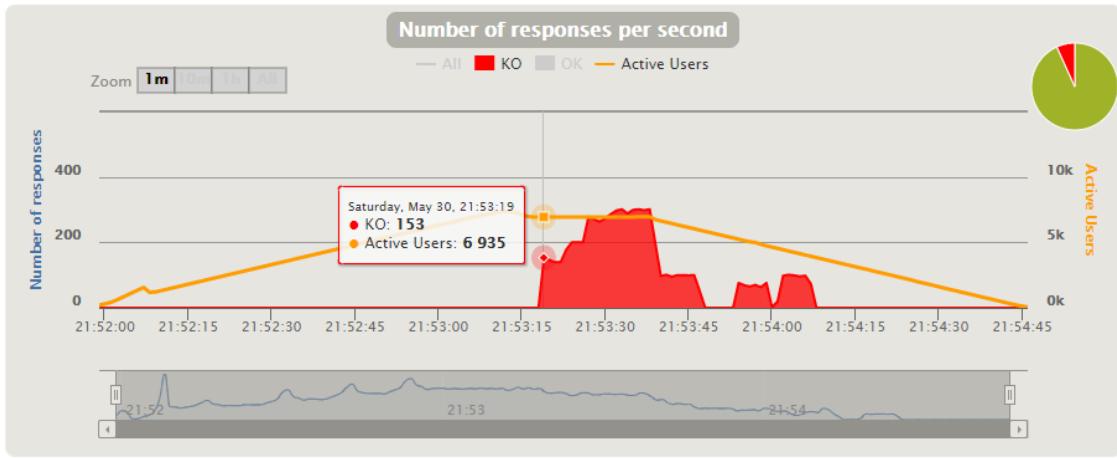
### Evidences:

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:



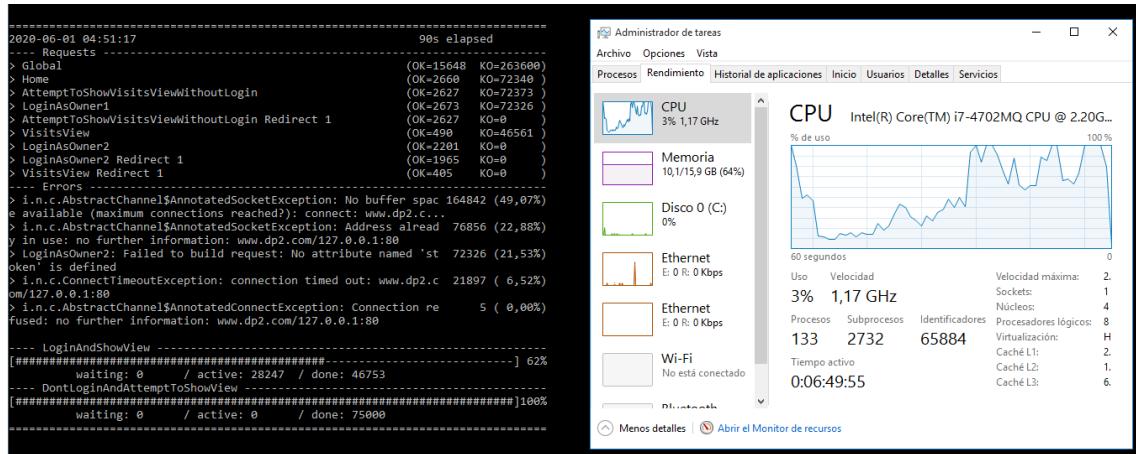
## [US5A] Upcoming visits view (pet owner)

A) 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:

## > Global Information

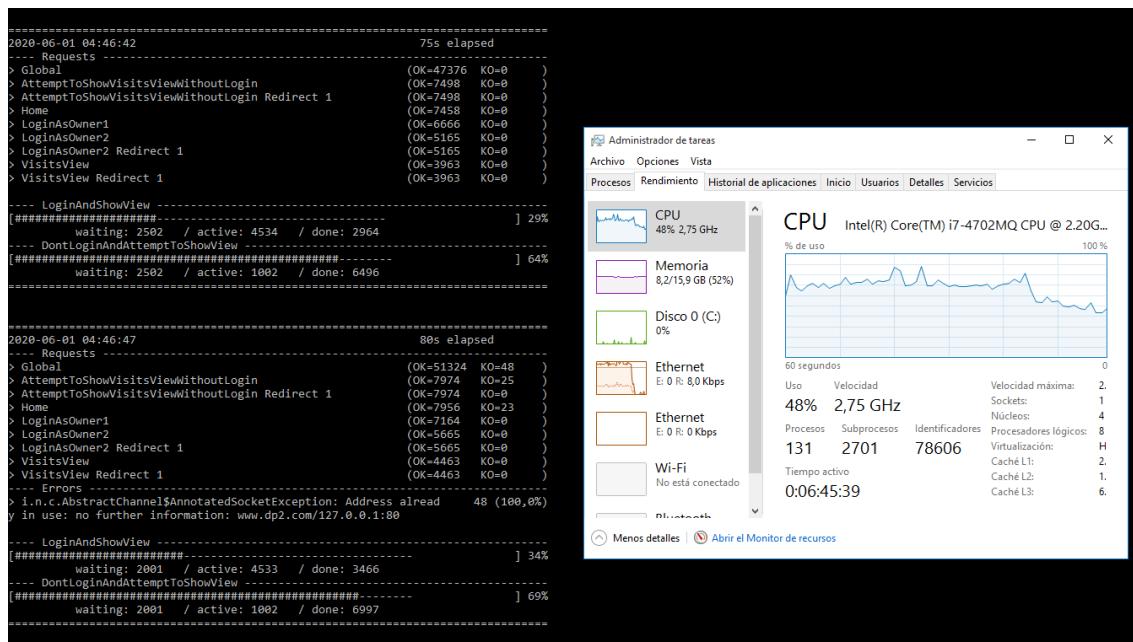


## B) Load Test → Maximum number that is supported with good performance → 5.700

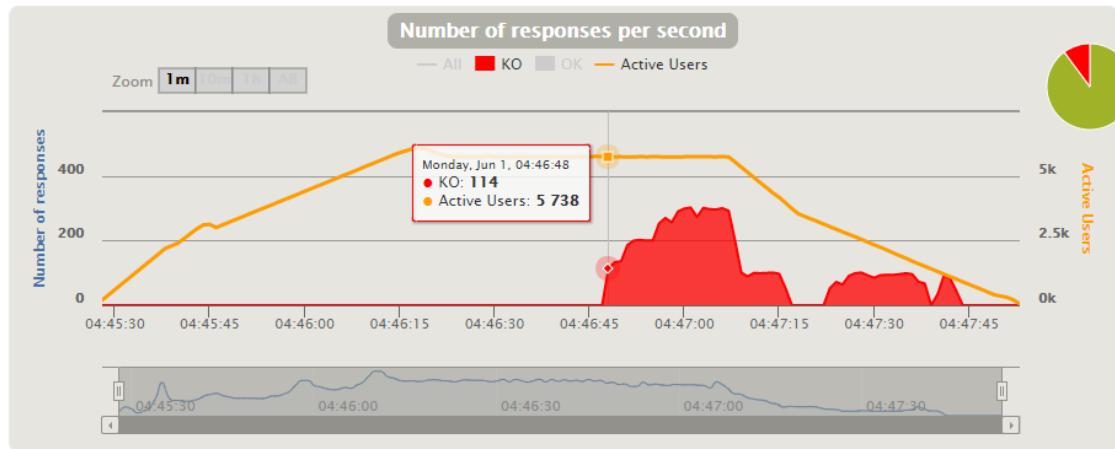
### Evidences:

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:



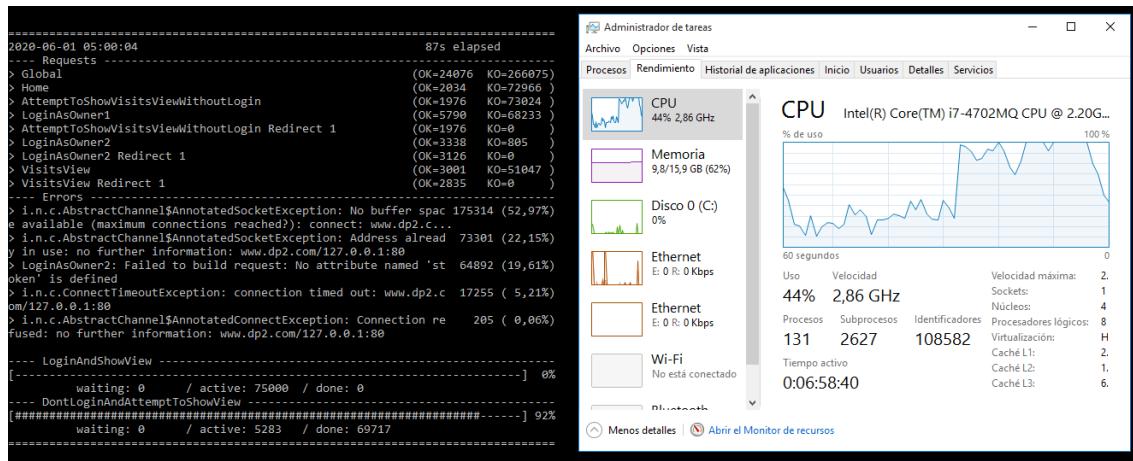
### [US5B] Past visits view (pet owner)

A) Stress Test → Minimum number that is not supported by our system → 150k

#### Evidences:

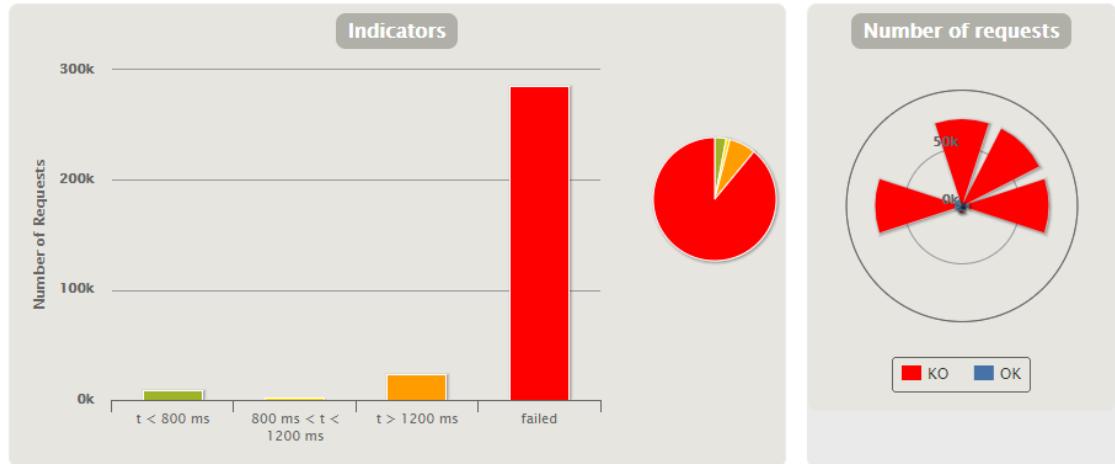
When performing a stress test with 150.000 (75.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

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



Capture of Gatling report:

## > Global Information

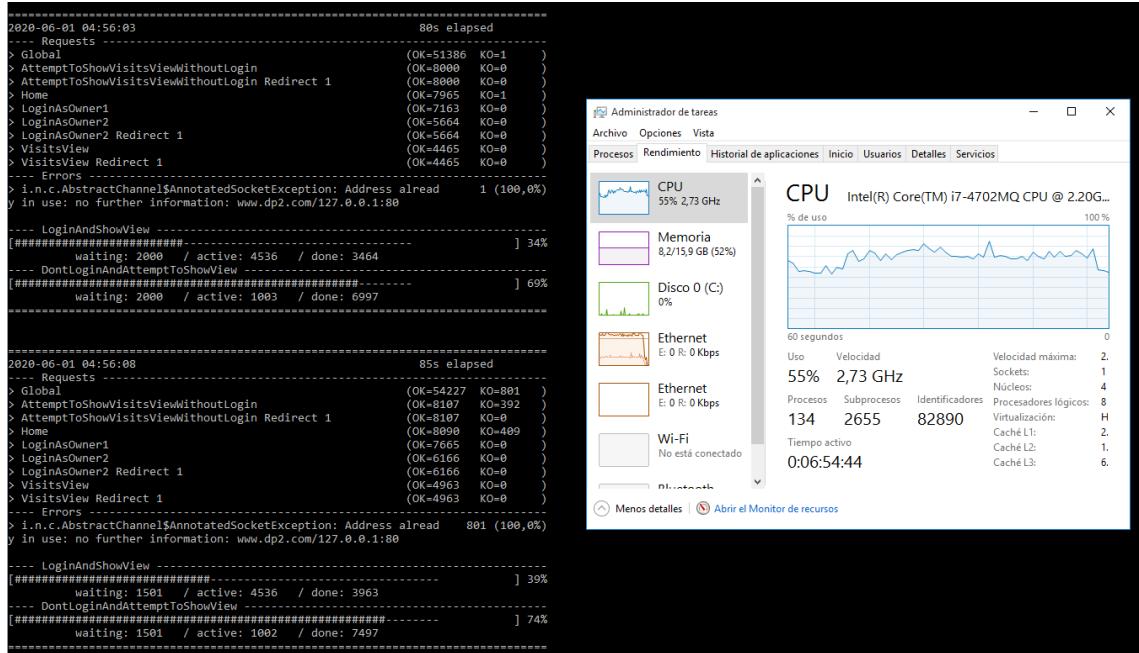


B) Load Test → Maximum number that is supported with good performance → 5.700

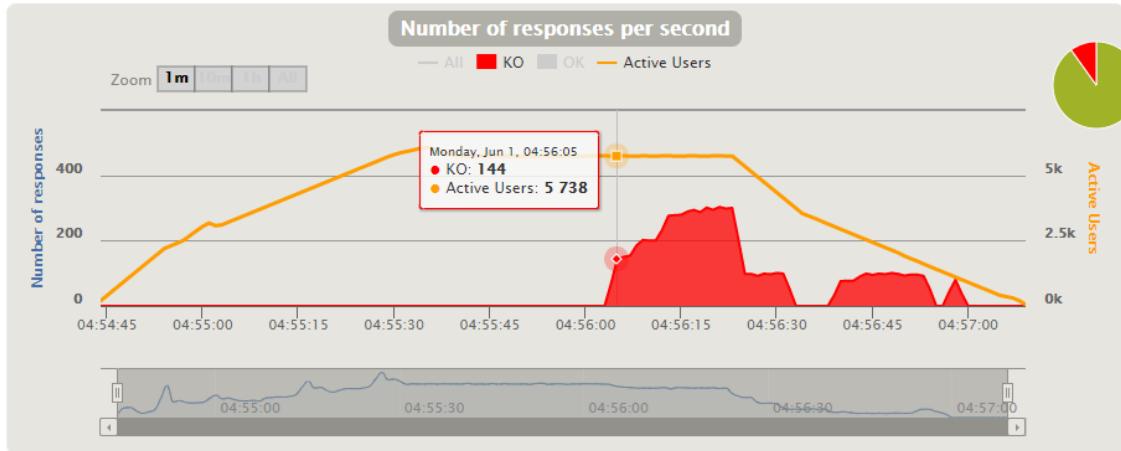
Evidences:

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:



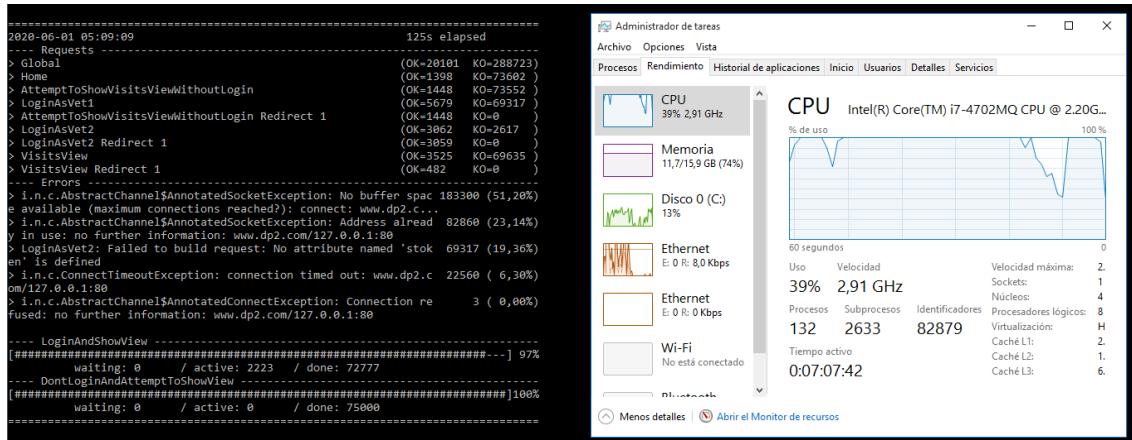
## [US6] Upcoming visits view (vet)

A) Stress Test → Minimum number that is not supported by our system → 150k

### Evidences:

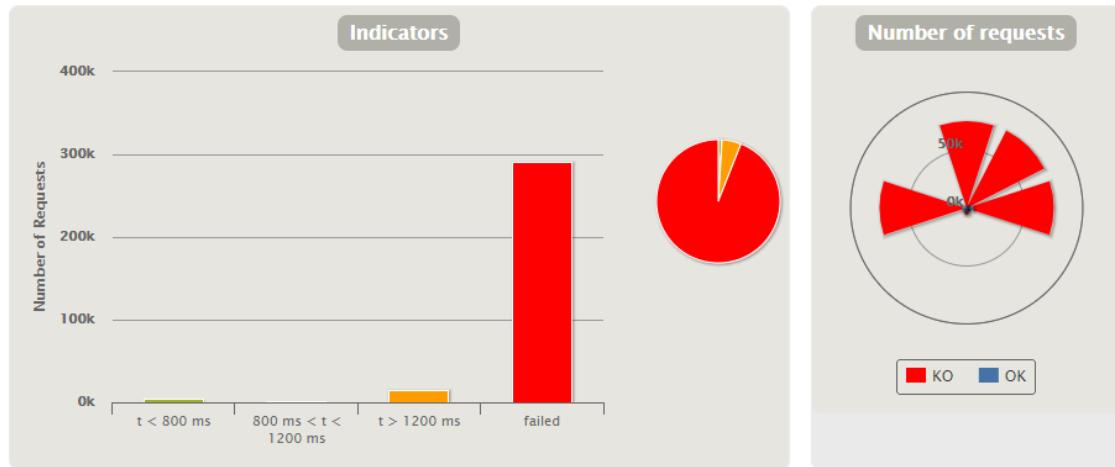
When performing a stress test with 150.000 (75.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

