

**grado en ingeniería informática – ingeniería del software**

**Software para la Gestión de Centros de Salud**

Clinic Management Software

**josé gamaza díaz**

**alberto murillo jaenes**

**francisco javier rosa rodríguez**

**iván santos domínguez**

**curso 2019/2020**

Escuela Técnica Superior de Ingeniería Informática

Tabla de contenidos

[1. Datos del grupo de trabajo 3](#_Toc42164947)

[2. Descripción a grandes rasgos de la aplicación web 4](#_Toc42164948)

[3. Tabla de repartición de historias de usuario por prioridad y parejas 4](#_Toc42164949)

[4. Historias de usuario, escenarios e informes de las pruebas de rendimiento, profiling y refactoring 5](#_Toc42164950)

[5. Refactorizaciones con SonarCloud 43](#_Toc42164951)

[6. Justificación de cómo han ido consiguiendo cada uno de los requisitos del nivel señalado 44](#_Toc42164952)

[7. Análisis retrospectivo de los sprints 46](#_Toc42164953)

## 1. Datos del grupo de trabajo

|  |  |
| --- | --- |
|  |  |
| Gamaza Díaz, José | Murillo Jaenes, Alberto |
| josgamdia1@alum.us.es | [albmurjae@alum.us.es](mailto:albmurjae@alum.us.es) |

|  |  |
| --- | --- |
|  |  |
| Rosa Rodríguez, Francisco Javier | Santos Domínguez, Iván |
| frarosrod2@alum.us.es | ivasandom@alum.us.es |

**Número de grupo:** G2-14.

**Nombre de la aplicación**: Centro de salud Acme-Health.

**Nivel de acabado en cada uno de los tres entregables:**

* Aplicación y Pruebas Unitarias: 9 puntos.
* Pruebas de integración y pruebas end-to-end: 9 puntos.
* Pruebas de rendimiento y refactorizaciones: 9 puntos.

**Tipo de aplicación:** Proyecto independiente.

## 2. Descripción a grandes rasgos de la aplicación web

Nuestra aplicación se va a centrar en la gestión de citas de una cadena de centros de salud, donde el cliente también podrá realizar la parte de pagos o asociaciones con aseguradoras o mutuas. Los profesionales especialistas podrán además añadir a las citas al acabar el diagnóstico y tratamiento conveniente.

## 3. Tabla de repartición de historias de usuario por prioridad y parejas

La tabla está actualizada, ya que a lo largo de los sprints se han ido modificando nombres de las historias de usuario y requerimiento (en el sprint 2 pasó de ser un 66% a un 50%).

|  |  |  |  |
| --- | --- | --- | --- |
| **Nº HISTORIA** | **NOMBRE** | **PAREJA** | **SPRINT** |
| 1 | ELECCIÓN DE CENTRO | ALBERTO E IVÁN | 2 |
| 2 | ASOCIACIÓN MUTUA O ASEGURADORA | JOSÉ Y PACO | 2 |
| 3 | **DATOS DE CLIENTE** | JOSE Y PACO | 2 |
| 4 | ELECCIÓN DE ESPECIALIDAD | ALBERTO E IVÁN | 2 |
| 5 | **VER PROFESIONALES POR CENTRO Y ESPECIALIDAD** | JOSÉ Y PACO | 2 |
| 6 | **DESCRIPCIÓN DE MOTIVO DE CITA** | ALBERTO E IVÁN | 2 |
| 7 | ELECCIÓN DE HORARIO | JOSÉ Y PACO | 2 |
| 8 | LISTAR CITAS | ALBERTO E IVÁN | 2 |
| 9 | GESTIONAR CITAS | ALBERTO E IVÁN | 2 |
| 10 | RECETAR MEDICAMENTOS | JOSÉ Y PACO | 2 |
| 13 | **Busqueda profesionales** | ALBERTO E IVÁN | 2 |
| 18 | **ELECCIÓN DE CONTRASEÑA** | JOSÉ Y PACO | 2 |
| 11 | VER ENFERMEDADES ANTERIORES | ALBERTO E IVÁN | 3 |
| 12 | VER TRATAMIENTOS ANTERIORES | JOSÉ Y PACO | 3 |
| 21 | PRECIO CONSULTA | JOSÉ Y PACO | 3 |
| 19 | LISTAR CITAS CLIENTES | ALBERTO E IVÁN | 3 |
| 20 | **MÉTODOS DE PAGO Repetidos** | JOSÉ Y PACO | 3 |
| 15 | **PAGOS CON TARJETA** | ALBERTO E IVÁN | 3 |
| 16 | DIAGNÓSTICO MÉDICO | JOSÉ Y PACO | 3 |
| 17 | PANEL DE CONTROL | JOSÉ Y PACO | 4 |
| 14 | FACTURAS | JOSÉ Y PACO | 4 |

## 4. Historias de usuario, escenarios e informes de las pruebas de rendimiento, profiling y refactoring

* 1. . **HU-001: Elección de centro**

**Como** cliente,

**Para** poder elegir el centro que esté más cerca en cada momento,

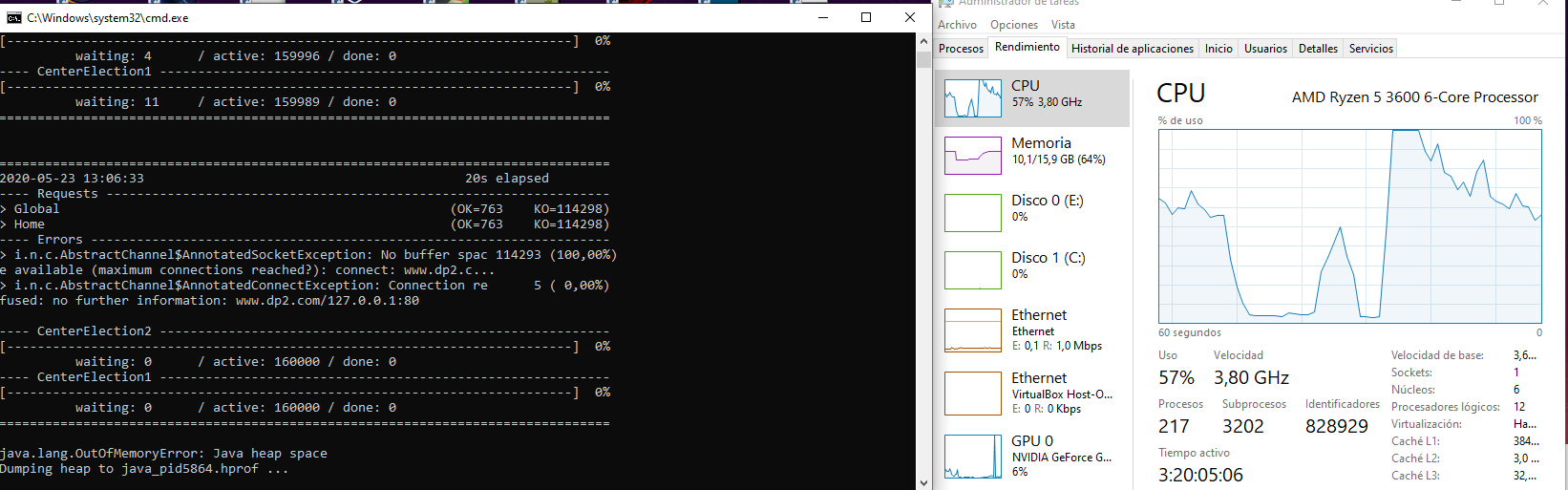
**Quiero** poder seleccionar en qué centro pedir la cita.