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



### Capture of Gatling report:

## > Global Information

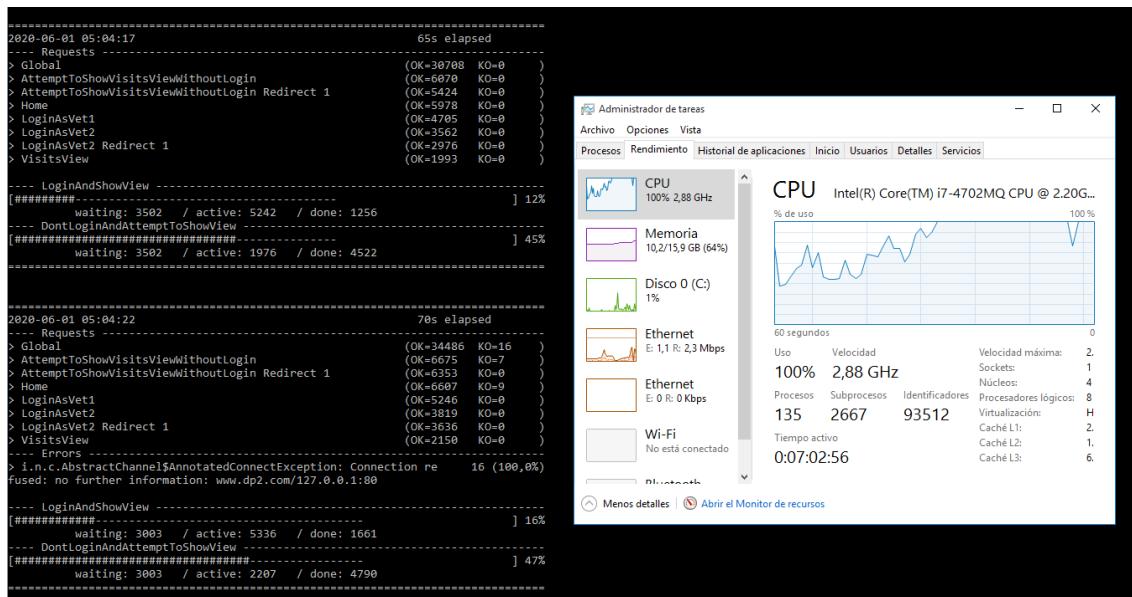


## B) Load Test → Maximum number that is supported with good performance → 7.500

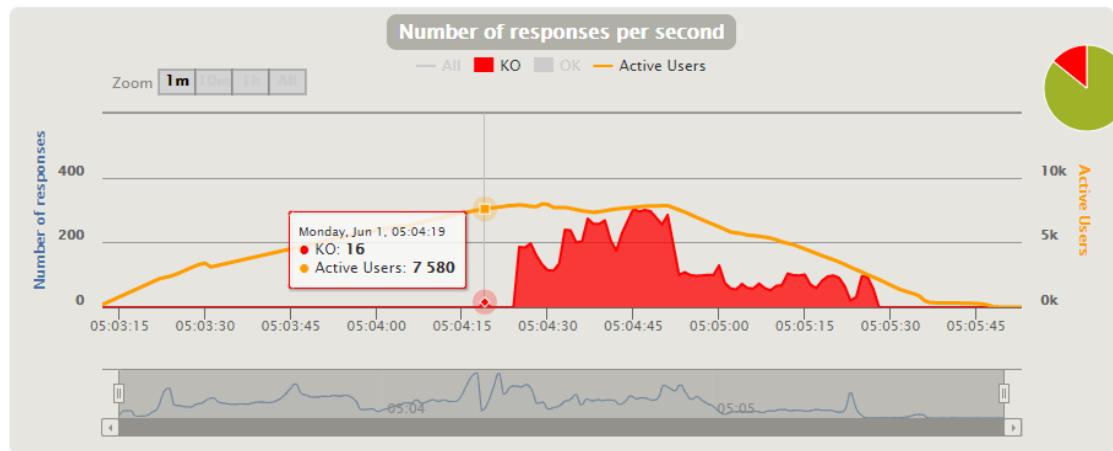
### Evidences:

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:



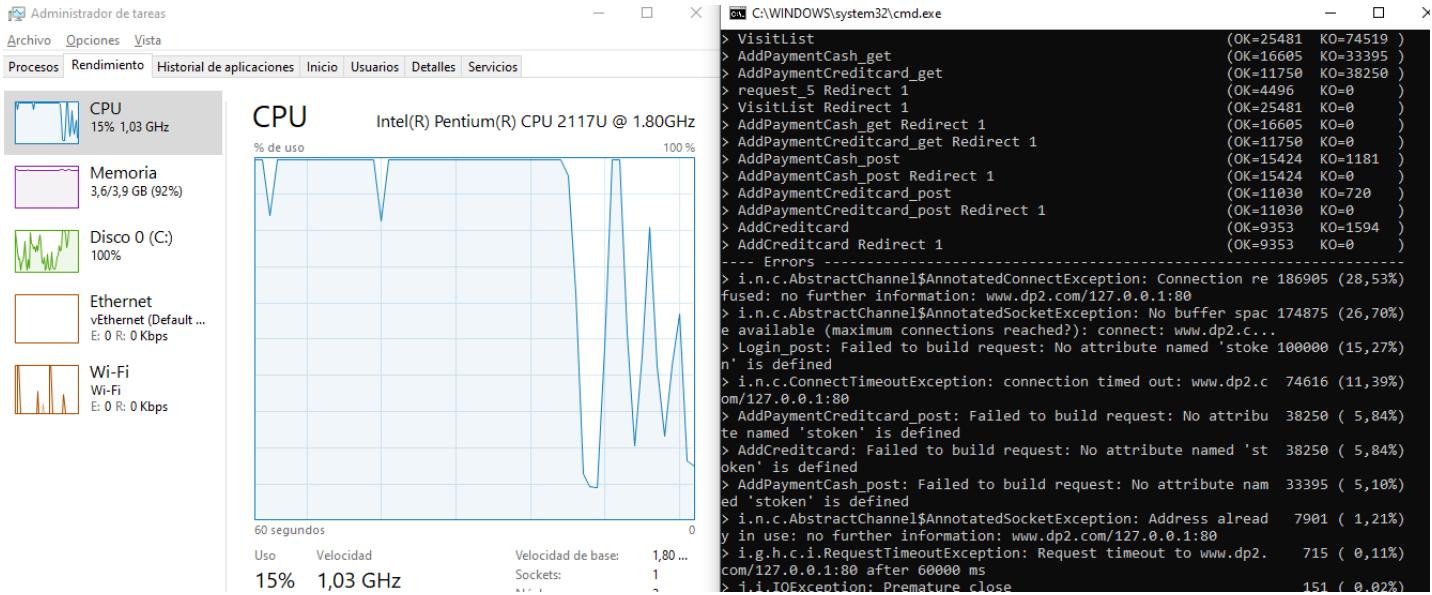
## [US7] Register a payment with credit card or cash

A) Stress Test → Minimum number that is not supported by our system → 100.000 in 10

### Evidences:

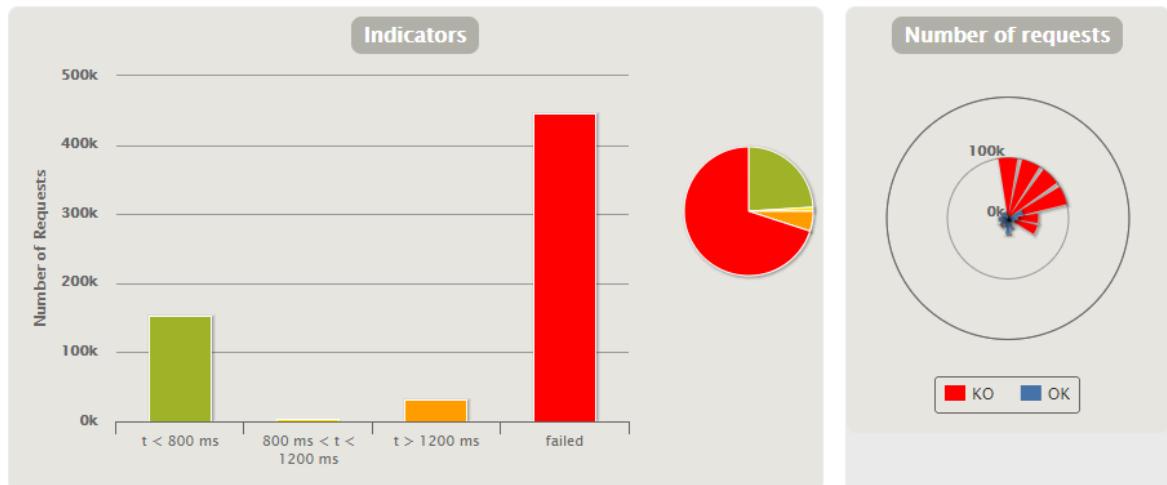
When doing a stress test with 100.000 (50.000 users for each scenario) current users, we can see that most requests are failed and the CPU has a bottleneck. So this is the minimum number that my system not supported because with this users there is bad performance.

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



- Capture of Gatling report:

### > Global Information

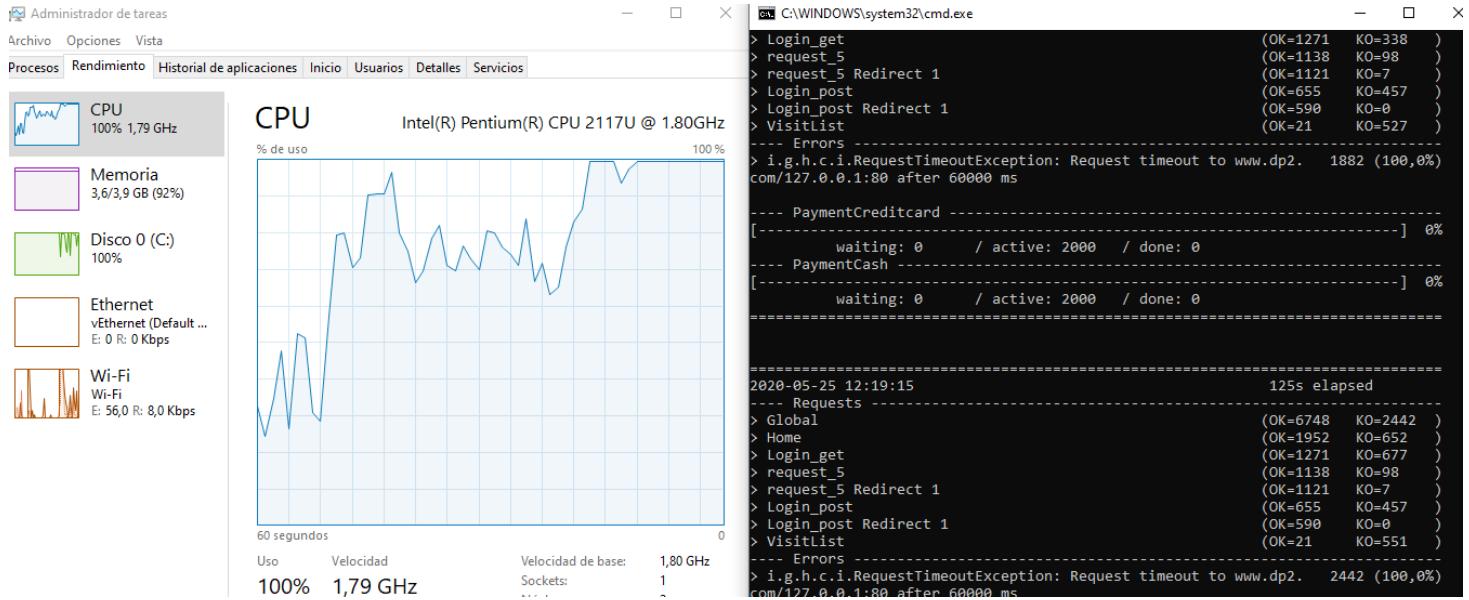


## B) Load Test → Maximum number that is supported with good performance → 4.000 in 100

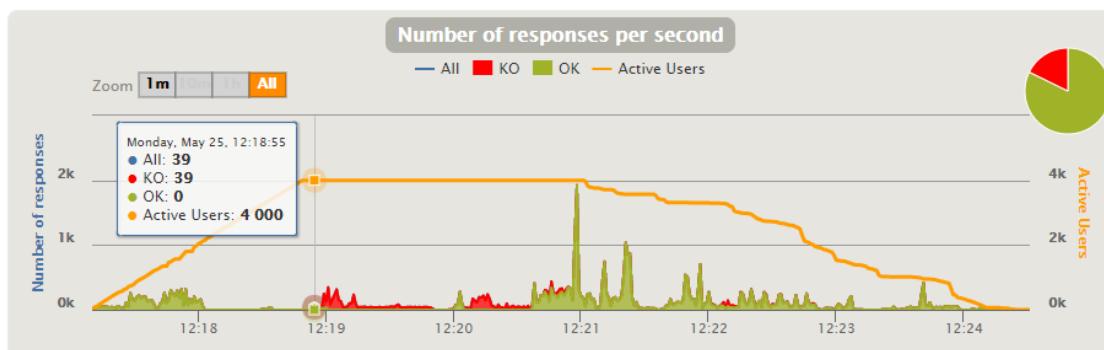
### Evidences:

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:



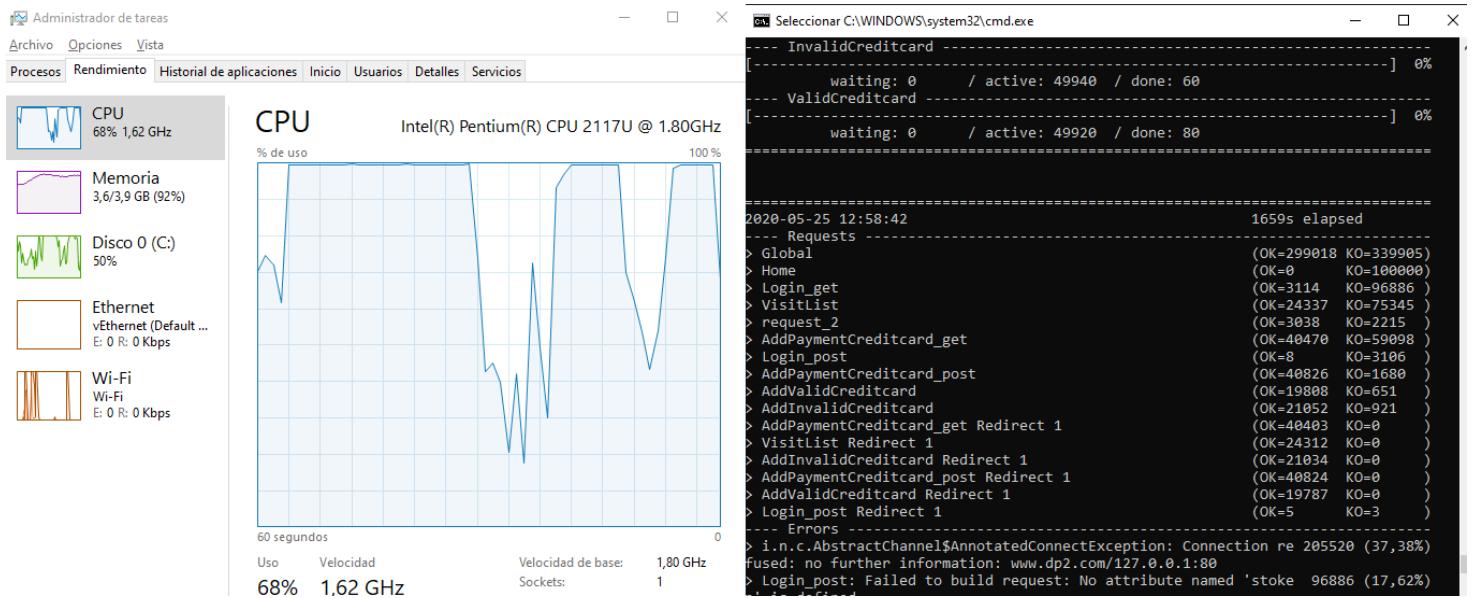
## [US8] Validate credit card

A) Stress Test → Minimum number that is not supported by our system → **100.000 in 10**

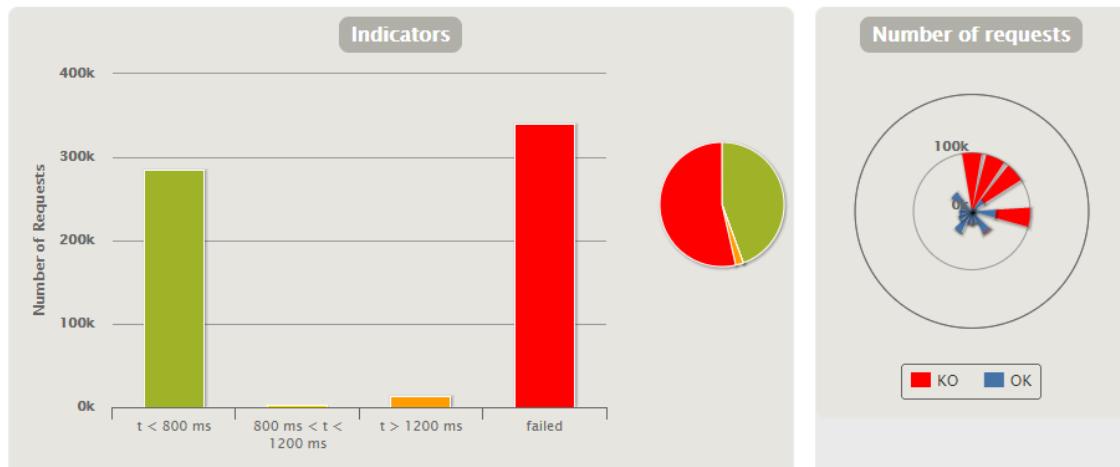
### Evidences:

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:

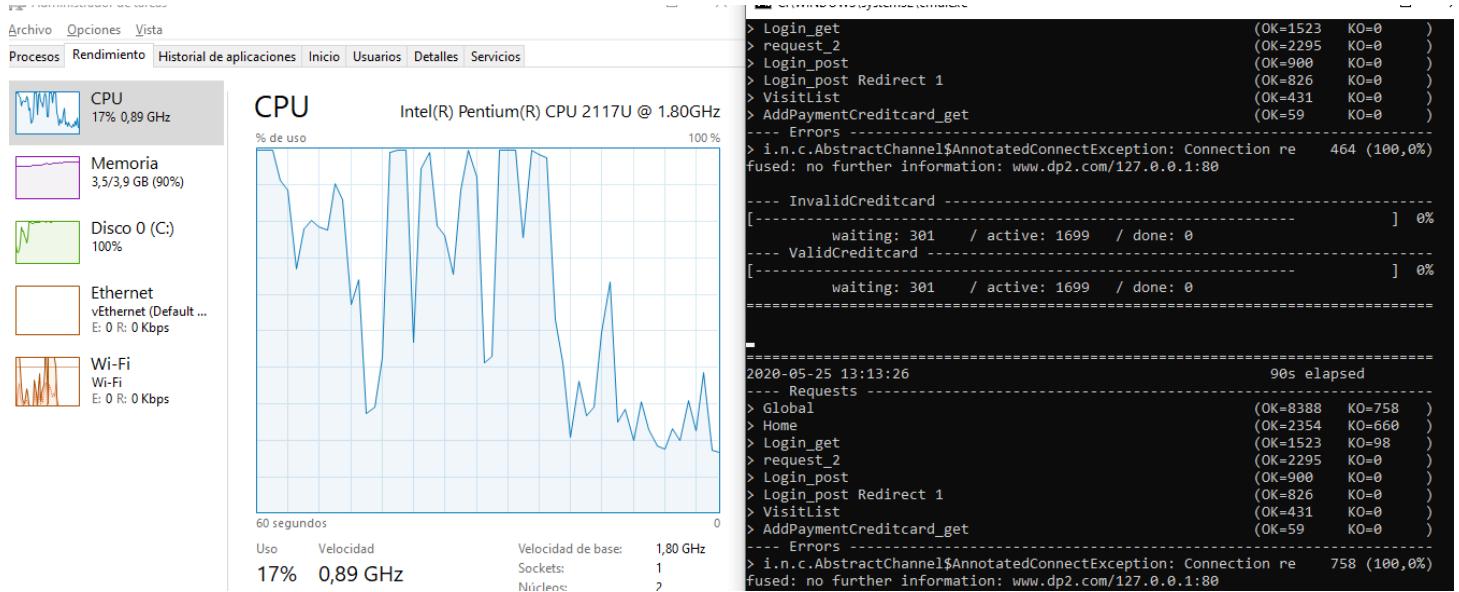


## B) Load Test → Maximum number that is supported with good performance → 3.000 in 100

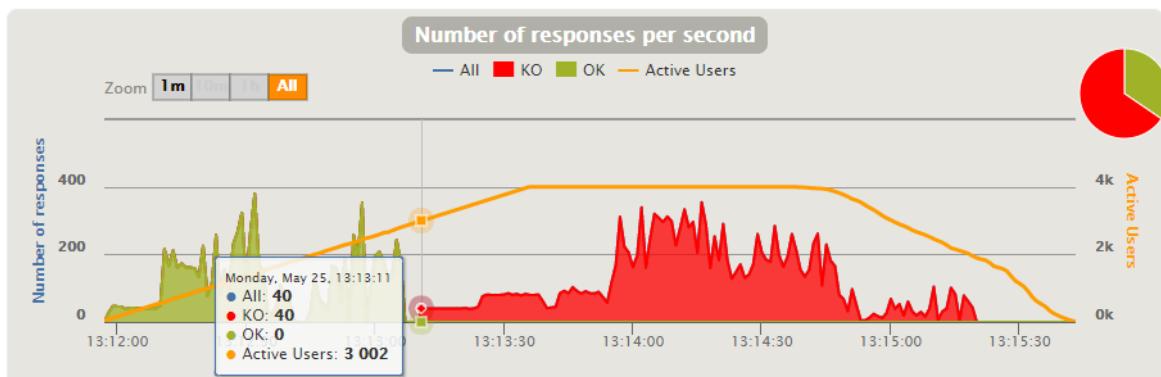
### Evidences:

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:



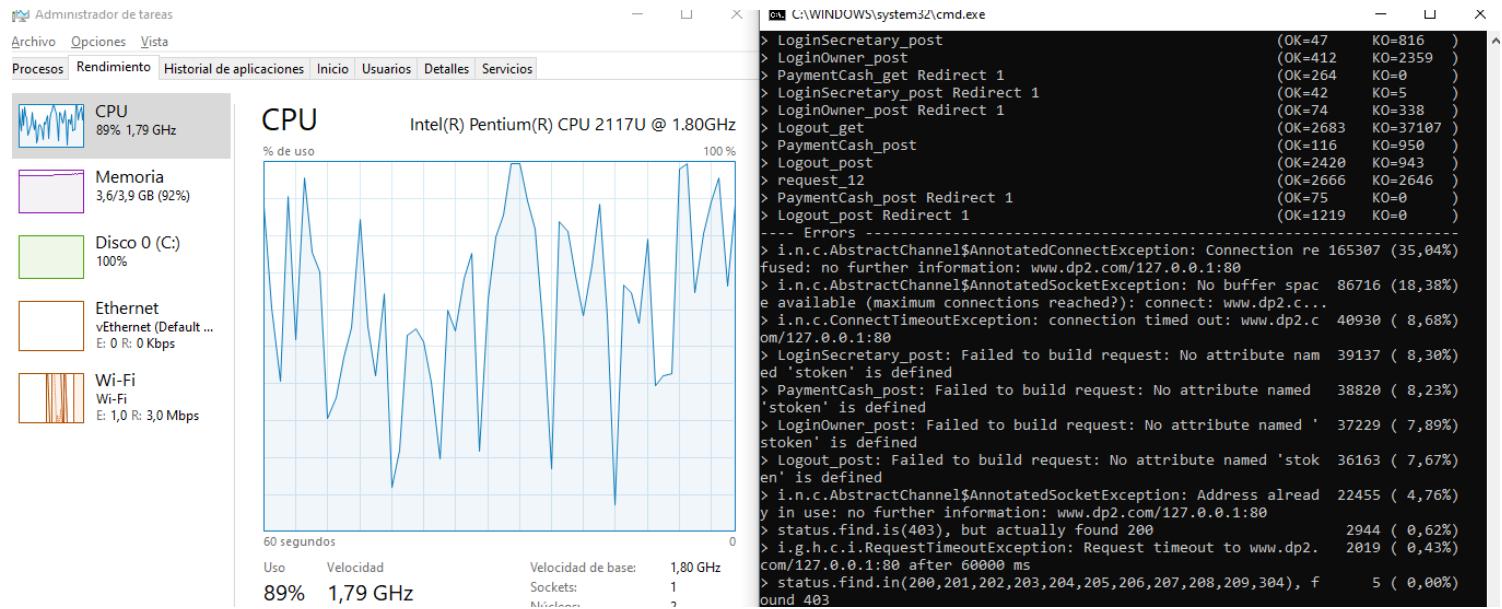
## [US9] Store who registered a payment

A) Stress Test → Minimum number that is not supported by our system → **80.000 in 10**

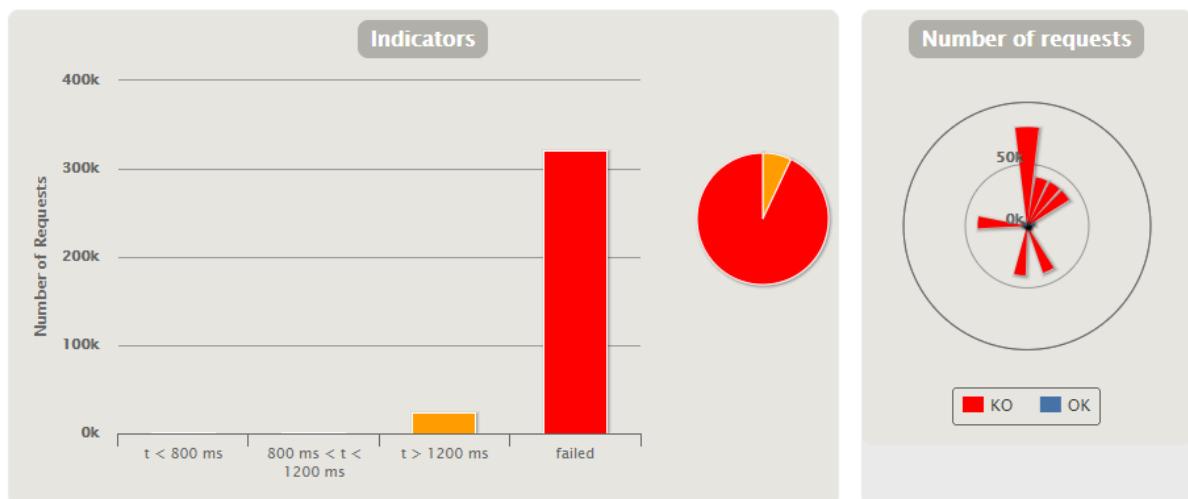
### Evidences:

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:

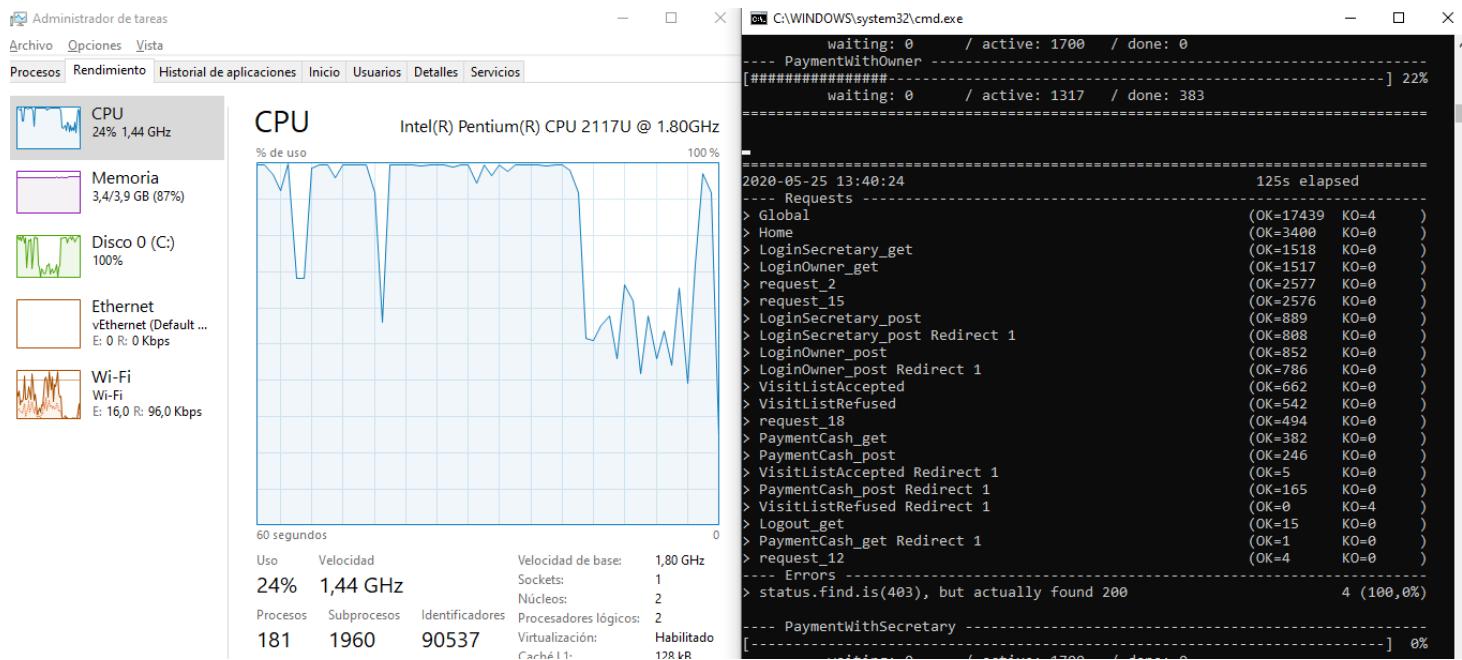


## B) Load Test → Maximum number that is supported with good performance → 3.000 in 100

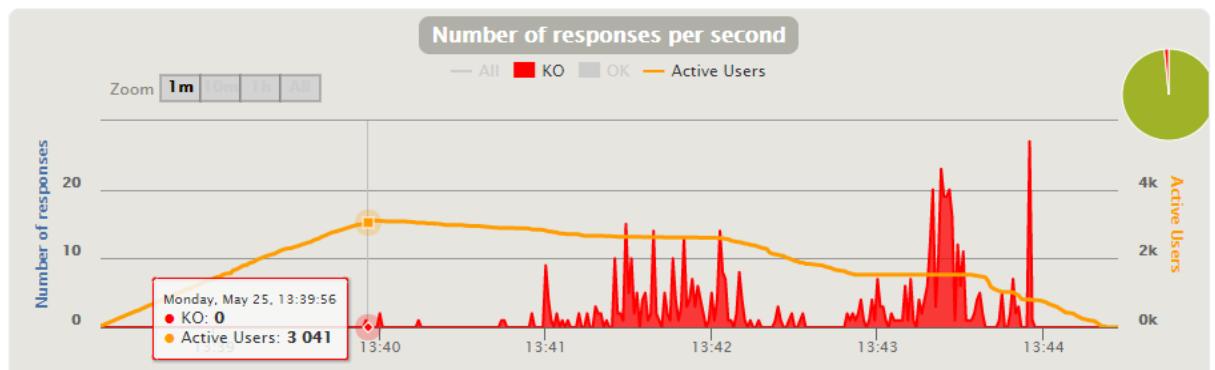
### Evidences:

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:



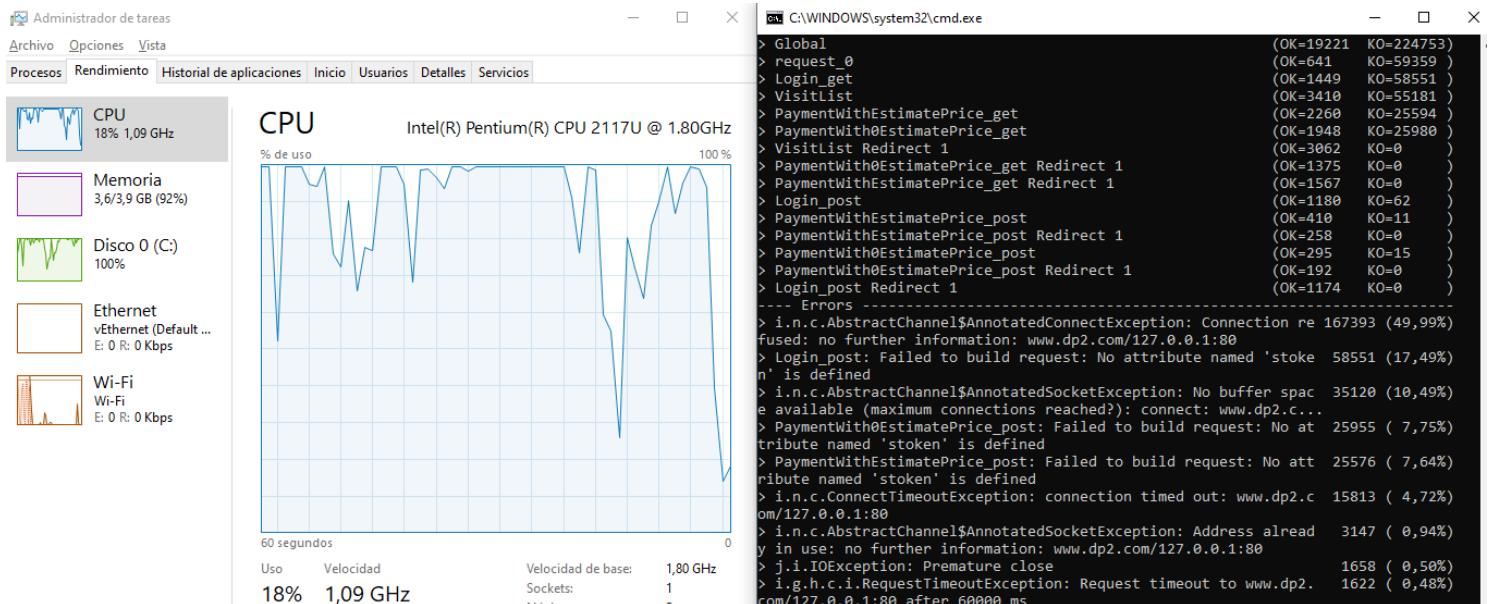
## [US10] Suggest price for a visit based on its type

A) Stress Test → Minimum number that is not supported by our system → **60.000 in 10**

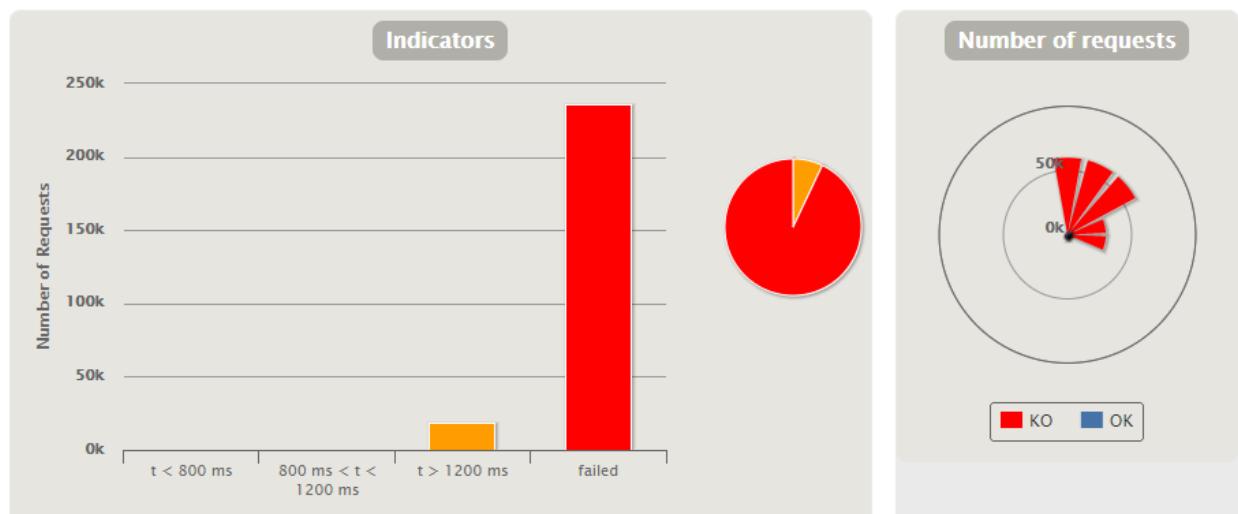
### Evidences:

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:

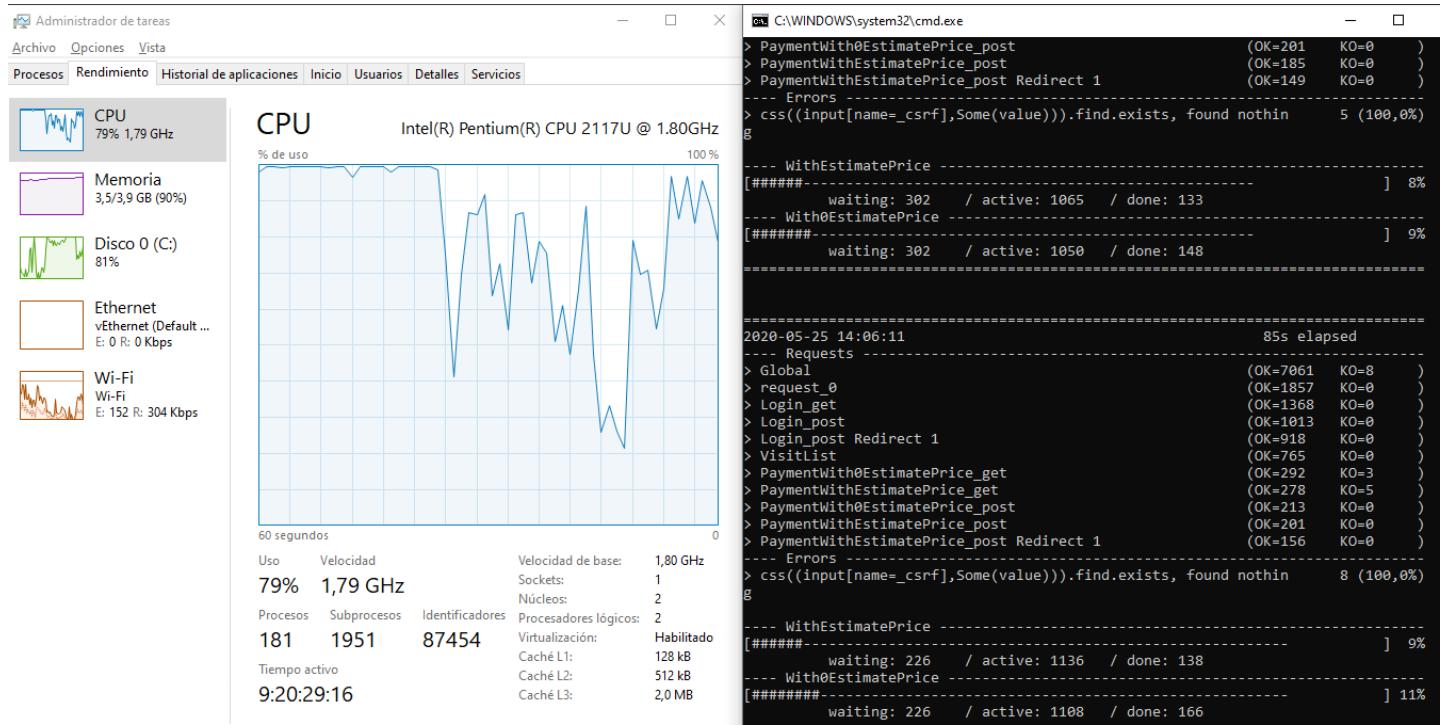


## B) Load Test → Maximum number that is supported with good performance → 2.200 in 100

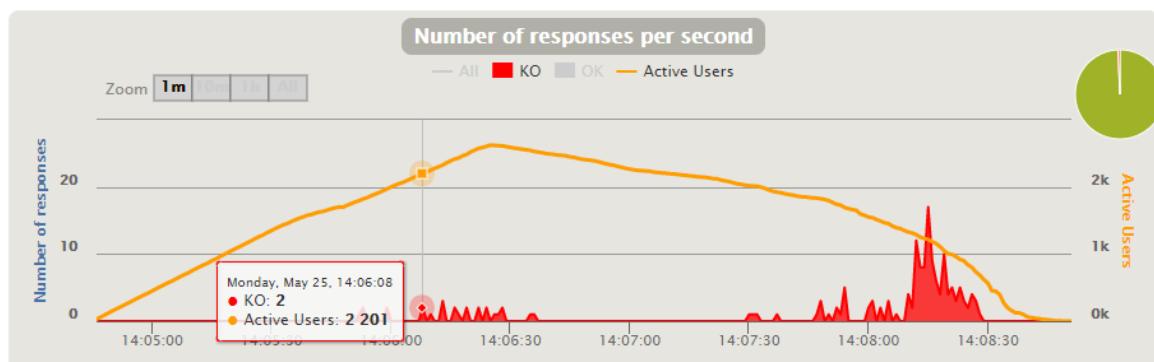
### Evidences:

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:



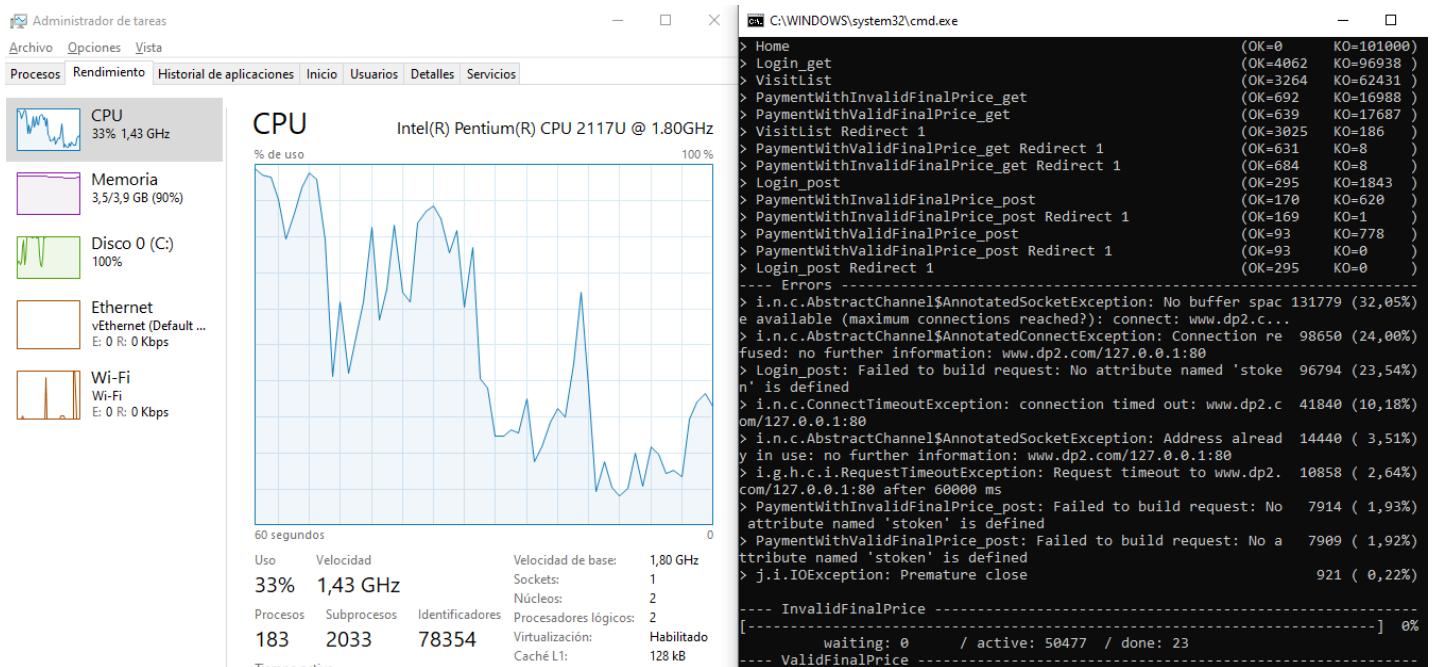
## [US11] Freely assign price to a visit

A) Stress Test → Minimum number that is not supported by our system → **101.000 in 10**

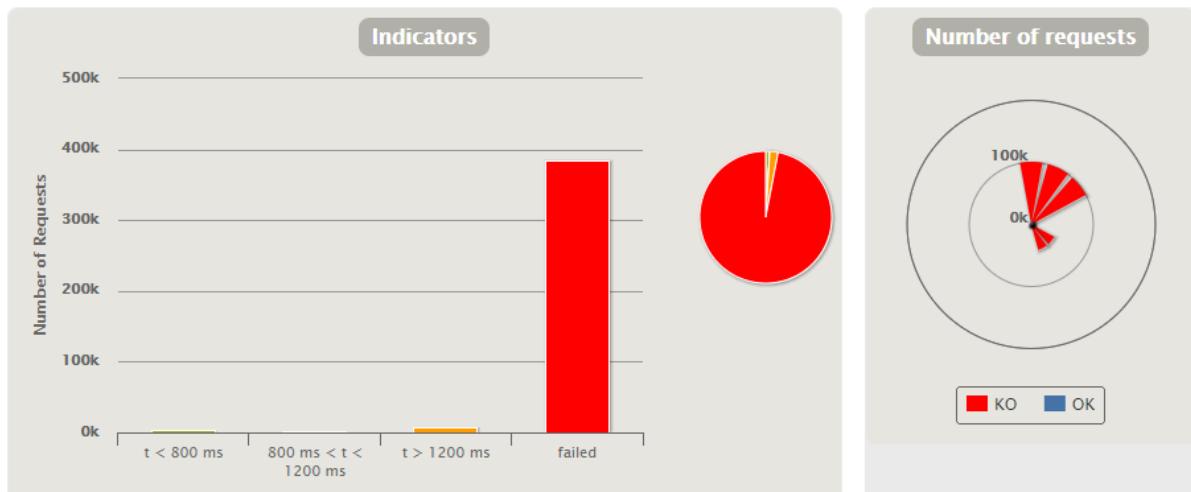
### Evidences:

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:

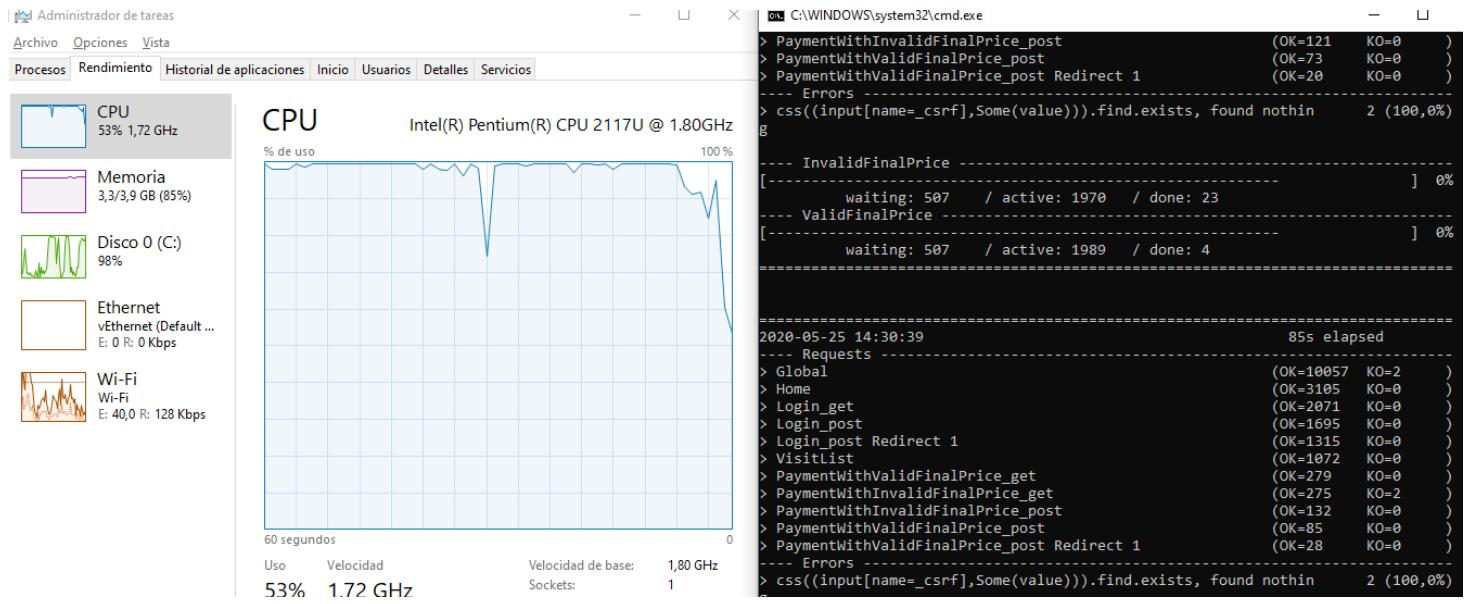


## B) Load Test → Maximum number that is supported with good performance → 4.000 in 100

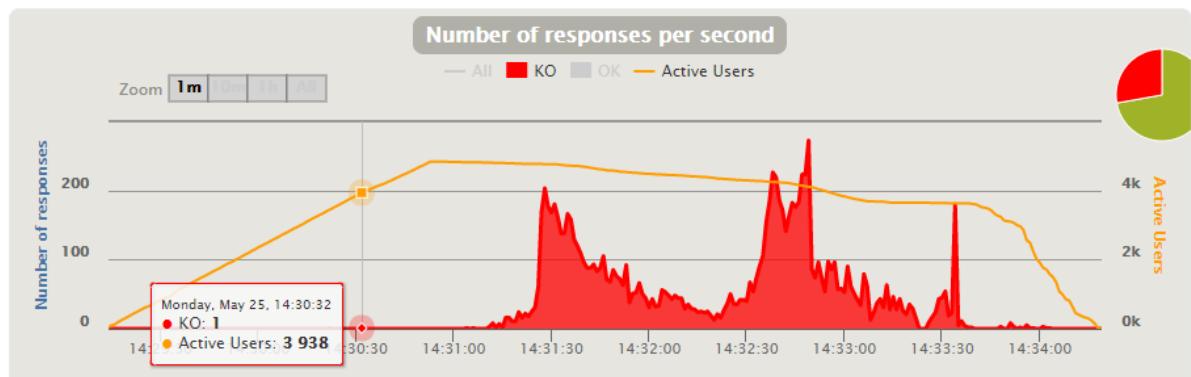
### Evidences:

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:



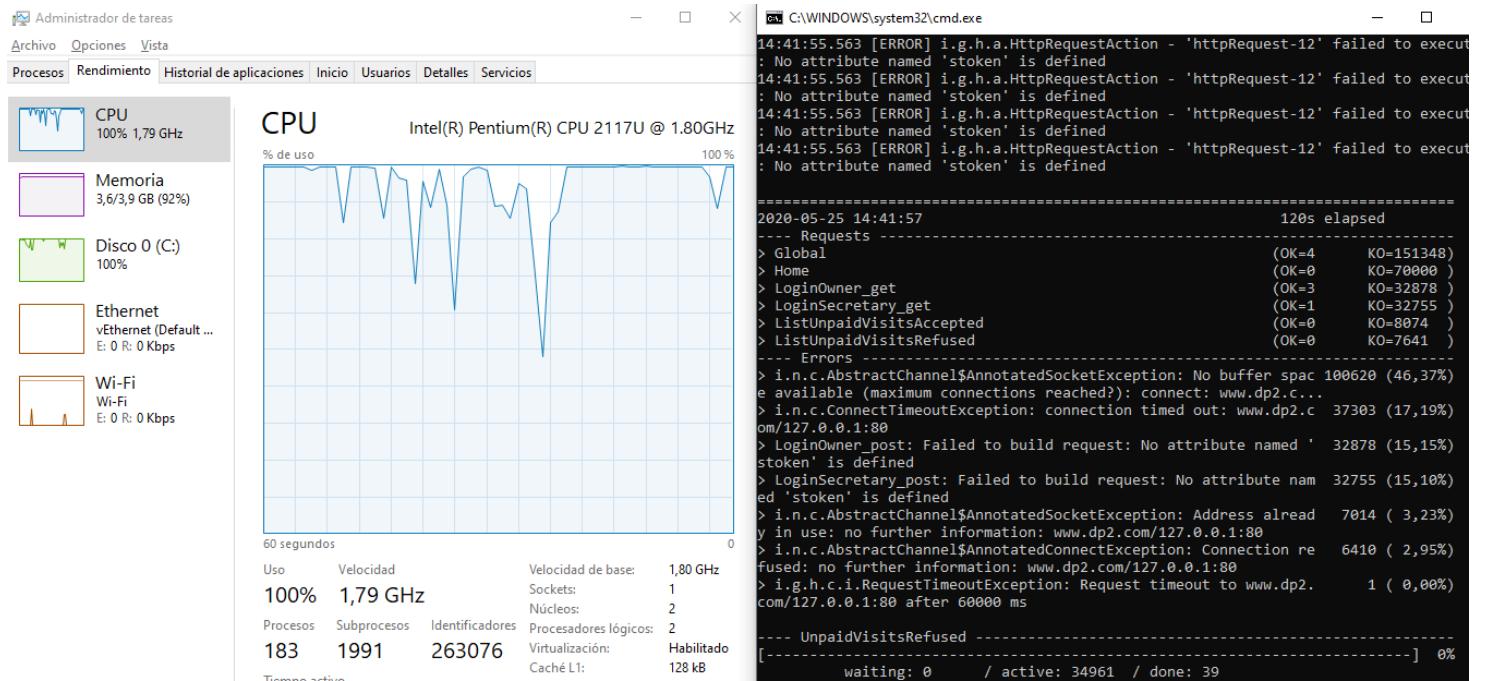
## [US12] View all unpaid visits

A) Stress Test → Minimum number that is not supported by our system → **70.000** in 10

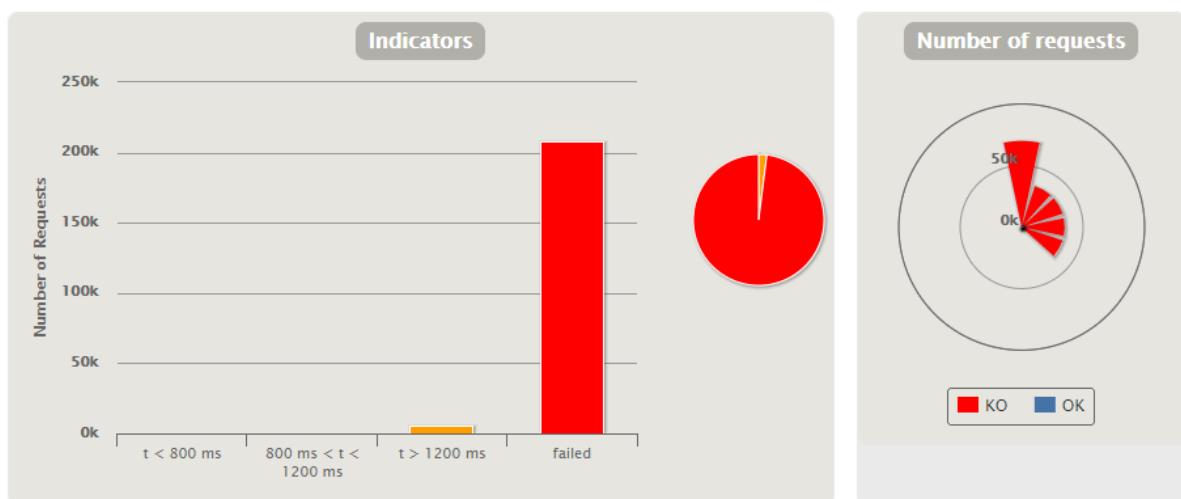
### Evidences:

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:

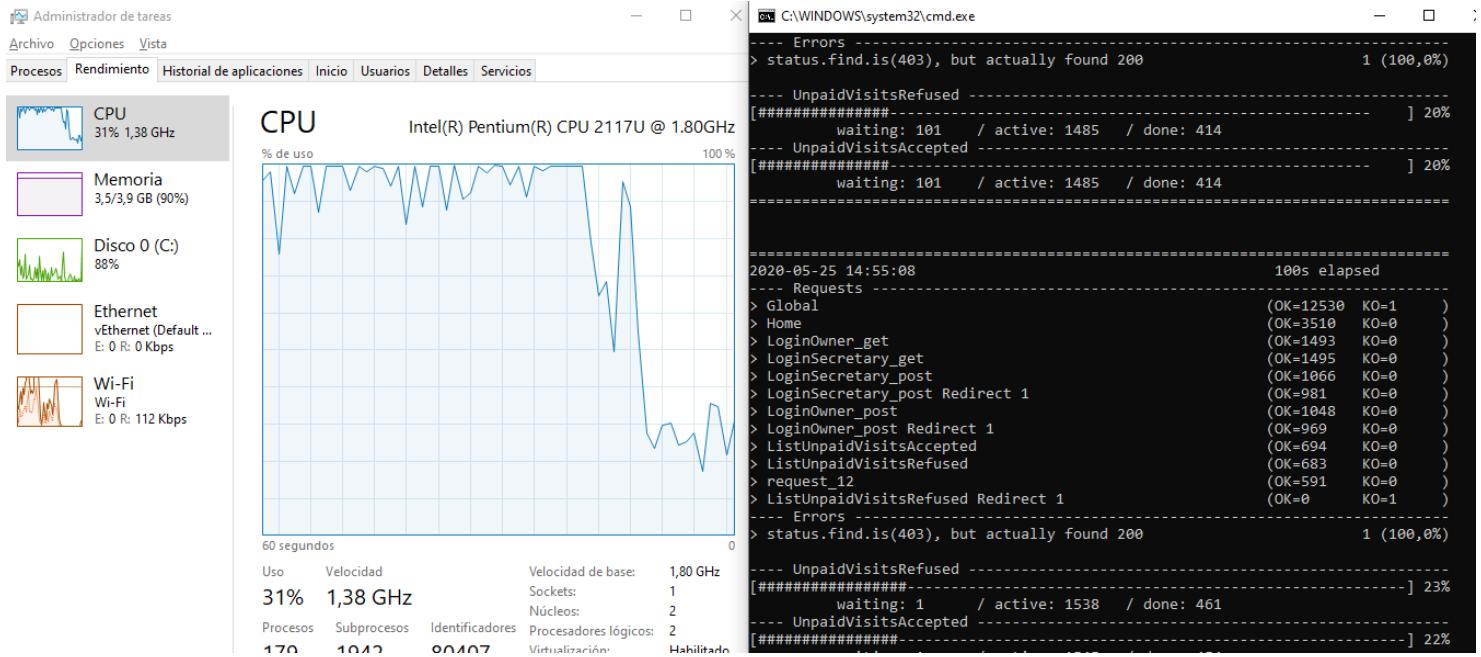


## B) Load Test → Maximum number that is supported with good performance → 3.000 in 100

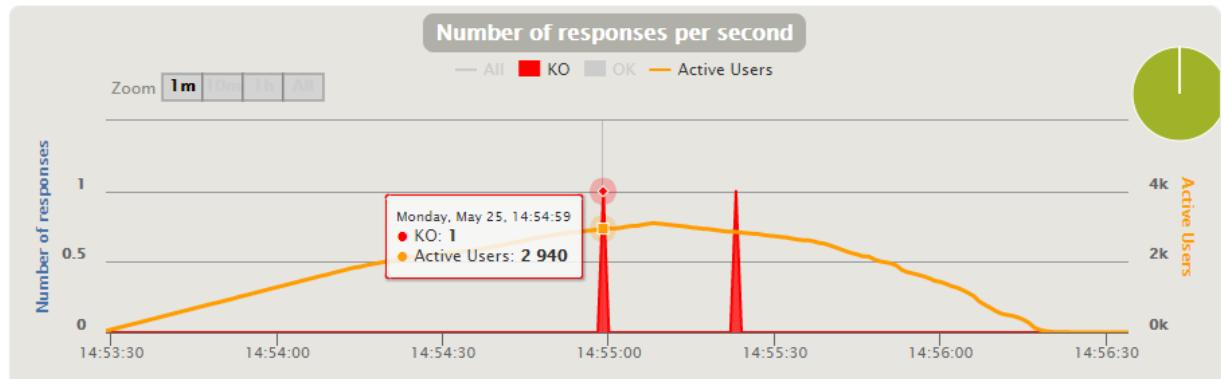
### Evidences:

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:



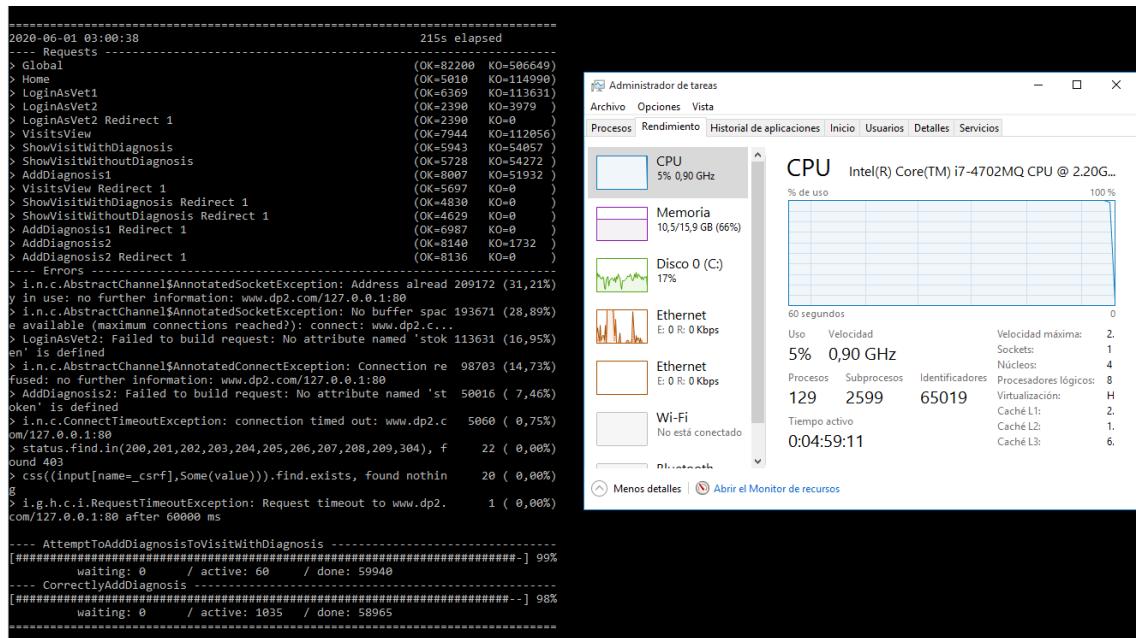
## [US13] Add diagnosis to a visit

A) Stress Test → Minimum number that is not supported by our system → 120k

### Evidences:

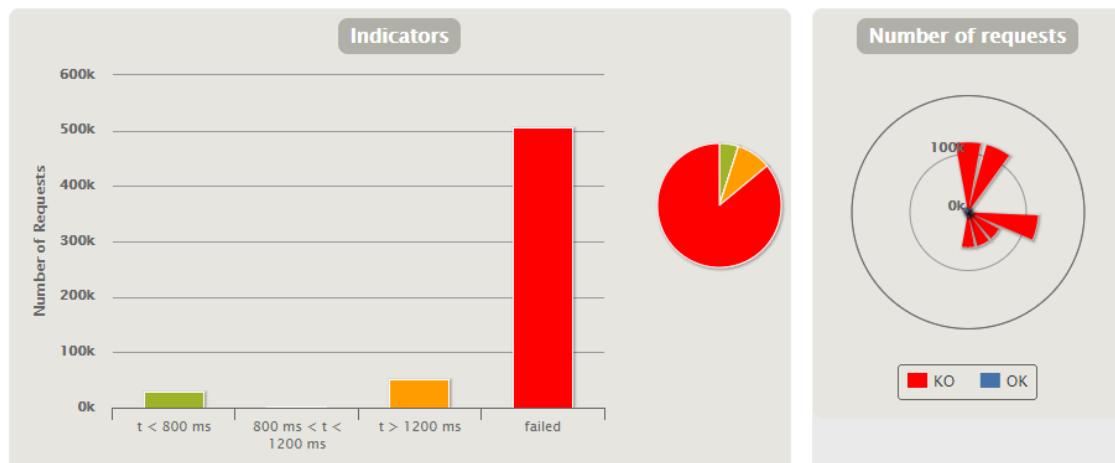
When performing a stress test with 120.000 (60.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

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



Capture of Gatling report:

### > Global Information

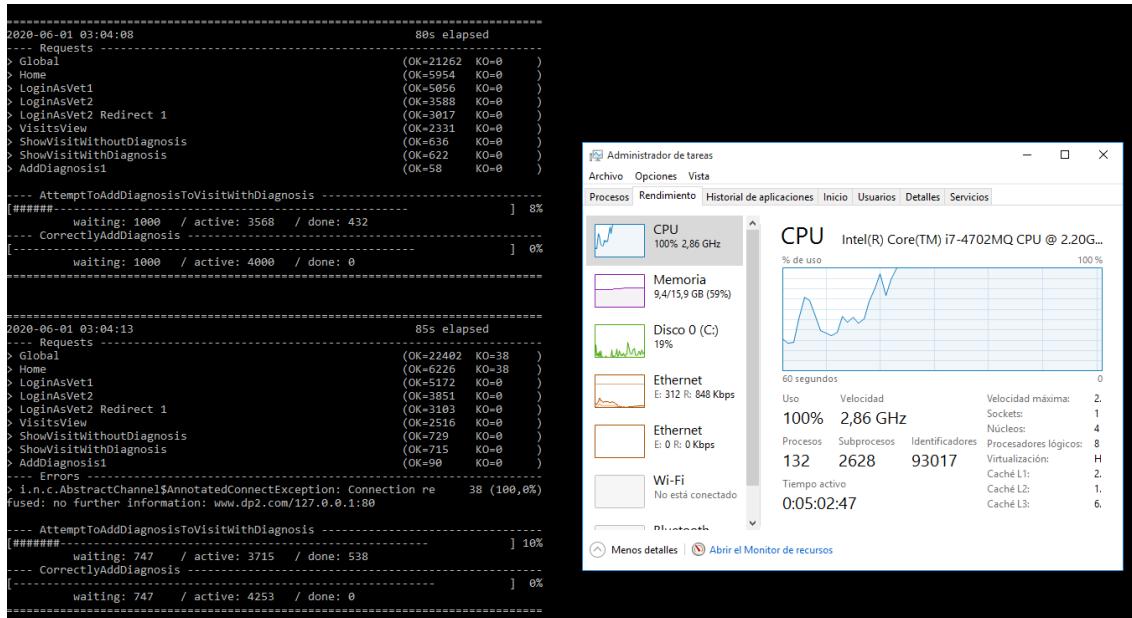


B) Load Test → Maximum number that is supported with good performance → 8.100

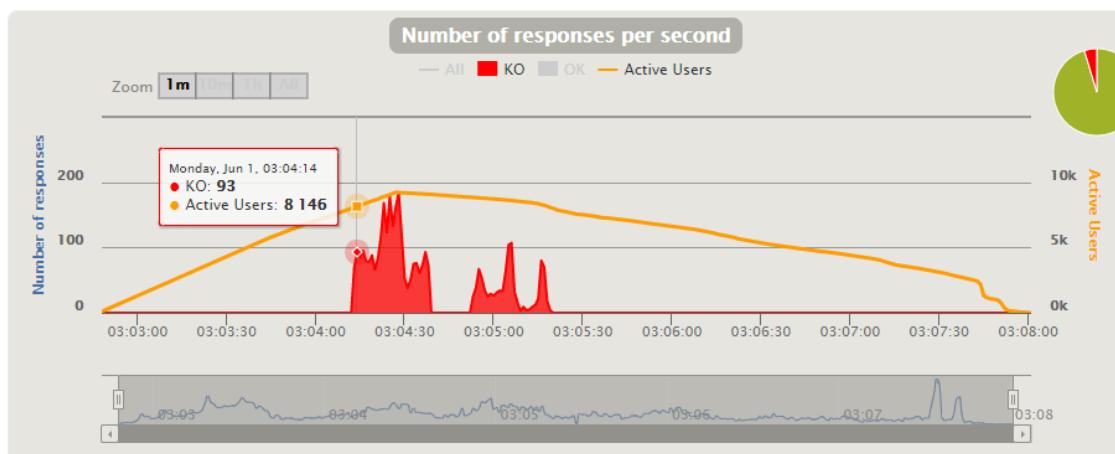
### Evidences:

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

A) 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.

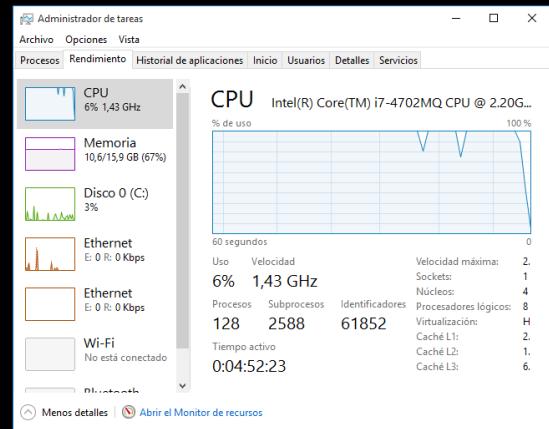
Capture of the performance monitor of our computer and the test execution console:

```

----- Requests -----
2020-06-01 02:53:49                                     280s elapsed
> Global                                         (OK=98662 KO=510837)
> Home                                          (OK=4759 KO=115241)
> LoginAsVet1                                    (OK=7289 KO=112711)
> LoginAsVet2                                    (OK=2149 KO=5140 )
> LoginAsVet2 Redirect 1                         (OK=2149 KO=0 )
> VisitView                                       (OK=621 KO=113679)
> ShowVisitWithDiagnosis                        (OK=623 KO=53763 )
> AttemptToAddEmptyPrescription1                (OK=925 KO=14095 )
> AddPrescription1                             (OK=9887 KO=50013 )
> AttemptToAddEmptyPrescription2                (OK=5464 KO=3139 )
> VisitView Redirect 1                          (OK=4418 KO=0 )
> AttemptToAddEmptyPrescription1 Redirect 1    (OK=5234 KO=0 )
> AddPrescription1 Redirect 1                  (OK=9213 KO=0 )
> ShowVisitWithDiagnosis Redirect 1            (OK=5383 KO=0 )
> AttemptToAddEmptyPrescription2 Redirect 1    (OK=4876 KO=0 )
> AddPrescription2                            (OK=9629 KO=3076 )
> AddPrescription2 Redirect 1                  (OK=9629 KO=0 )
----- Errors -----
> i.n.c.AbstractChannel$AnnotatedSocketException: No buffer spac 193873 (26,85%)
  e available (maximum connections reached): connect: www.dp2...
> i.n.c.AbstractChannel$AnnotatedSocketException: Address already 164490 (22,78%)
  y in use: no further information: www.dp2.com@127.0.0.1:80
> i.n.c.AbstractChannel$AnnotatedConnectException: Connection re 137749 (19,07%)
  Fused: further information: www.dp2.com@127.0.0.1:80
> LoginAsVet2: Failed to build request: No attribute named 'stok 112711 (15,61%)'
en is defined
> AttemptToAddEmptyPrescription2: Failed to build request: No at 51397 ( 7,12%)
attribute named 'stoken' is defined
> AddPrescription2: Failed to build request: No attribute named 47212 ( 6,54%)
'stoken' is defined
> i.n.c.ConnectTimeoutException: connection timed out: www.dp2.c 14562 ( 2,02%)
on@127.0.0.1:80
> status.find.in(200,201,202,203,204,205,206,207,208,209,304), f 142 ( 0,02%)
ound 483
> j.i.IOException: Premature close               19 ( 0,00%)
> css((input[name=_csrf]),Some(value)).find.exists, found nothing 1 ( 0,00%)
g
> i.g.h.c.i.RequestTimeoutException: Request timeout to www.dp2. 1 ( 0,00%)
com@127.0.0.1:80 after 60000 ms

----- AttemptToAddIncorrectPrescription -----
[##### waiting: 0 / active: 43 / done: 5957 ] 99%
----- CorrectlyAddPrescription -----
[##### waiting: 0 / active: 517 / done: 59483 ] 99%

```



Capture of Gatling report:

#### > Global Information



B) 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:

```

=====
2020-05-31 21:05:19          105s elapsed
--- Requests -----
> Global                         (OK=3402  KO=0    )
> Home                           (OK=1282  KO=0    )
> LoginAsVet1                     (OK=882   KO=0    )
> LoginAsVet2                     (OK=593   KO=0    )
> LoginAsVet2 Redirect 1          (OK=366   KO=0    )
> VisitasView                    (OK=263   KO=0    )
> ShowVisitWithDiagnosis         (OK=14    KO=0    )
> AttemptToAddEmptyPrescription1 (OK=12    KO=0    )

--- AttemptToAddIncorrectPrescription -----
[----] 0% waiting: 0 / active: 1000 / done: 0
--- CorrectlyAddPrescription -----
[----] 0% waiting: 0 / active: 1000 / done: 0
=====

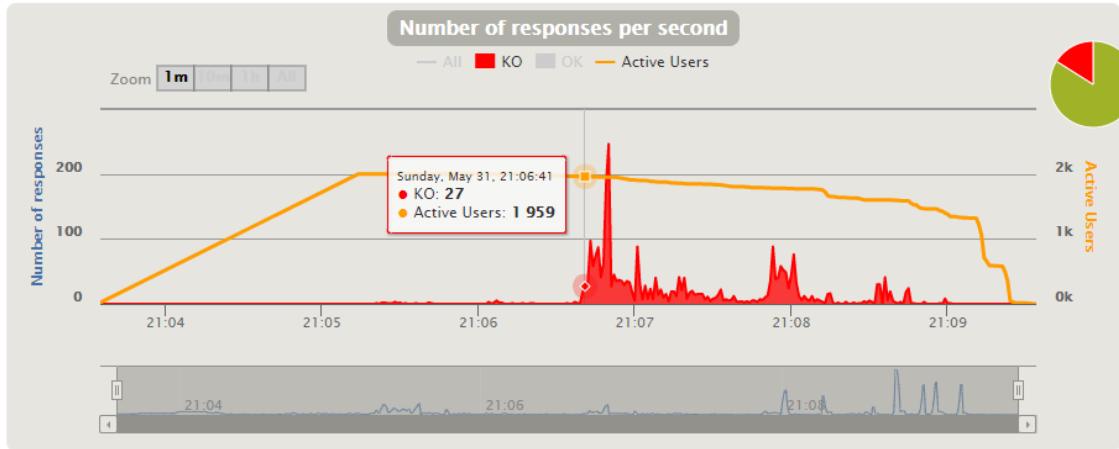
=====
2020-05-31 21:05:24          110s elapsed
--- Requests -----
> Global                         (OK=4210  KO=5    )
> Home                           (OK=1714  KO=0    )
> LoginAsVet1                     (OK=972   KO=0    )
> LoginAsVet2                     (OK=698   KO=3    )
> LoginAsVet2 Redirect 1          (OK=464   KO=0    )
> VisitasView                    (OK=279   KO=1    )
> ShowVisitWithDiagnosis         (OK=44    KO=0    )
> AttemptToAddEmptyPrescription1 (OK=39    KO=1    )

--- Errors -----
> i.g.h.c.i.RequestTimeoutException: Request timeout to www.dp2.      5 (100,0%)
com/127.0.0.1:80 after 60000 ms

--- AttemptToAddIncorrectPrescription -----
[----] 0% waiting: 0 / active: 1000 / done: 0
--- CorrectlyAddPrescription -----
[----] 0% waiting: 0 / active: 1000 / done: 0
=====