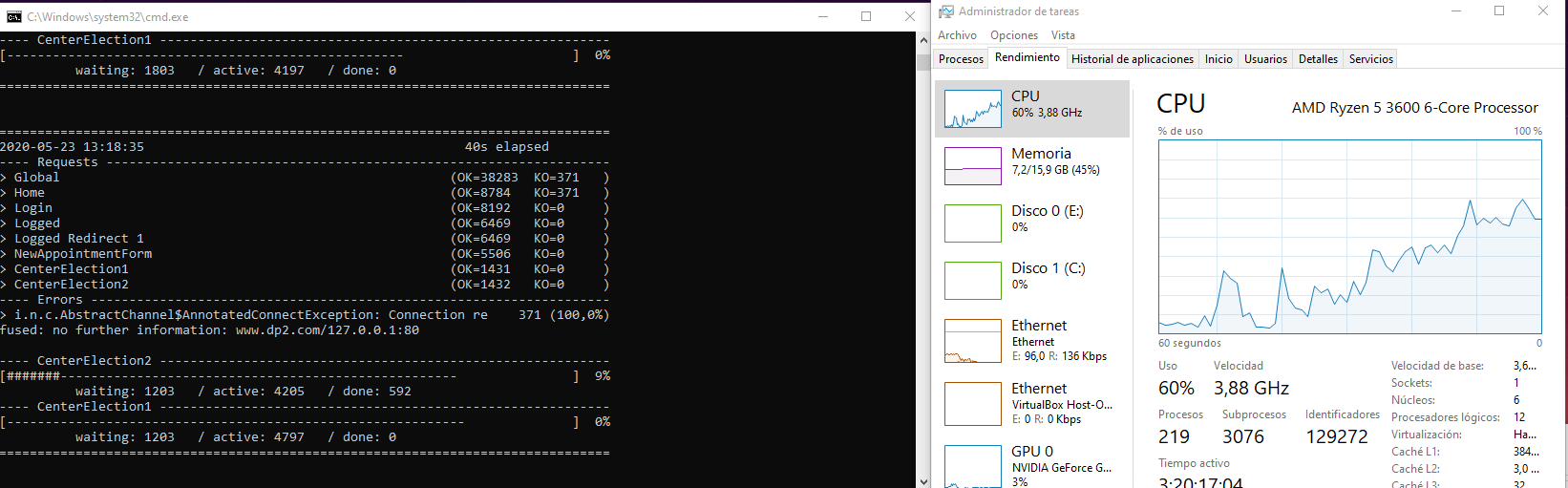
✔ Pepe Gotera al pedir cita elige el centro de Sevilla y el sistema le deja seguir sin problemas.

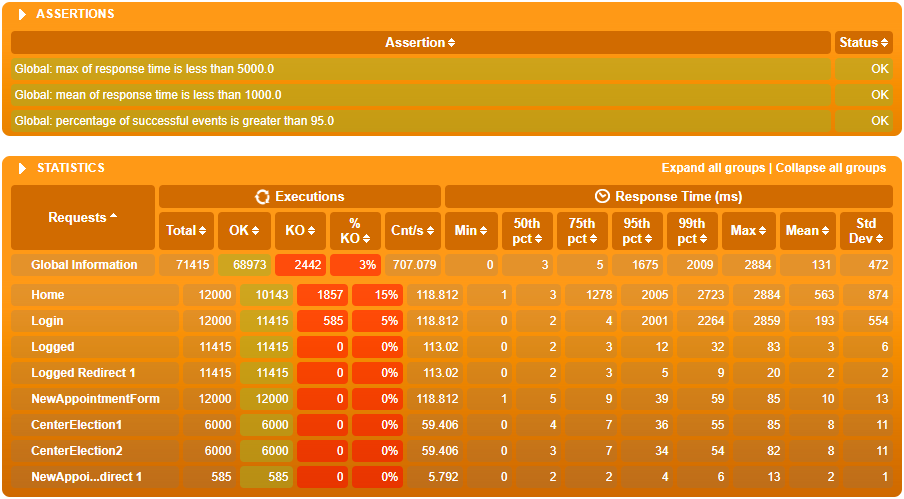
✘ Elena Nito al pedir cita se le olvida elegir centro, entonces el sistema le muestra el error de que debe elegir un centro de los disponibles.

**Entidades involucradas:** Centro, Cliente.

**Número mínimo de usuarios que no son soportados:** 320.000 en 10 segundos.



**Número máximo de usuarios simultáneos soportados:** 9.002 en 50 segundos.



* 1. . **HU-002: ASOCIACIÓN CON MÚTUA O ASEGURADORA**

**Como** administrador del sistema,

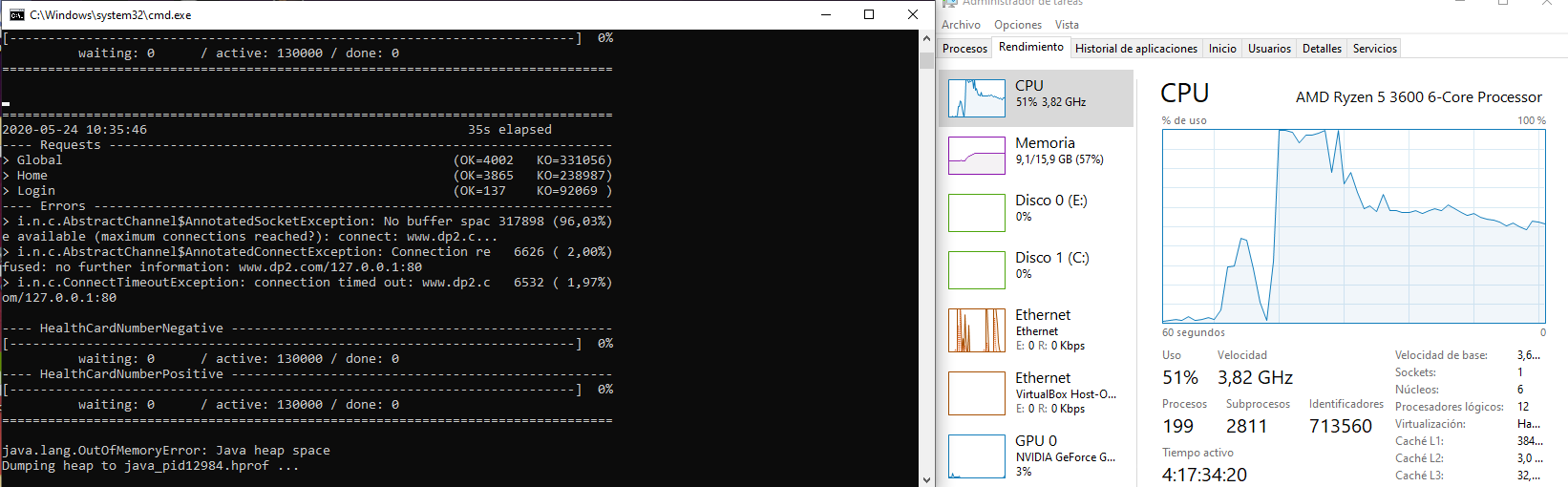
**Para** ofrecer un servicio de acuerdo con el estado de cada cliente,

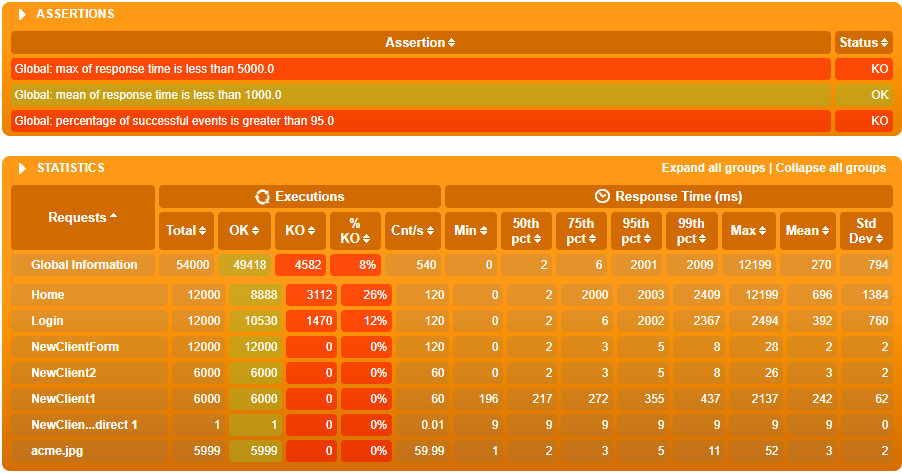
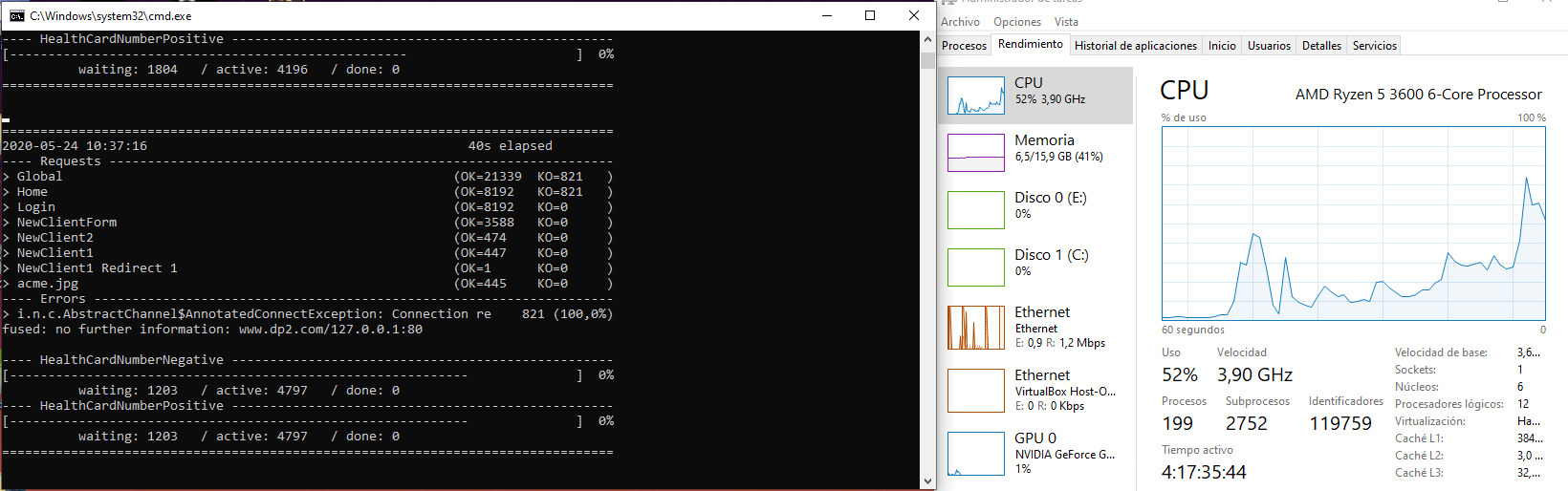
**Quiero** saber si el cliente que solicita una cita dispone de mutua o aseguradora, si procede.

✔ Pepe Gotera al registrarse en el sistema escoge elige la aseguradora Adeslas y escribe el número de tarjeta sanitaria, y el sistema le deja seguir sin problemas.

✘ Elena Nito al registrarse en el sistema elige la mutua Sanitas, pero no establece el número de tarjeta sanitaria, por lo que el sistema le muestra el error de que debe escribir el número de tarjeta sanitaria.

**Entidades involucradas:** Cliente.

 **Número mínimo de usuarios que no son soportados:** 260.000 en 10 segundos.

******Número máximo de usuarios simultáneos soportados:** 9.594 en 50 segundos.

* 1. . **HU-003: DATOS DE CLIENTE**

**Como** administrador del sistema,

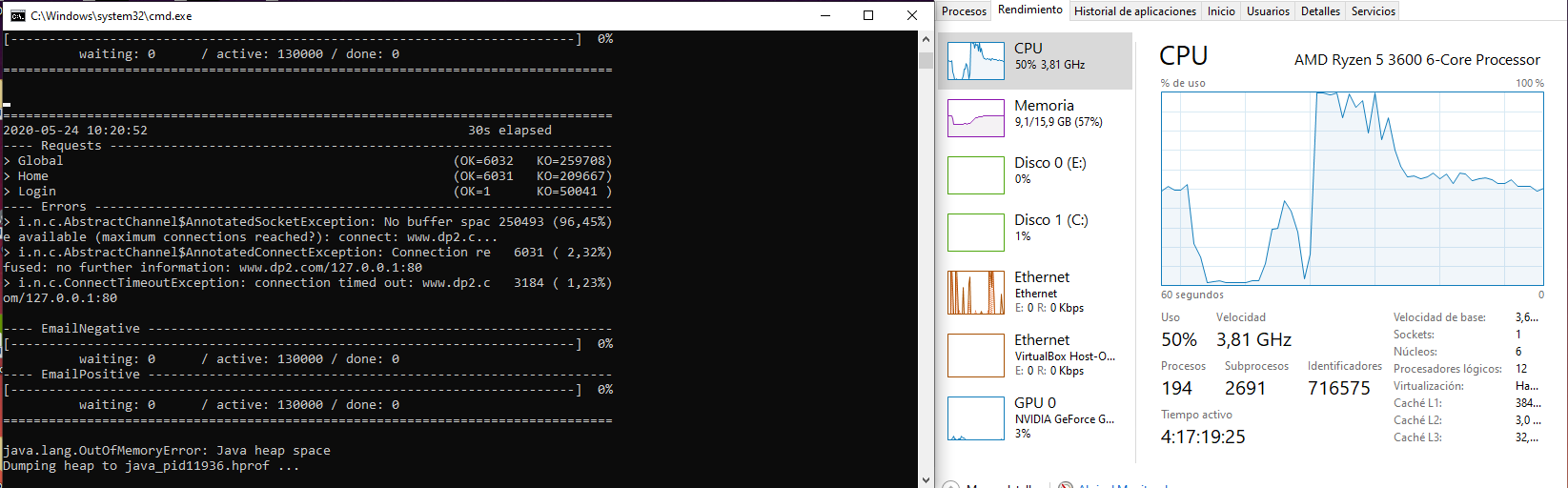
**Para** notificar las citas de cada cliente,