```

Capture of Gatling report:



### [US15] Select medicine from database

A) 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:

```

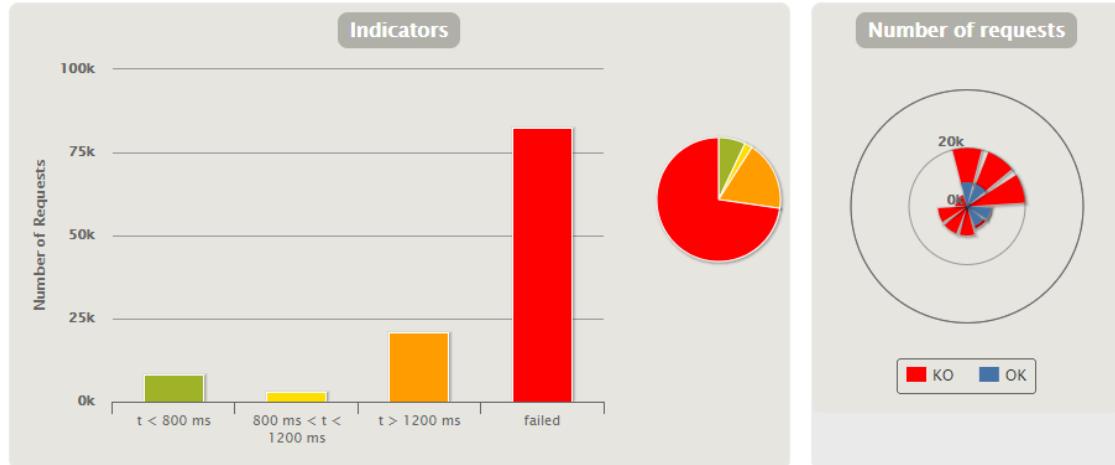
----- Requests -----
2020-06-01 02:45:57                                     280s elapsed
> Global                                         (OK=96796 KO=512208)
> Home                                          (OK=2590 KO=117410)
> LoginAsVet1                                    (OK=75 KO=112243)
> LoginAsVet2                                    (OK=268 KO=5229 )
> LoginAsVet2 Redirect 1                         (OK=2528 KO=5229 )
> VisitView                                      (OK=6909 KO=112091)
> ShowVisitWithDiagnosis                        (OK=6467 KO=53533 )
> AttemptToAddEmptyPrescription1                (OK=6514 KO=53486 )
> AddPrescription1                             (OK=8646 KO=51338 )
> AttemptToAddEmptyPrescription1 Redirect 1      (OK=5615 KO=0   )
> ShowVisitWithDiagnosis Redirect 1             (OK=5626 KO=0   )
> VisitView Redirect 1                          (OK=5895 KO=0   )
> AddPrescription1 Redirect 1                  (OK=7885 KO=0   )
> AttemptToAddEmptyPrescription2                (OK=5552 KO=3770 )
> AttemptToAddEmptyPrescription2 Redirect 1      (OK=4877 KO=0   )
> AddPrescription2                            (OK=8204 KO=3108 )
> AddPrescription2 Redirect 1                  (OK=8203 KO=0   )

----- Errors -----
> i.n.c.AbstractChannel$AnnotatedSocketException: No buffer spac 183938 (25,42%)
e available (maximum connections reached): connect: www.dp2.c...
> i.n.c.AbstractChannel$AnnotatedConnectException: Connection re 177218 (24,49%)
fused: no further information: www.dp2.com/127.0.0.1:80
> i.n.c.AbstractChannel$AnnotatedSocketException: Address already 138750 (19,18%)
y in use, no further information: www.dp2.com/127.0.0.1:80
> org.jboss.msc.Failed to build request: No attribute named 'stok' 112243 (15,51%)
en' is defined
> AttemptToAddEmptyPrescription2: Failed to build request: No a 50646 ( 7,00%)
tribute named 'stoken' is defined
> AddPrescription2: Failed to build request: No attribute named 48460 ( 6,70%)
'stoken' is defined
> i.n.c.ConnectTimeoutException: connection timed out: www.dp2.c 12137 ( 1,68%)
om/127.0.0.1:80
> status.find.in(200,201,202,203,204,205,206,207,208,209,304), f 78 ( 0,01%)
ound 403
> j.i.IOException: Premature close               41 ( 0,01%)
> i.g.h.c.i.RequestTimeoutException: Request timeout to www.dp2. 31 ( 0,00%)
com/127.0.0.1:80 after 60000 ms
> css((input[name=_csrf],Some(value))).find.exists, found nothin 15 ( 0,00%)
g

----- AttemptToAddIncorrectPrescription -----
[##### waiting: 0 / active: 102 / done: 59898 ] 99%
----- CorrectlyAddPrescription -----
[##### waiting: 0 / active: 718 / done: 59282 ] 98%
-----
```

Capture of Gatling report:

#### > Global Information

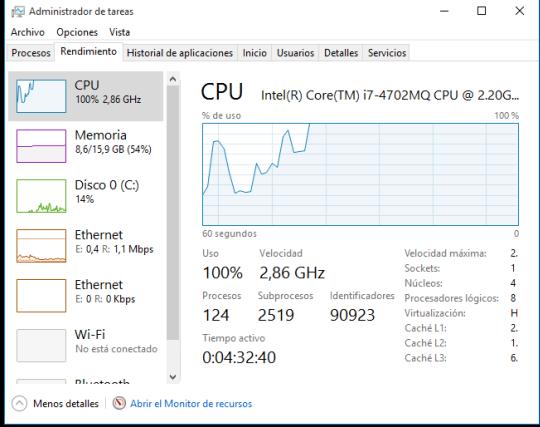


B) Load Test → Maximum number that is supported with good performance → **8.400**

#### 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:



```

2020-06-01 02:34:00
----- Requests -----
> Global (OK=25572 KO=0 )
> Home (OK=6986 KO=0 )
> LoginsVet1 (OK=5719 KO=0 )
> LoginsVet2 (OK=4407 KO=0 )
> LoginsVet2 Redirect 1 (OK=3919 KO=0 )
> VisitsView (OK=2869 KO=0 )
> AttemptToAddEmptyPrescription1 (OK=965 KO=0 )
> ShowVisitWithDiagnosis (OK=966 KO=0 )
> AddPrescription1 (OK=456 KO=0 )
> AttemptToAddEmptyPrescription2 (OK=285 KO=0 )

----- AttemptToAddIncorrectPrescription -----
[ waiting: 1002 / active: 3995 / done: 3 ] 0%
----- CorrectlyAddPrescription -----
[ waiting: 1001 / active: 3999 / done: 0 ] 0%

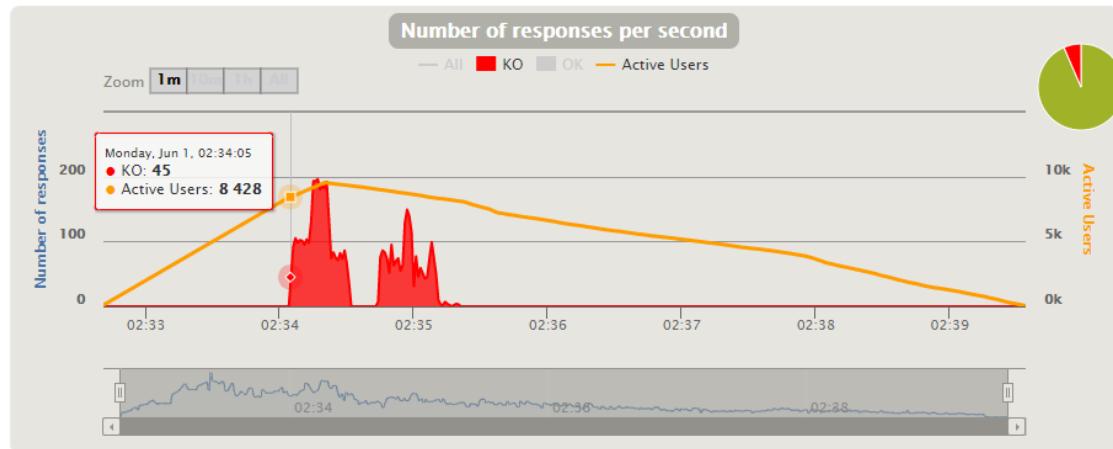
----- Requests -----
2020-06-01 02:34:05 85s elapsed
> Global (OK=28242 KO=99 )
> Home (OK=7362 KO=99 )
> LoginsVet1 (OK=5934 KO=0 )
> LoginsVet2 (OK=4642 KO=0 )
> LoginsVet2 Redirect 1 (OK=4143 KO=0 )
> VisitsView (OK=3019 KO=0 )
> AttemptToAddEmptyPrescription1 (OK=1078 KO=0 )
> ShowVisitWithDiagnosis (OK=1880 KO=0 )
> AddPrescription1 (OK=565 KO=0 )
> AttemptToAddEmptyPrescription2 (OK=418 KO=0 )

----- Errors -----
> i.n.c.AbstractChannel$AnnotatedConnectException: Connection re fused: no further information: www.dp2.com/127.0.0.1:80 99 (100,0%)

----- AttemptToAddIncorrectPrescription -----
[ waiting: 750 / active: 4163 / done: 87 ] 1%
----- CorrectlyAddPrescription -----
[ waiting: 750 / active: 4250 / done: 0 ] 0%

```

Capture of Gatling report:



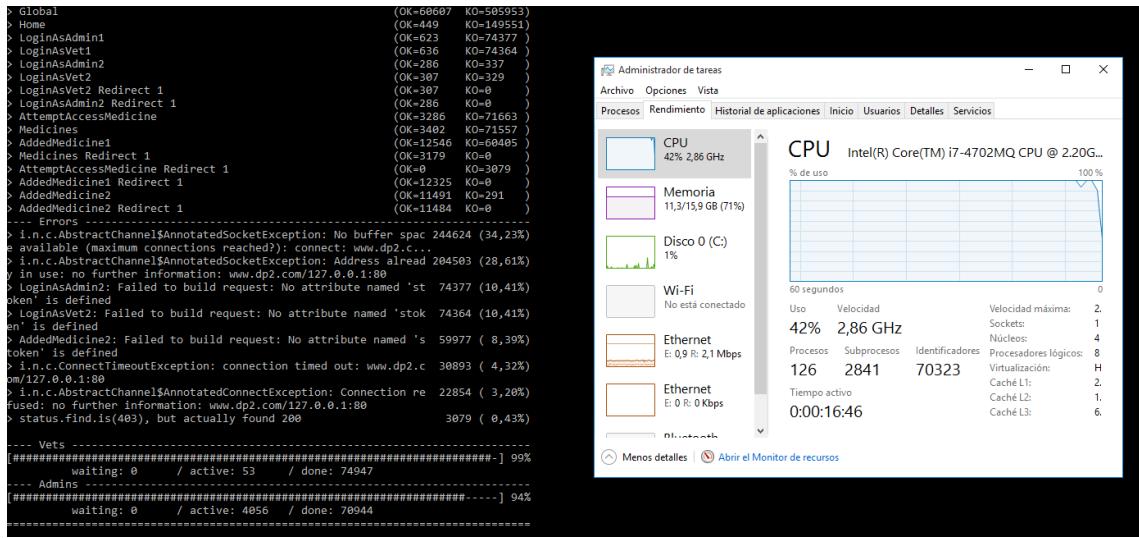
### [US16] Add new medicine the system

A) Stress Test → Minimum number that is not supported by our system → 150k

#### Evidences:

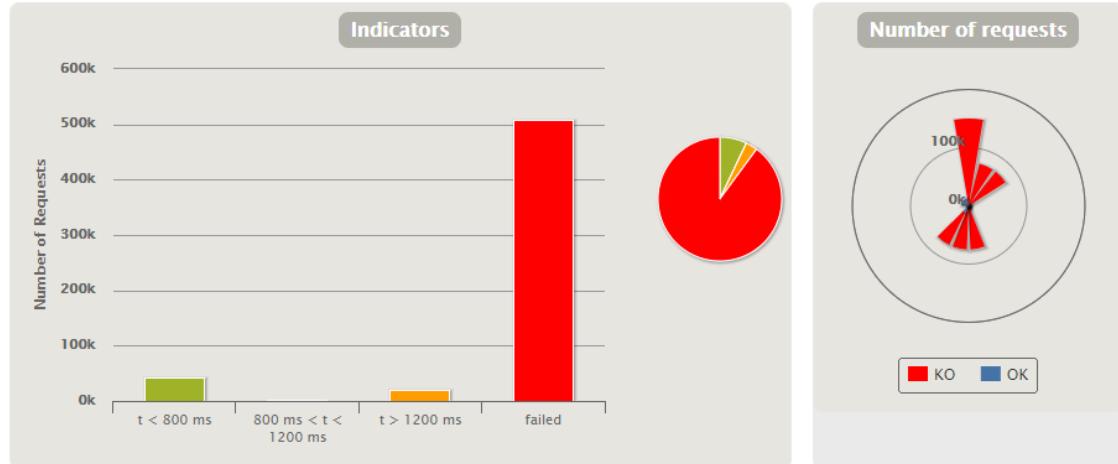
When performing a stress test with 150.000 (75.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

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



Capture of Gatling report:

#### > Global Information



**B) 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:

```

=====
2020-06-01 12:45:07
---- Requests -----
> Global (OK=39513 KO=0 )
> Home (OK=9192 KO=0 )
> LoginAsVet1 (OK=4187 KO=0 )
> LoginAsAdmin1 (OK=4189 KO=0 )
> LoginAsAdmin2 (OK=3437 KO=0 )
> LoginAsVet2 (OK=3438 KO=0 )
> LoginAsAdmin2 Redirect 1 (OK=3337 KO=0 )
> LoginAsVet2 Redirect 1 (OK=3330 KO=0 )
> AttemptAccessMedicine (OK=2523 KO=0 )
> Medicines (OK=2679 KO=0 )
> AddedMedicine1 (OK=2133 KO=0 )
> AddedMedicine2 (OK=655 KO=0 )
> AddedMedicine2 Redirect 1 (OK=413 KO=0 )

---- Vets -----
[#####-----] 36% waiting: 255 / active: 3219 / done: 1526
---- Admins -----
[#####-----] 0% waiting: 255 / active: 4745 / done: 0
=====

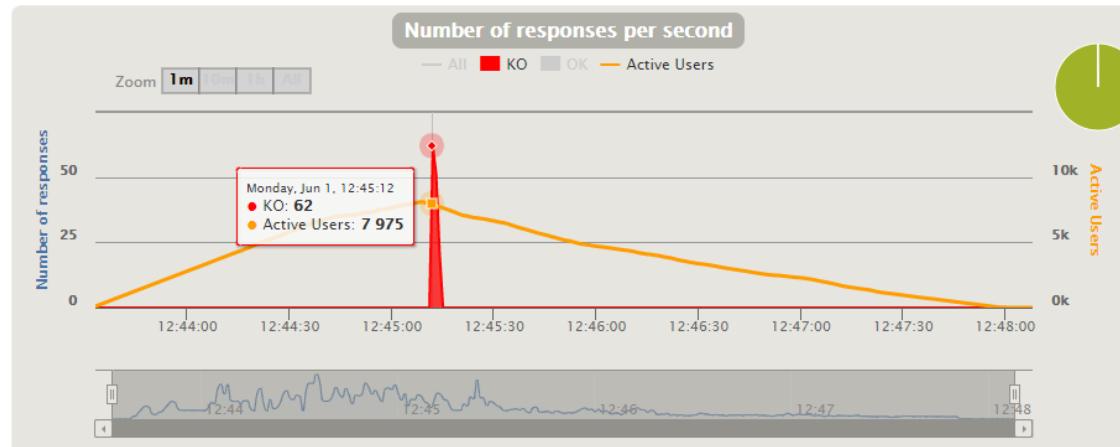
2020-06-01 12:45:12
100s elapsed
---- Requests -----
> Global (OK=4175 KO=97 )
> Home (OK=9491 KO=97 )
> LoginAsVet1 (OK=3446 KO=0 )
> LoginAsAdmin1 (OK=3437 KO=0 )
> LoginAsAdmin2 (OK=3595 KO=0 )
> LoginAsVet2 (OK=3591 KO=0 )
> LoginAsAdmin2 Redirect 1 (OK=3454 KO=0 )
> LoginAsVet2 Redirect 1 (OK=3451 KO=0 )
> AttemptAccessMedicine (OK=2718 KO=0 )
> Medicines (OK=2817 KO=0 )
> AddedMedicine1 (OK=2252 KO=0 )
> AddedMedicine2 (OK=843 KO=0 )
> AddedMedicine2 Redirect 1 (OK=678 KO=0 )

> i.n.c.AbstractChannels$AnnotatedConnectException: Connection re 97 (100,0%
fused: no further information: www.dp2.com/127.0.0.1:80

---- Vets -----
[#####-----] 37% waiting: 1 / active: 3102 / done: 1897
---- Admins -----
[#####-----] 5% waiting: 1 / active: 4720 / done: 279
=====


```

Capture of Gatling report:



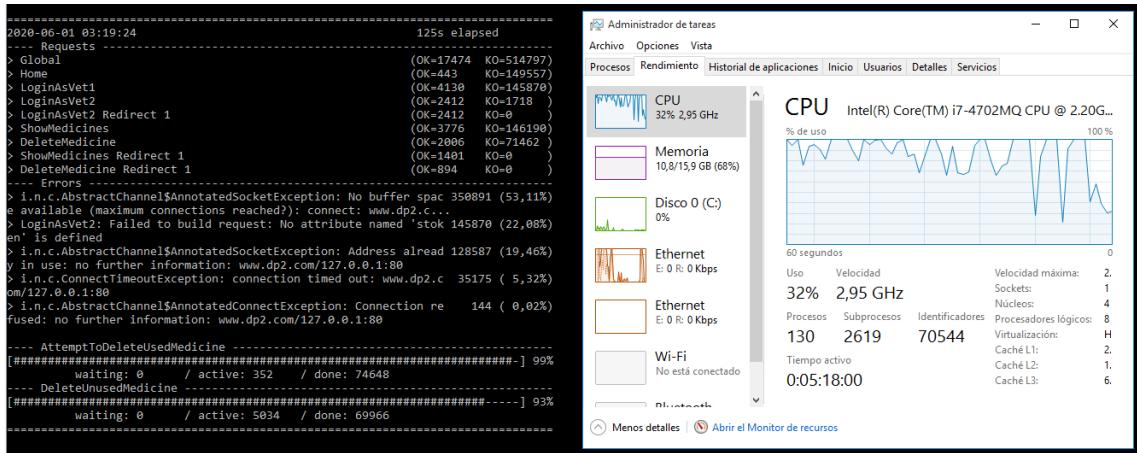
### [US17] Edit or delete the medicines in the system