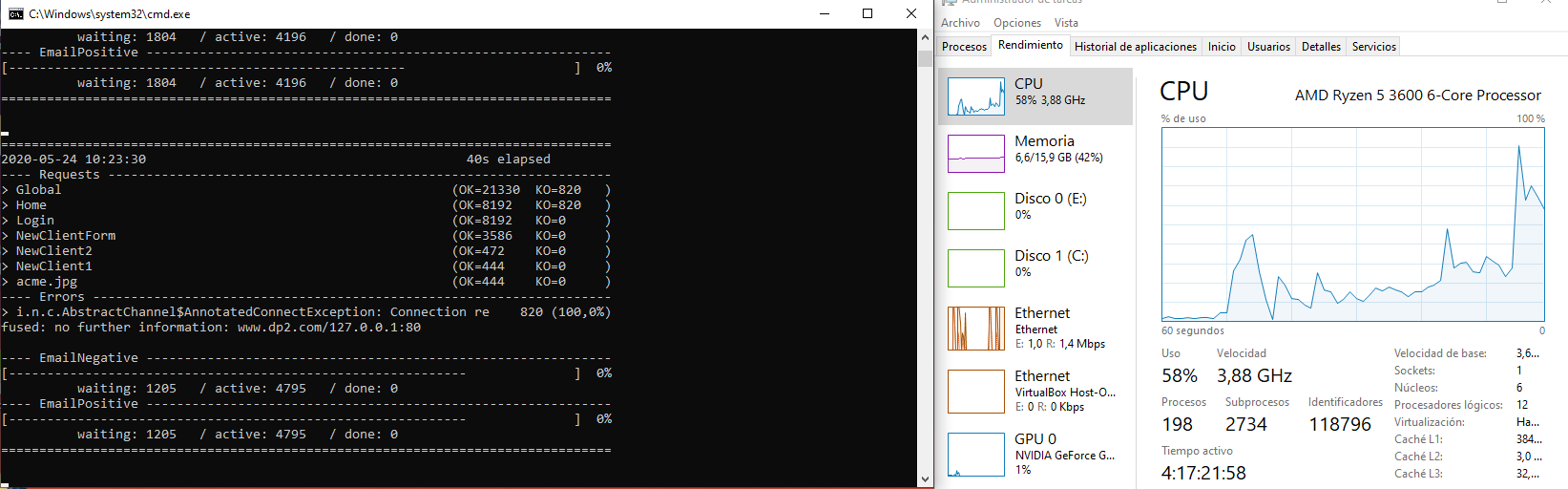
**Quiero** conocer los datos personales, tales como nombre, apellidos, fecha de nacimiento y número de teléfono, y además correo electrónico de los clientes.

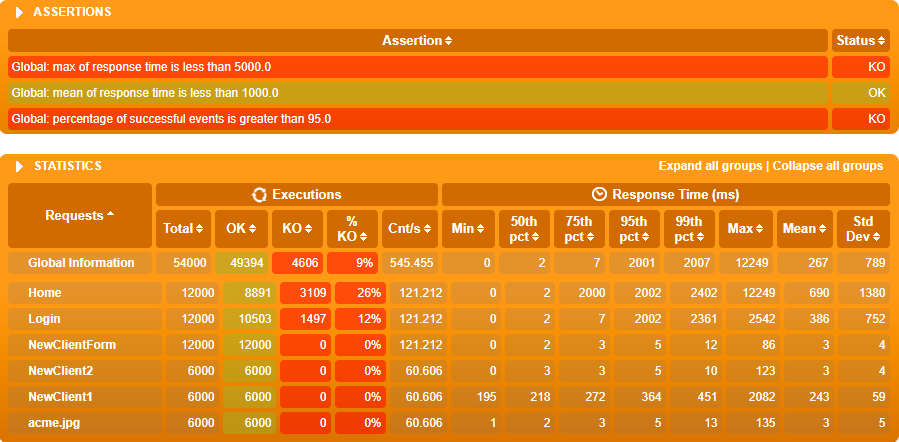
✔ Luis Pérez se registra rellenando todos sus datos personales correctamente.

✘ María Sanz al registrarse se olvida de rellenar el campo de correo electrónico y se produce un error.

**Entidades involucradas:** Cliente.

**Número mínimo de usuarios que no son soportados:** 260.000 en 10 segundos.

**Número máximo de usuarios simultáneos soportados:** 9.590 en 50 segundos.



* 1. . **HU-004: ELECCIÓN De especialidad**

**Como** cliente,

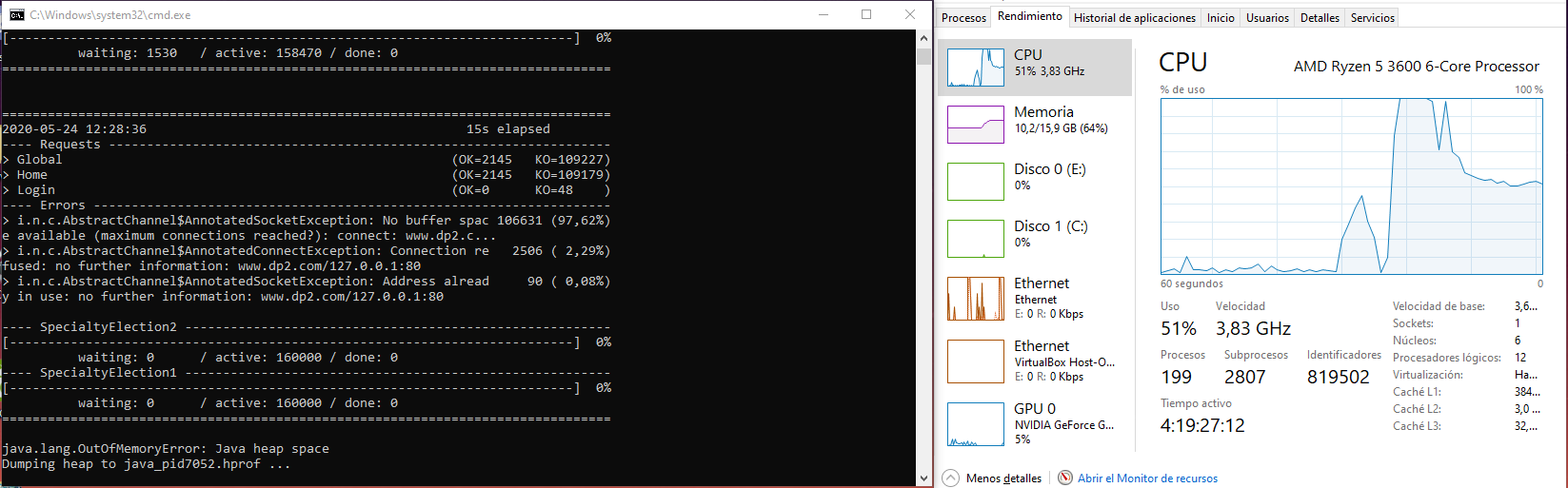
**Para** poder elegir un profesional cualificado según mis necesidades,

**Quiero** poder filtrar por especialidad a los médicos al pedir una cita.

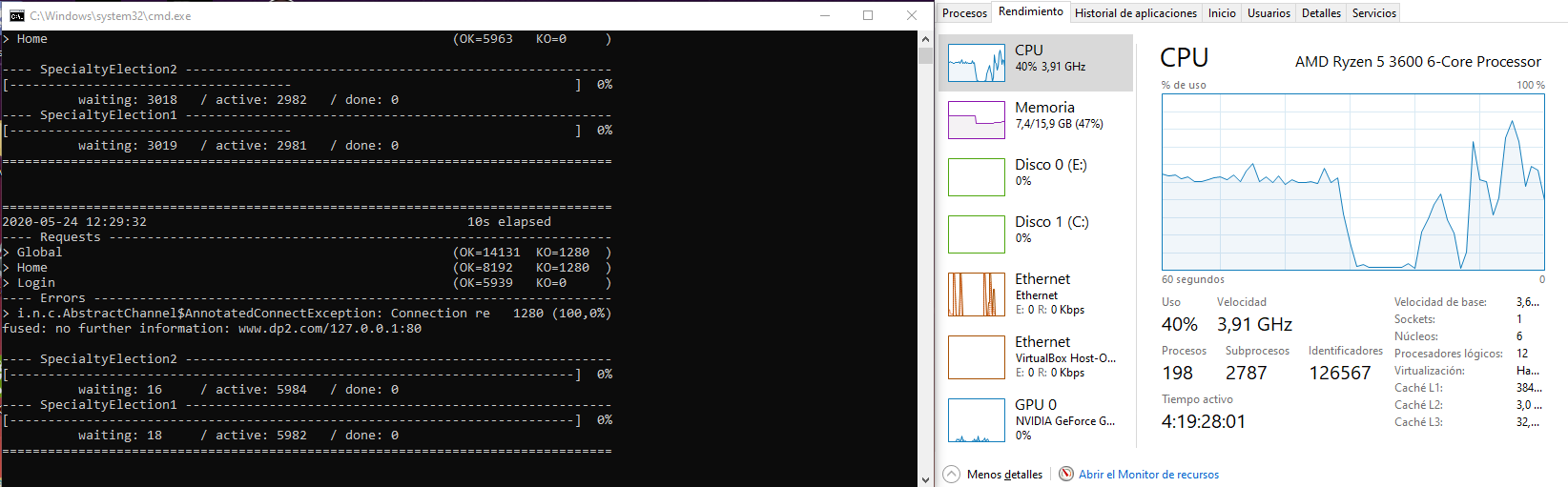
✔ Pepe Gotera al crear una cita en Sevilla filtra según dermatología y le aparecen solo los profesionales propios de la esta.

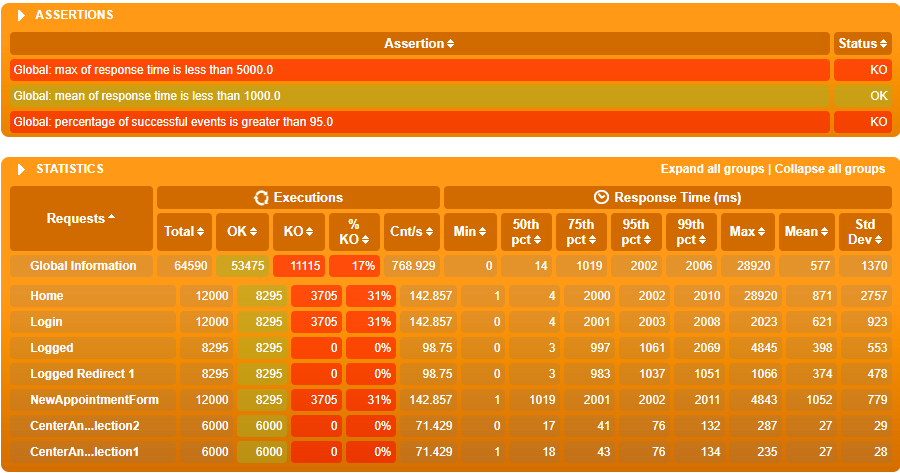
✘ Elena Nito al crear una cita en Sevilla se le olvida elegir especialidad, entonces al intentar avanzar el sistema le mostrará el error de que debe elegir una especialidad válida.

**Entidades involucradas:** Cita, Especialidad, Centro.

**Número mínimo de usuarios que no son soportados:** 260.000 en 10 segundos.

**Número máximo de usuarios simultáneos soportados:** 11.966 en 50 segundos.

****



* 1. . **HU-005: VER PROFESIONALES POR CENTRO Y ESPECIALIDAD**

**Como** cliente,

**Para** seleccionar el profesional que más me convenga,

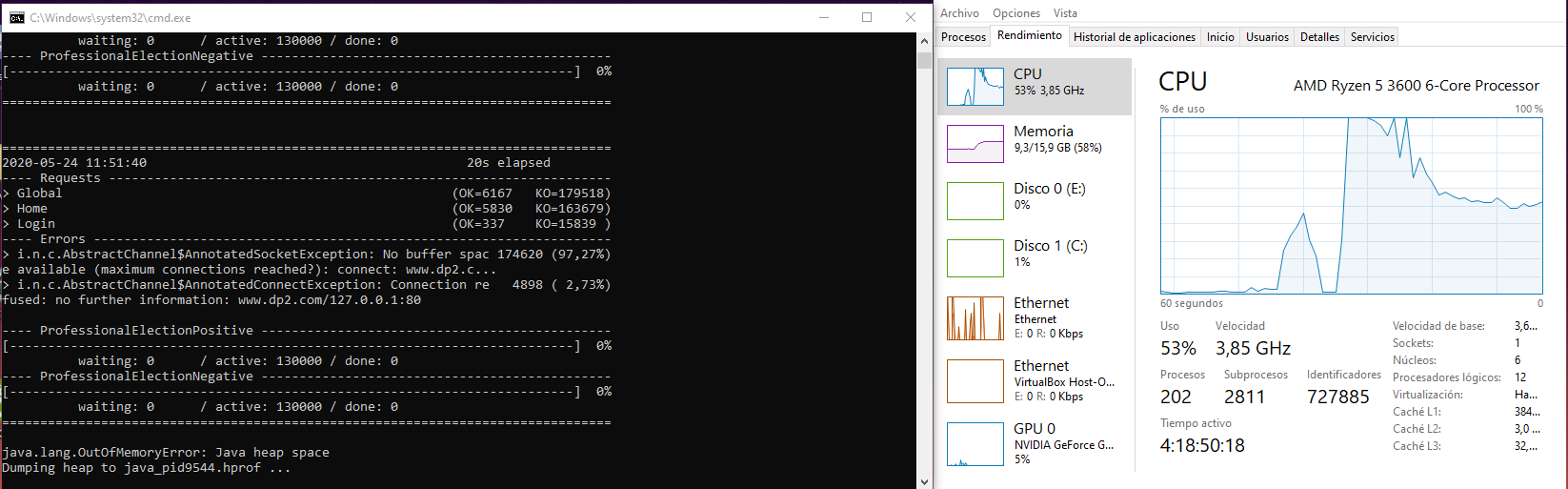
**Quiero** obtener una lista con los profesionales que se encuentren en la ciudad que desee y que estén especializados en la rama de la medicina seleccionada.