A) Stress Test → Minimum number that is not supported by our system → 150k

Evidences:

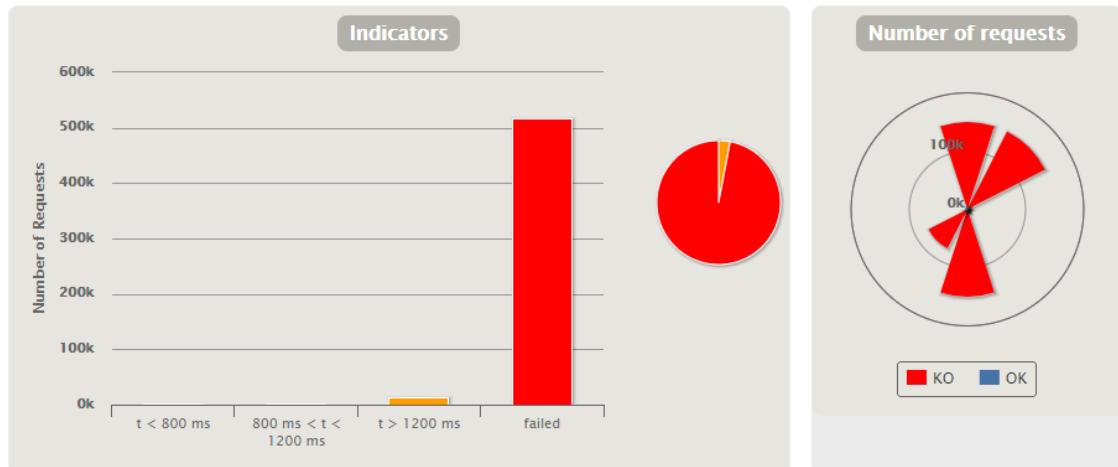
When performing a stress test with 150.000 (75.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

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



Capture of Gatling report:

#### > Global Information

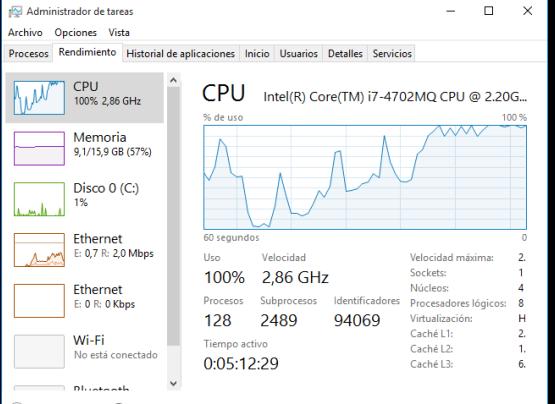


B) Load Test → Maximum number that is supported with good performance → **8.600**

#### 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:



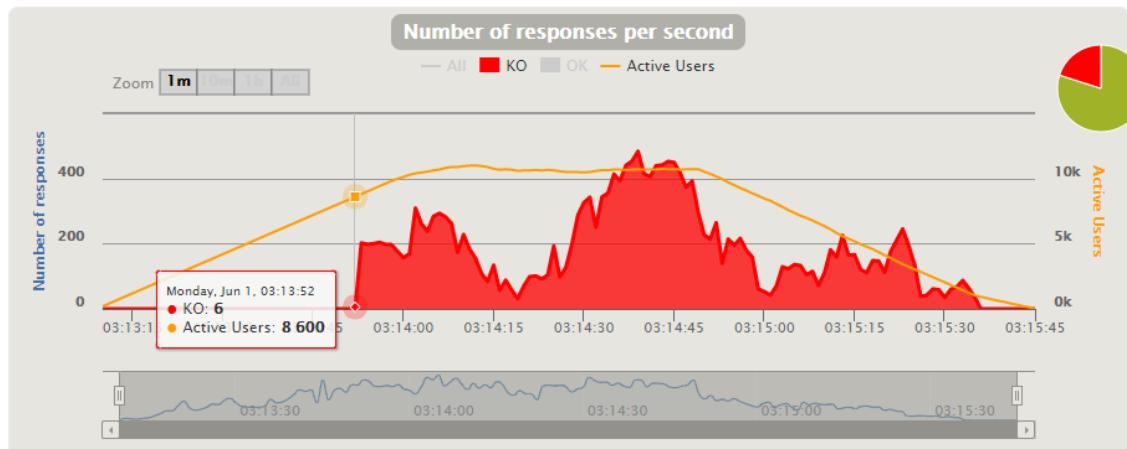
```

2020-06-01 03:13:49                                         40s elapsed
---- Requests ----
> Global                                         (OK=20066 KO=0    )
> Home                                           (OK=7736   KO=0    )
> LoginAsVet1                                     (OK=6159   KO=0    )
> LoginAsVet2                                     (OK=3162   KO=0    )
> LoginAsVet2 Redirect 1                         (OK=2847   KO=0    )
> ShowMedicines                                  (OK=168    KO=0    )

---- AttemptToDeleteUsedMedicine
[----- waiting: 6002 / active: 3998 / done: 0 ] 0%
[----- DeleteUnusedMedicine                      ] 0%
[----- waiting: 6002 / active: 3998 / done: 0 ] 0%
-----                                         45s elapsed
2020-06-01 03:13:54                                         45s elapsed
---- Requests ----
> Global                                         (OK=24959 KO=594   )
> Home                                           (OK=8192   KO=594   )
> LoginAsVet1                                     (OK=7919   KO=0    )
> LoginAsVet2                                     (OK=4328   KO=0    )
> LoginAsVet2 Redirect 1                         (OK=4268   KO=0    )
> ShowMedicines                                  (OK=1152   KO=0    )
---- Errors -----
> i.n.c.AbstractChannel$AnnotatedConnectException: Connection re     504 (100,0%)
Fused: no further information: www.dp2.com/127.0.0.1:80
---- AttemptToDeleteUsedMedicine
[----- waiting: 5581 / active: 4499 / done: 0 ] 0%
---- DeleteUnusedMedicine                      ] 0%
[----- waiting: 5582 / active: 4498 / done: 0 ] 0%
-----                                         45s elapsed
2020-06-01 03:13:54                                         45s elapsed
---- Requests ----
> Global                                         (OK=24959 KO=594   )
> Home                                           (OK=8192   KO=594   )
> LoginAsVet1                                     (OK=7919   KO=0    )
> LoginAsVet2                                     (OK=4328   KO=0    )
> LoginAsVet2 Redirect 1                         (OK=4268   KO=0    )
> ShowMedicines                                  (OK=1152   KO=0    )
---- Errors -----
> i.n.c.AbstractChannel$AnnotatedConnectException: Connection re     504 (100,0%)
Fused: no further information: www.dp2.com/127.0.0.1:80
---- AttemptToDeleteUsedMedicine
[----- waiting: 5581 / active: 4499 / done: 0 ] 0%
---- DeleteUnusedMedicine                      ] 0%
[----- waiting: 5582 / active: 4498 / done: 0 ] 0%
-----                                         45s elapsed

```

Capture of Gatling report:



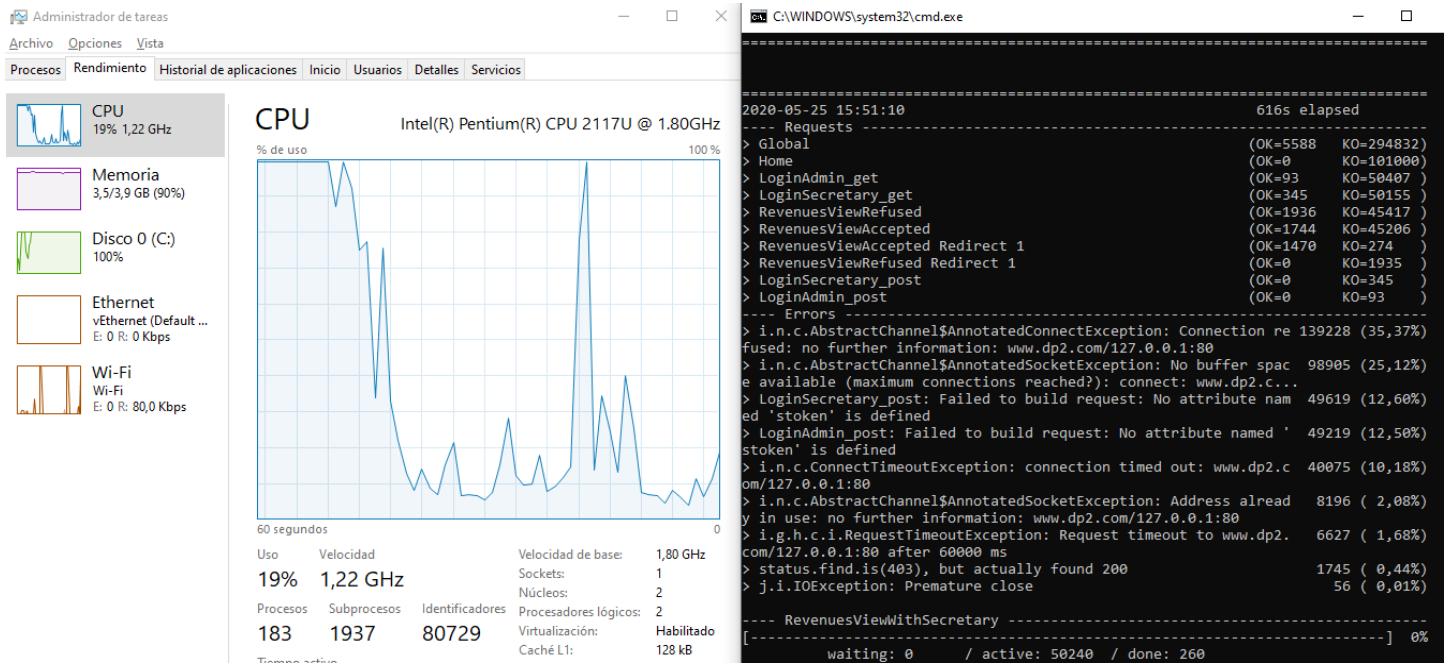
## [US18] View revenue by month

A) Stress Test → Minimum number that is not supported by our system → **101.000 in 10**

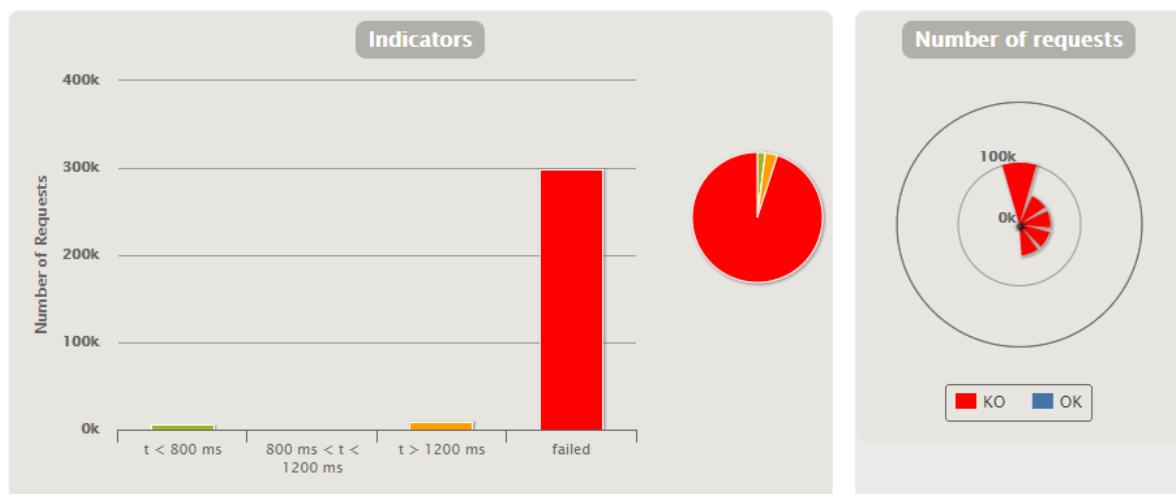
### Evidences:

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:

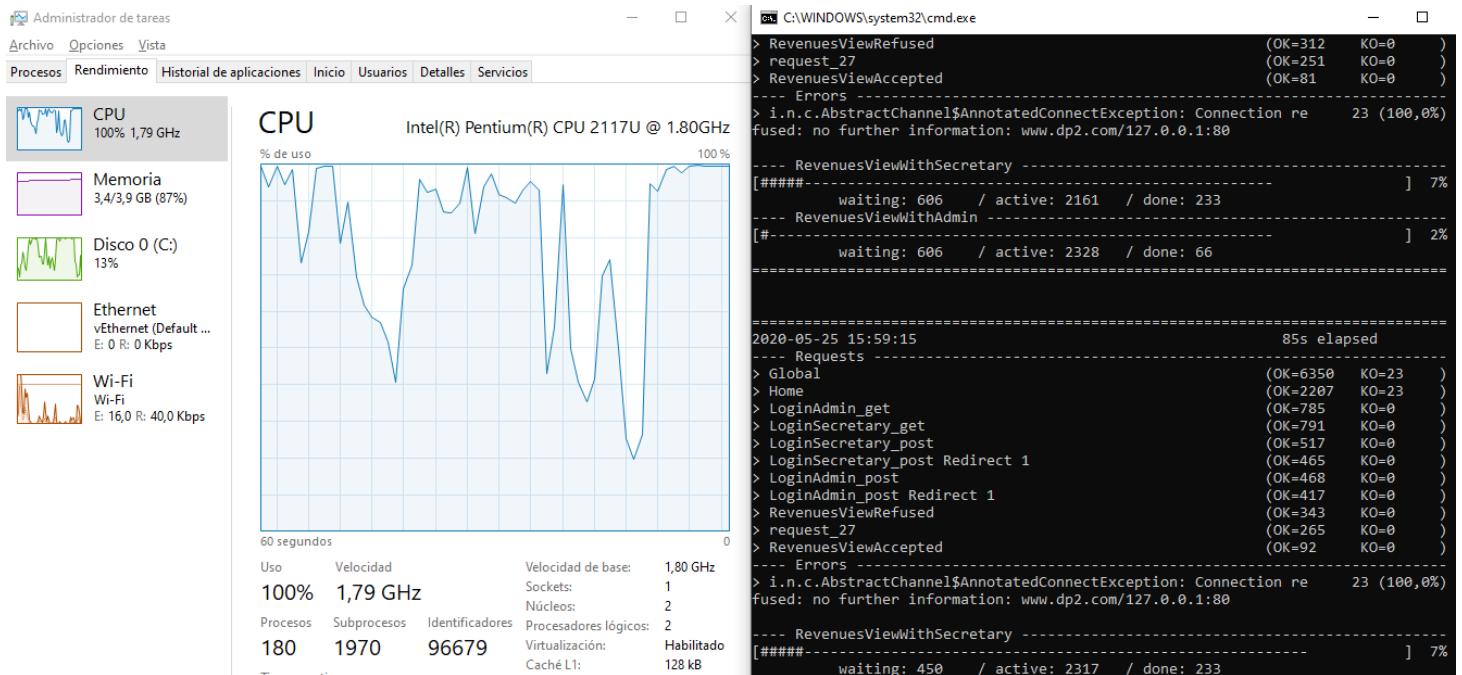


## B) Load Test → Maximum number that is supported with good performance → 4.800 in 100

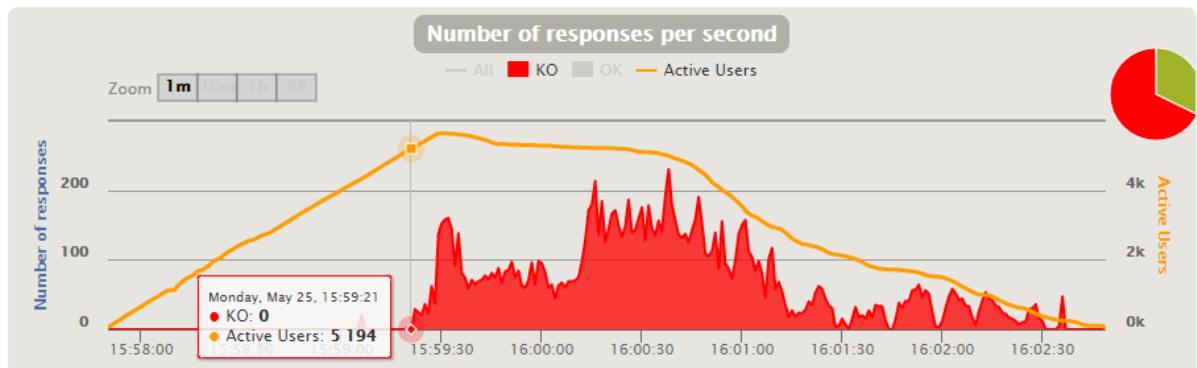
### Evidences:

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:



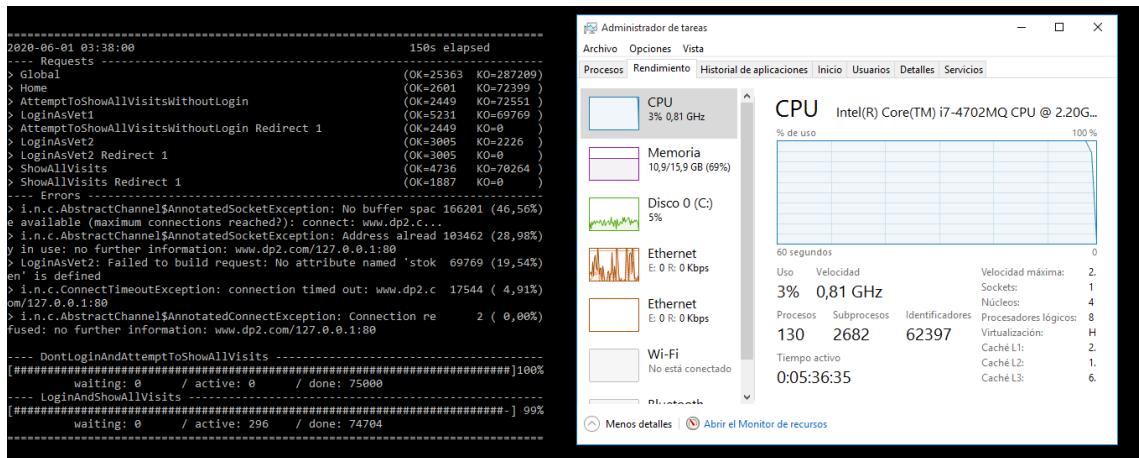
## [US19] See all the characteristics of visits already made

A) 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:

### > Global Information



B) Load Test → Maximum number that is supported with good performance → 7.600

### Evidences:

Performing a load test with 20.000 (10.000 users for each scenario) concurrent users, when the number of active users is 7.600 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:

```

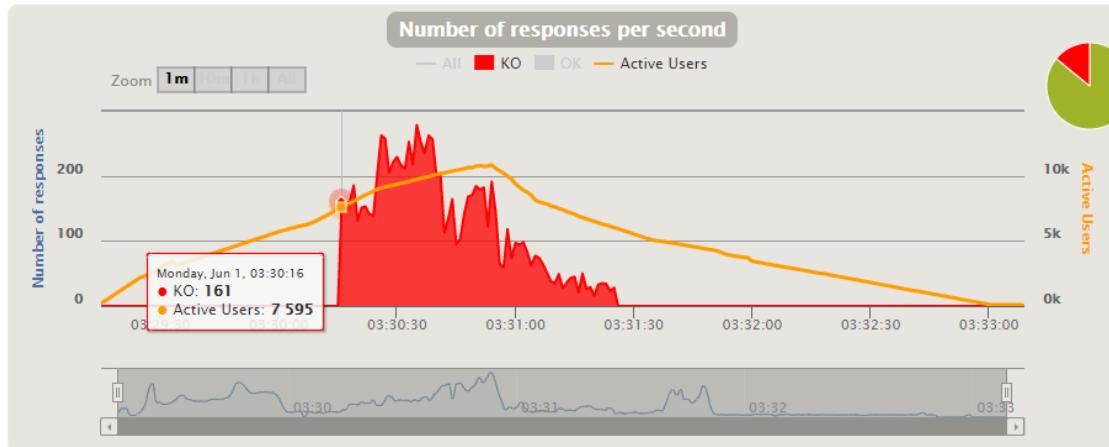
----- Requests ----- 65s elapsed
> Global (OK=22414 KO=0 )
> AttemptToShowAllVisitsWithoutLogin (OK=4696 KO=0 )
> AttemptToShowAllVisitsWithoutLogin Redirect 1 (OK=4357 KO=0 )
> Home (OK=4593 KO=0 )
> LoginAsVet1 (OK=3774 KO=0 )
> LoginAsVet2 (OK=2222 KO=0 )
> LoginAsVet2 Redirect 1 (OK=1986 KO=0 )
> ShowAllVisits (OK=876 KO=0 )

---- DontLoginAndAttemptToShowAllVisits -----
[##### waiting: 4001 / active: 1685 / done: 4314 ] 43%
---- LoginAndShowAllVisits -----
[### waiting: 4001 / active: 5555 / done: 444 ] 4%
----- Requests ----- 65s elapsed
> Global (OK=22923 KO=751 )
> AttemptToShowAllVisitsWithoutLogin (OK=4758 KO=382 )
> AttemptToShowAllVisitsWithoutLogin Redirect 1 (OK=4308 KO=0 )
> Home (OK=4745 KO=369 )
> LoginAsVet1 (OK=3927 KO=0 )
> LoginAsVet2 (OK=2341 KO=0 )
> LoginAsVet2 Redirect 1 (OK=2027 KO=0 )
> ShowAllVisits (OK=1997 KO=0 )

---- Errors -----
> i.n.c.AbstractChannels$AnnotatedConnectException: Connection re fused: no further information: www.dp2.com/127.0.0.1:80
---- DontLoginAndAttemptToShowAllVisits -----
[##### waiting: 3501 / active: 2163 / done: 4336 ] 43%
---- LoginAndShowAllVisits -----
[### waiting: 3501 / active: 5848 / done: 651 ] 6%

```

Capture of Gatling report:



## [US20] Add new types of visit

A) Stress Test → Minimum number that is not supported by our system → 150k

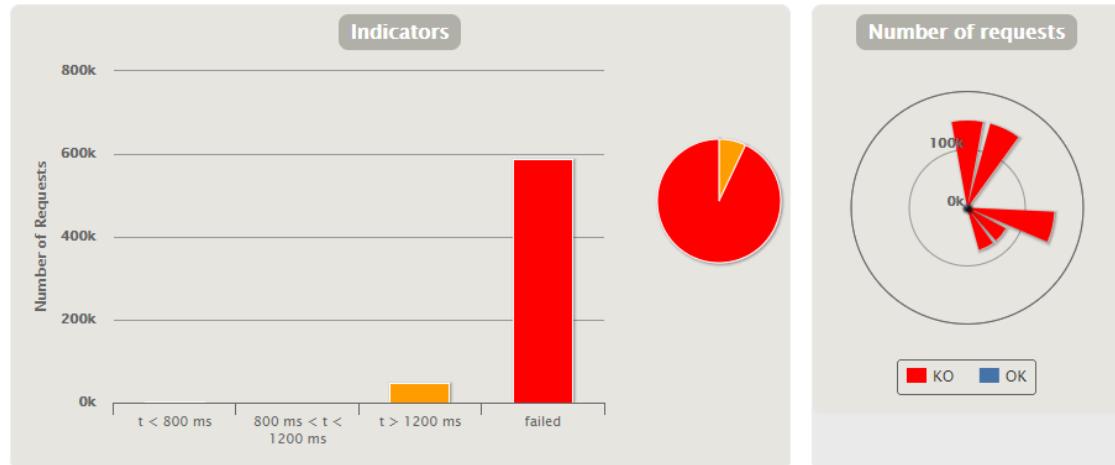
Evidences:

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:

## > Global Information



**B) Load Test** → Maximum number that is supported with good performance → 8.500

## Evidences:

Performing a load test with 10.000 (5.000 users for each scenario) concurrent users, when the number of active users is 8.500 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:

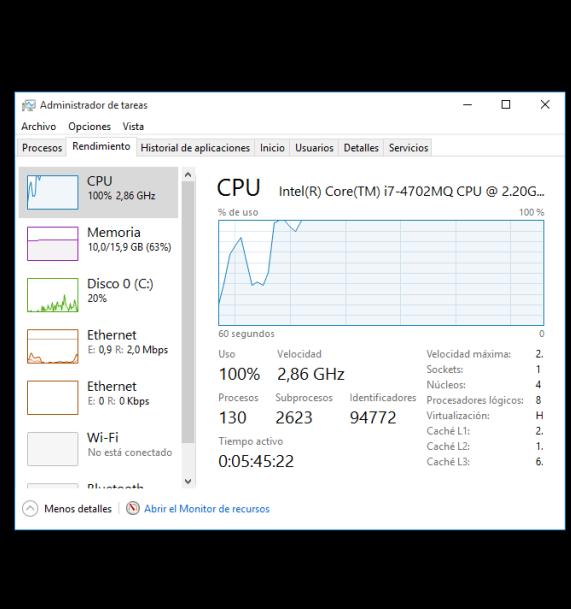
```
=====
2020-06-01 03:46:42                                         80s elapsed
---- Requests ---
> Global                                         (OK=16902 KO=0   )
> Home                                          (OK=5415 KO=0   )
> LoginAsVet1                                     (OK=4294 KO=0   )
> LoginAsVet2                                     (OK=3812 KO=0   )
> LoginAsVet2 Redirect 1                         (OK=2196 KO=0   )
> ShowVisitTypes                                  (OK=1744 KO=0   )
> AttemptToAddInvalidVisitType1                  (OK=123 KO=0    )
> AddVisitType1                                    (OK=118 KO=0    )

---- AtemptToAddIncorrectVisitType ---
[      waiting: 1000 / active: 4000 / done: 0 ] 0%
--> AddValidVisitType                            [      waiting: 1000 / active: 4000 / done: 0 ] 0%
[      waiting: 1000 / active: 4000 / done: 0 ] 0%

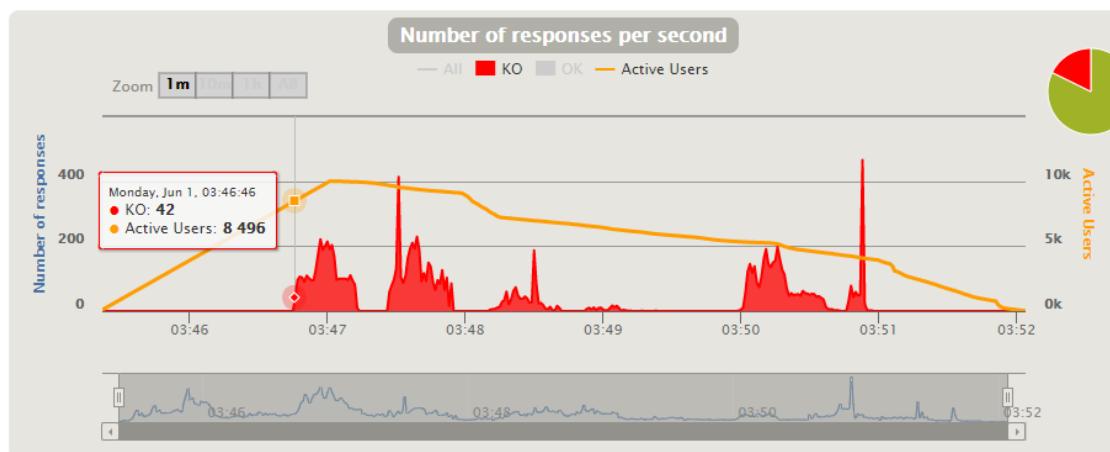

=====
2020-06-01 03:46:47                                         85s elapsed
---- Requests ---
> Global                                         (OK=18991 KO=88  )
> Home                                           (OK=6123 KO=88  )
> LoginAsVet1                                     (OK=4457 KO=0   )
> LoginAsVet2                                     (OK=3489 KO=0   )
> LoginAsVet2 Redirect 1                         (OK=2431 KO=0   )
> ShowVisitTypes                                  (OK=1849 KO=0   )
> AttemptToAddInvalidVisitType1                  (OK=327 KO=0    )
> AddVisitType1                                    (OK=315 KO=0    )

---- Errors ---
> i.n.c.AbstractChannel$AnnotatedConnectException: Connection re fused: no further information: www.dp2.com/127.0.0.1:80          88 (100,0%)

---- AtemptToAddIncorrectVisitType ---
[      waiting: 752 / active: 4248 / done: 0 ] 0%
--> AddValidVisitType                            [      waiting: 752 / active: 4248 / done: 0 ] 0%
```



## Capture of Gatling report:



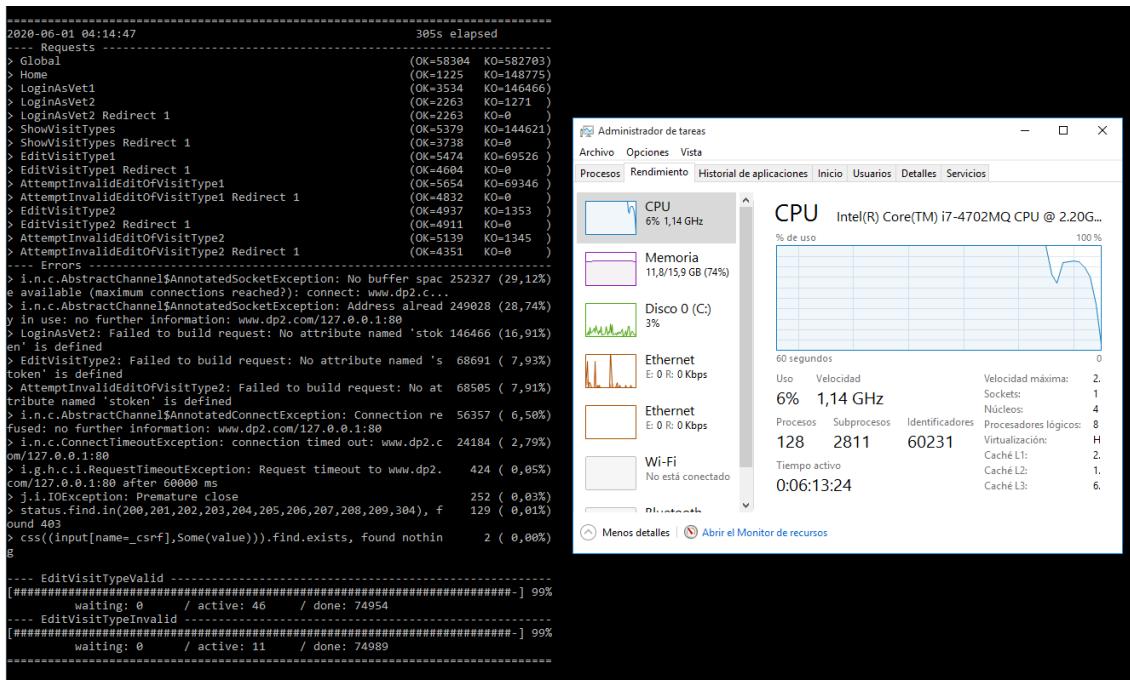
## [US21] Edit types of visit

A) Stress Test → Minimum number that is not supported by our system → 150k

## Evidences:

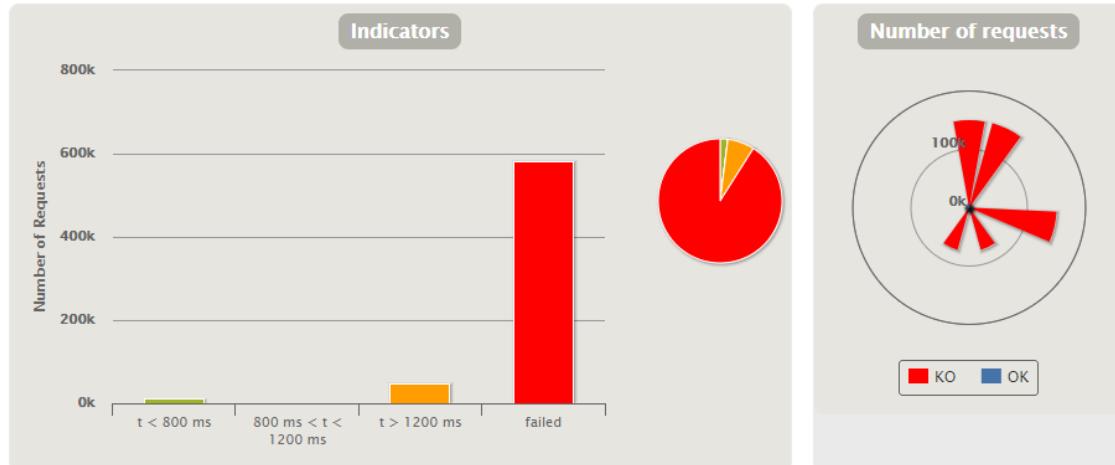
When performing a stress test with 150.000 (75.000 users for each scenario) concurrent users, we can see that most requests fail and that the CPU has a bottleneck. Therefore, this is the minimum number that our system does not support because with this many users we see bad performance.

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



Capture of Gatling report:

#### > Global Information



B) Load Test → Maximum number that is supported with good performance → 8.500

Evidences:

Performing a load test with 10.000 (5.000 users for each scenario) concurrent users, when the number of active users is 8.500 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:

```

=====
2020-06-01 04:02:57                               88s elapsed
---- Requests -----
> Global                                         (OK=13138 KO=0   )
> Home                                           (OK=4644  KO=0   )
> LoginAsVet1                                    (OK=3714  KO=0   )
> LoginAsVet2                                    (OK=2229  KO=0   )
> LoginAsVet2 Redirect 1                         (OK=1566  KO=0   )
> ShowVisitTypes                                 (OK=985   KO=0   )

---- EditVisitTypeValid -----
[ waiting: 1001 / active: 3999 / done: 0 ] 0%
---- EditVisitTypeInvalid -----
[ waiting: 1001 / active: 3999 / done: 0 ] 0%
=====

2020-06-01 04:03:02                               85s elapsed
---- Requests -----
> Global                                         (OK=13496 KO=103  )
> Home                                           (OK=4743  KO=103  )
> LoginAsVet1                                    (OK=3743  KO=0   )
> LoginAsVet2                                    (OK=2310  KO=0   )
> LoginAsVet2 Redirect 1                         (OK=1589  KO=0   )
> ShowVisitTypes                                (OK=1091  KO=0   )
> EditVisitType                                 (OK=4    KO=0   )
> AttemptInvalidEditOfVisitType1               Errors
> i.n.c.AbstractChannelsAnnotatedConnectException: Connection re 103 (100,0%)
Fused: further information: www.dp2.com/127.0.0.1:80

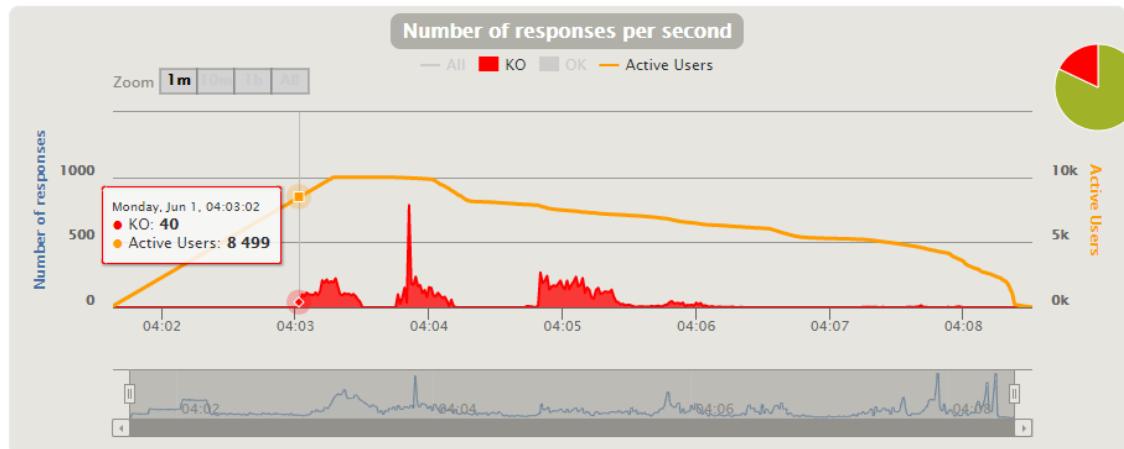
---- EditVisitTypeValid -----
[ waiting: 748 / active: 4252 / done: 0 ] 0%
---- EditVisitTypeInvalid -----
[ waiting: 749 / active: 4251 / done: 0 ] 0%
=====

  CPU Intel(R) Core(TM) i7-4702MQ CPU @ 2.20G... 100%
  % de uso
  60 segundos
  Uso   Velocidad   Velocidad máxima: 2.
  100%  2,86 GHz   Sockets: 1
  Procesos Subprocesos Identificadores Procesadores lógicos: 8
  129   2525      93721   Virtualización: H
  Tiempo activo
  0:06:01:38   Caché L1: 2.
  Caché L2: 1.
  Caché L3: 6.

  Administrador de tareas
  Archivo Opciones Vista
  Procesos Rendimiento Historial de aplicaciones Inicio Usuarios Detalles Servicios
  CPU Intel(R) Core(TM) i7-4702MQ CPU @ 2.20G...
  Memoria 10,7/15,9 GB (67%)
  Disco 0 (C:) 14%
  Ethernet E: 96,0 R: 224 Kbps
  Ethernet E: 0 R: 0 Kbps
  Wi-Fi No está conectado
  Bluetooth
  Menos detalles | Abrir el Monitor de recursos

```

Capture of Gatling report:



## MAXIMUM TOTAL SYSTEM PERFORMANCE:

The user story that supports the least number of concurrent users with good performance is US14, with 1900 users. Therefore, the maximum number of users that our application can support at the same time is at least 1900.