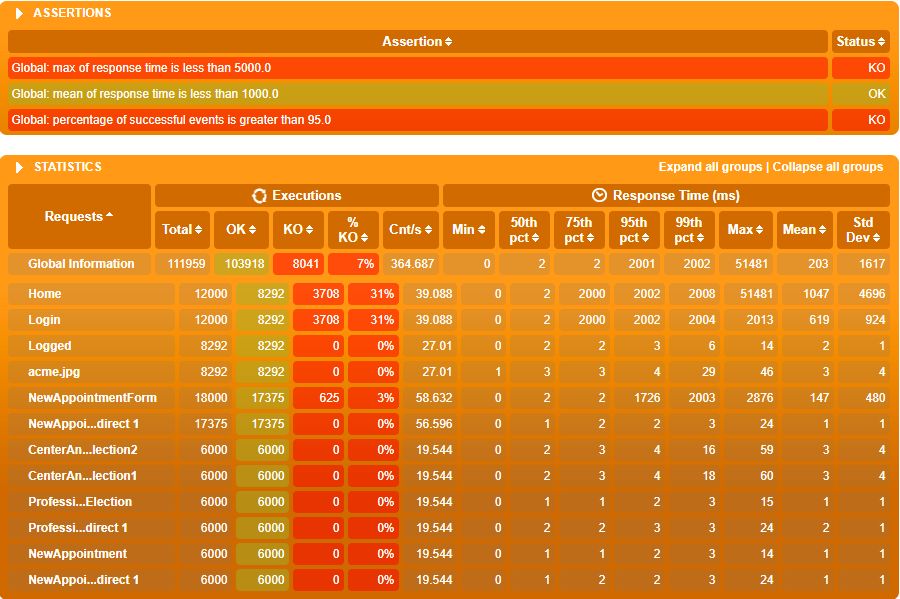
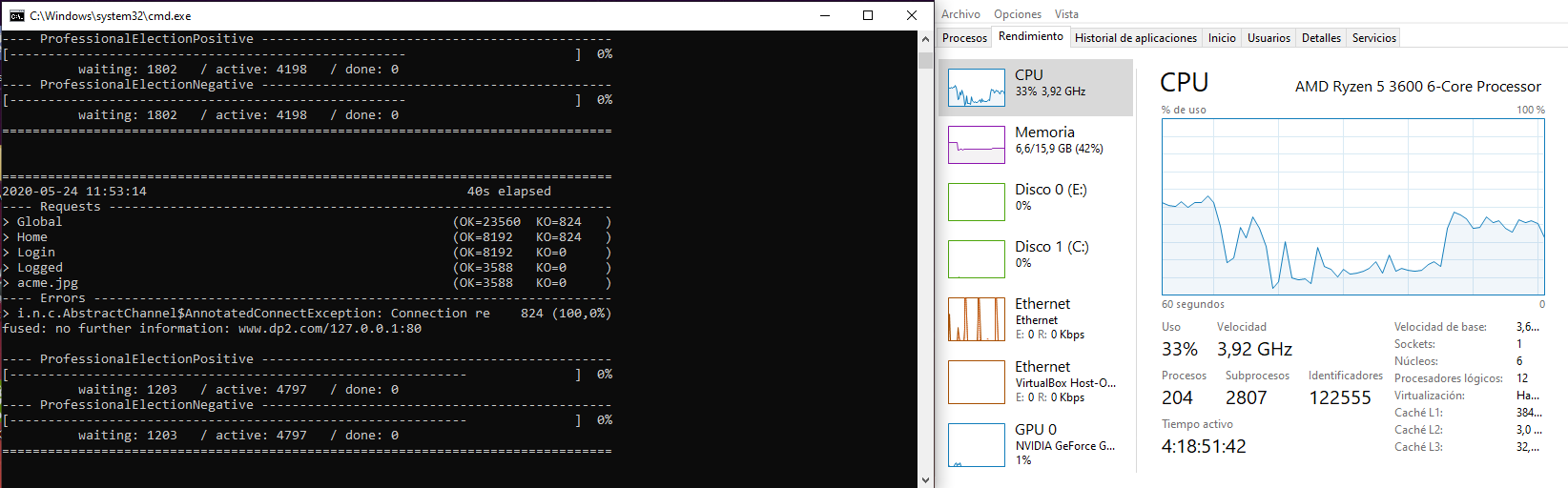
✔ Pepe Gotera al crear una cita elige la ciudad de Sevilla, la especialidad de dermatología y tras esto selecciona al profesional Guillermo Díaz.

✘ Elena Nito al crear una cita elige la ciudad de Cádiz y la especialidad de cirugía, tras esto, intenta seleccionar un profesional, pero no aparece ninguno.

**Entidades involucradas:** Cita, Centro, Profesional, Especialidad.

**Número mínimo de usuarios que no son soportados:** 260.000 en 10 segundos.

****

**Número máximo de usuarios simultáneos soportados:** 9.594 en 50 segundos.

* 1. **. HU-006: DESCRIPCIÓN DE MOTIVO DE CITA**

**Como** cliente,

**Para** que el profesional tenga más información sobre mí cita,

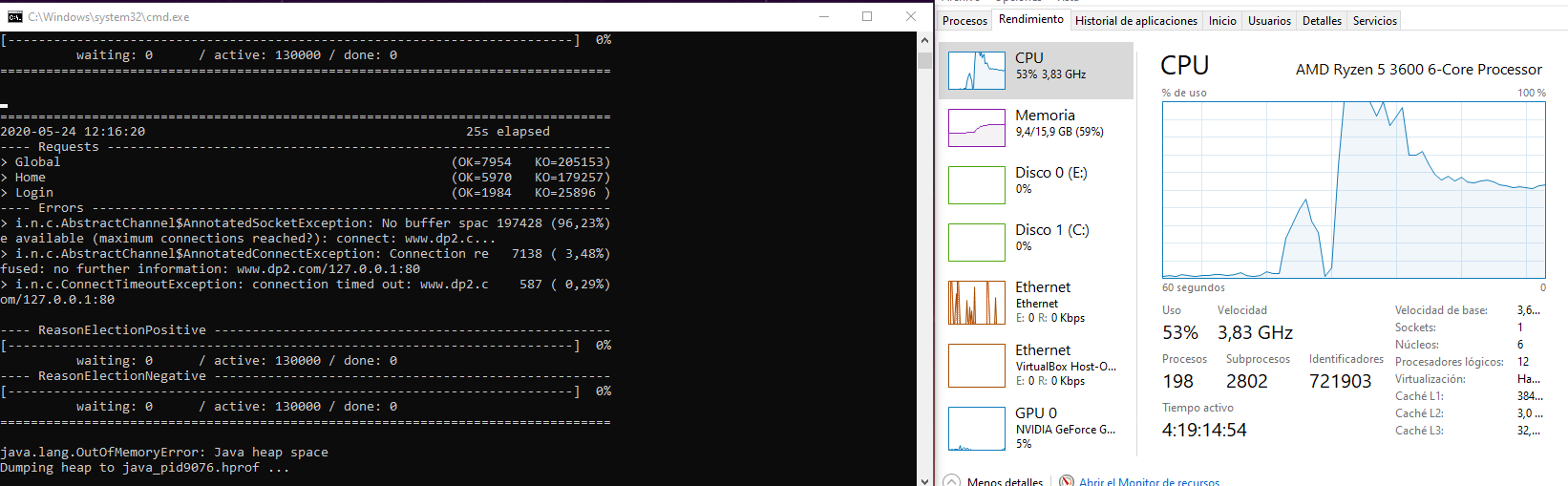
**Quiero** especificar el motivo por el que realizo mí cita.

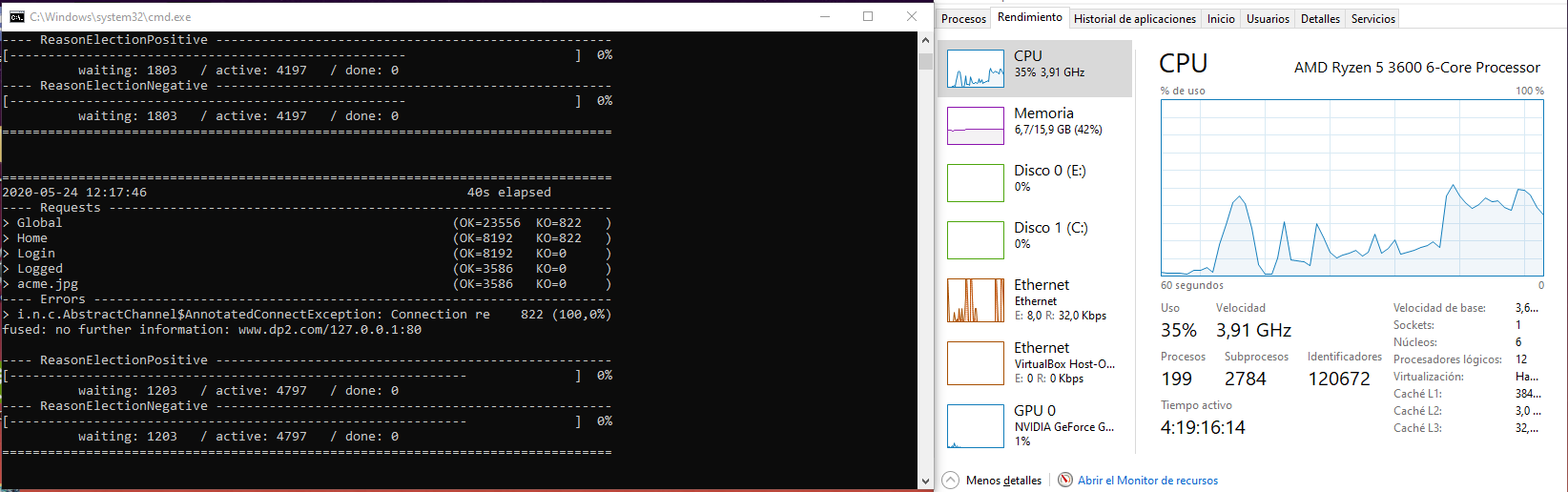
✔ Pepe Gotera al crear una cita con Guillermo Díaz específica como motivo dolor abdominal.

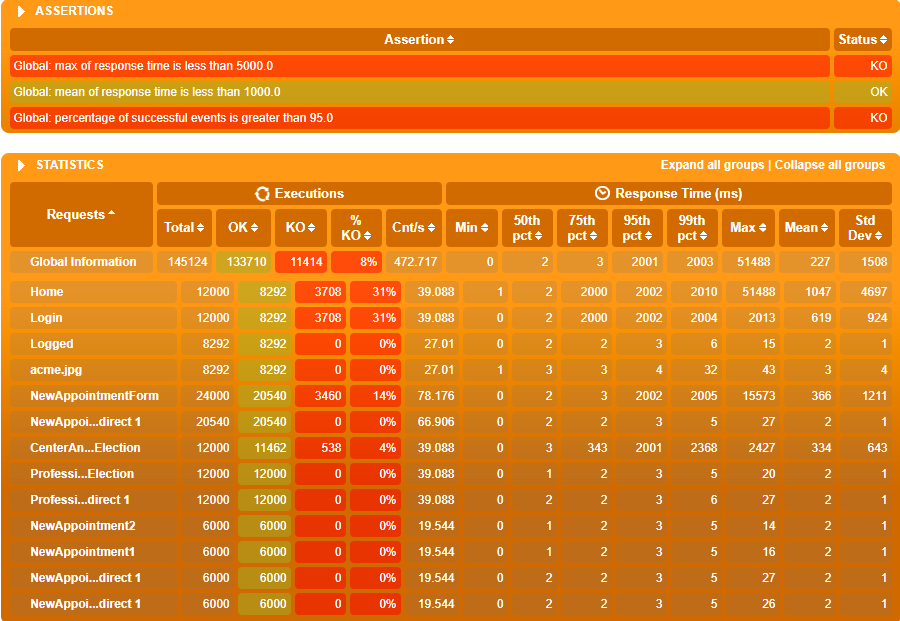
✘ Elena Nito al crear una cita con Guillermo Díaz no específica nada como motivo de cita, dejando el campo en blanco, entonces el sistema le muestra el mensaje de que tiene que elegir un motivo.

**Entidades involucradas:** Cita.

**Número mínimo de usuarios que no son soportados:** 260.000 en 10 segundos.



 **Número máximo de usuarios simultáneos soportados:** 9.594 en 50 segundos.



* 1. . **HU-007: ELECCIÓN DE HORARIO**

**Como** cliente,

**Para** tener una mayor comodidad a la hora de pedir la cita según me convenga,

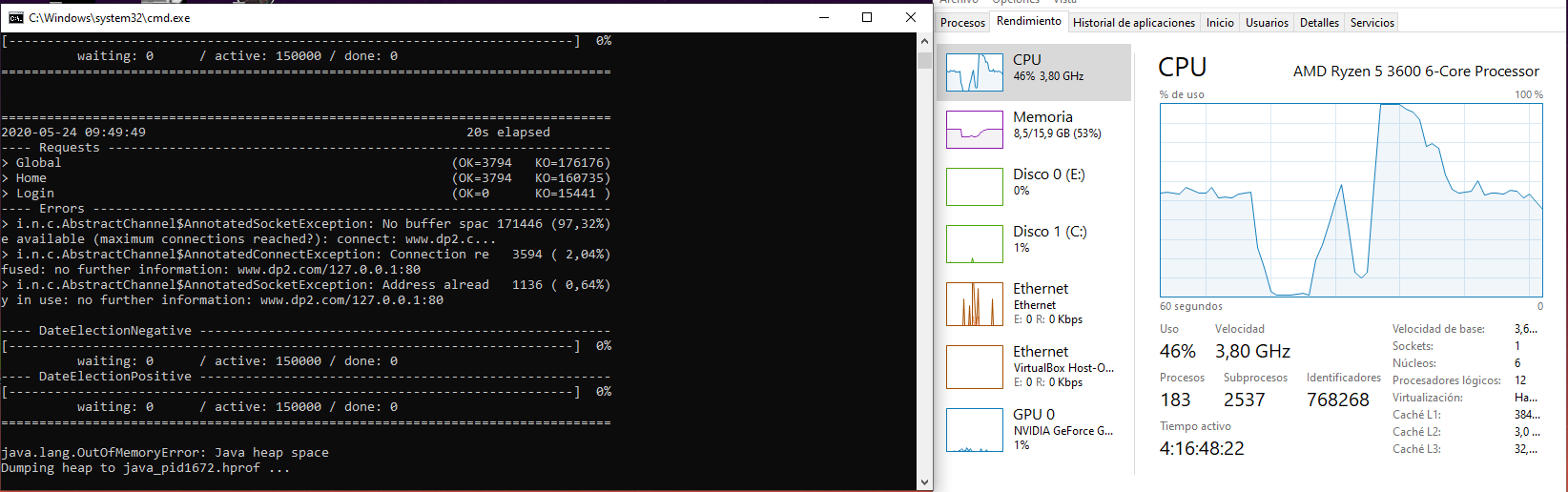
**Quiero** poder seleccionar el horario de cita del profesional que desee, según disponibilidad.

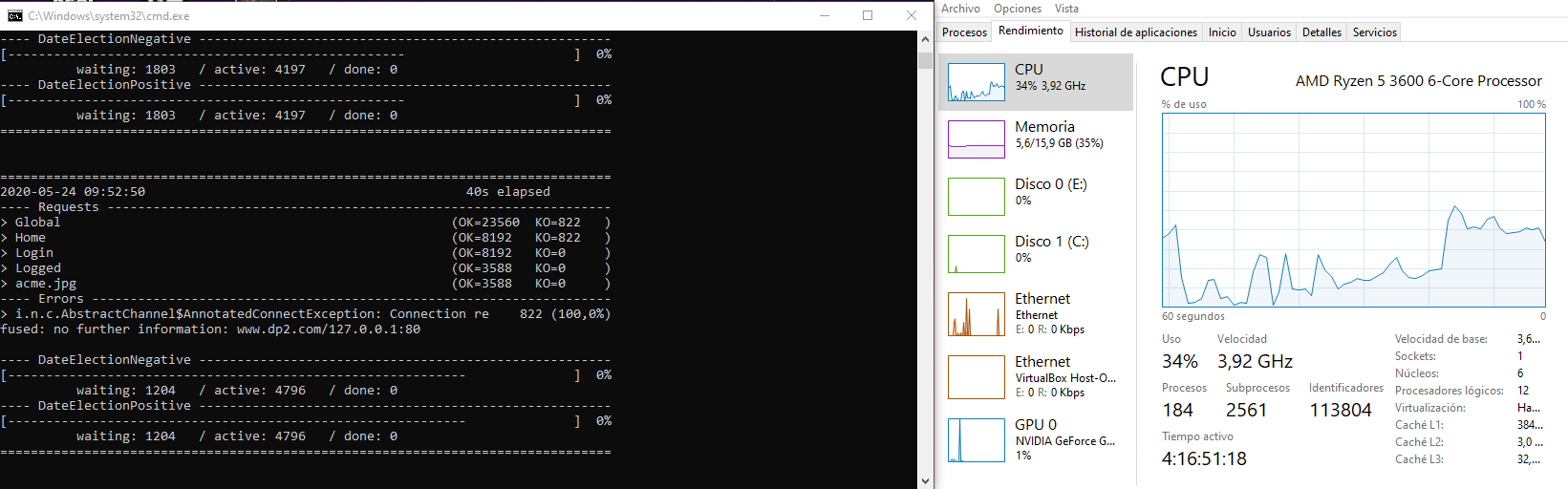
✔ Pepe Gotera al crear una cita con Guillermo Díaz escoge la fecha para el 12/02/2021 a las 16:15.

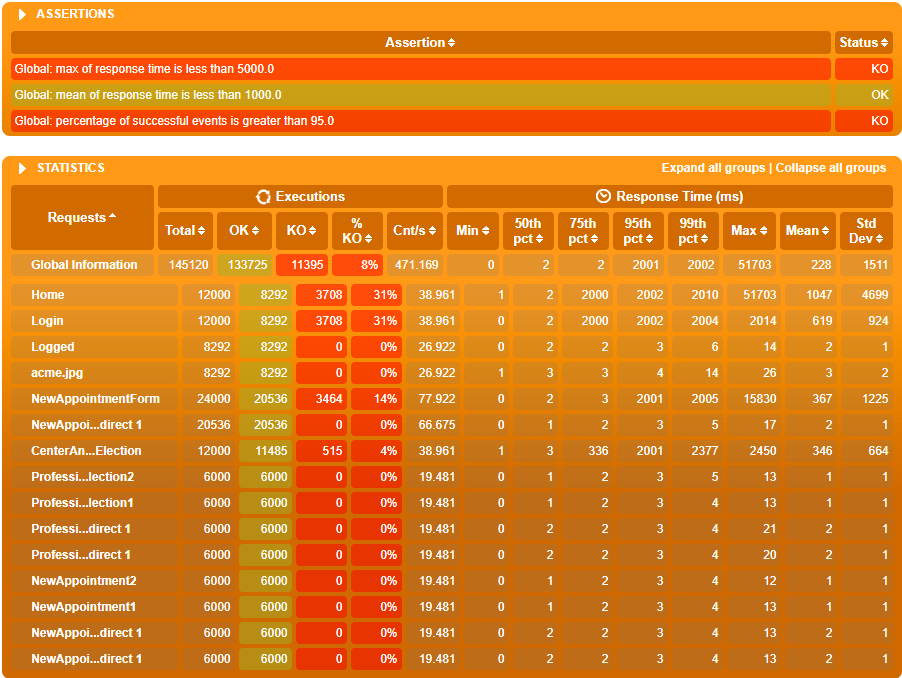
✘ Elena Nito al crear una cita con Guillermo Díaz escribe una fecha pasada, como el 10/10/2019 y el sistema le muestra el error describiendo que tiene que elegir un día válido.

**Entidades involucradas:** Cita, Profesional, Horario.

**Número mínimo de usuarios que no son soportados:** 300.000 en 10 segundos.

****

**Número máximo de usuarios simultáneos soportados:** 9.592 en 50 segundos.



* 1. . **HU-008: LISTAR CITAS**

**Como** médico,

**Para** saber cuáles son las próximas citas y organizarme mejor,

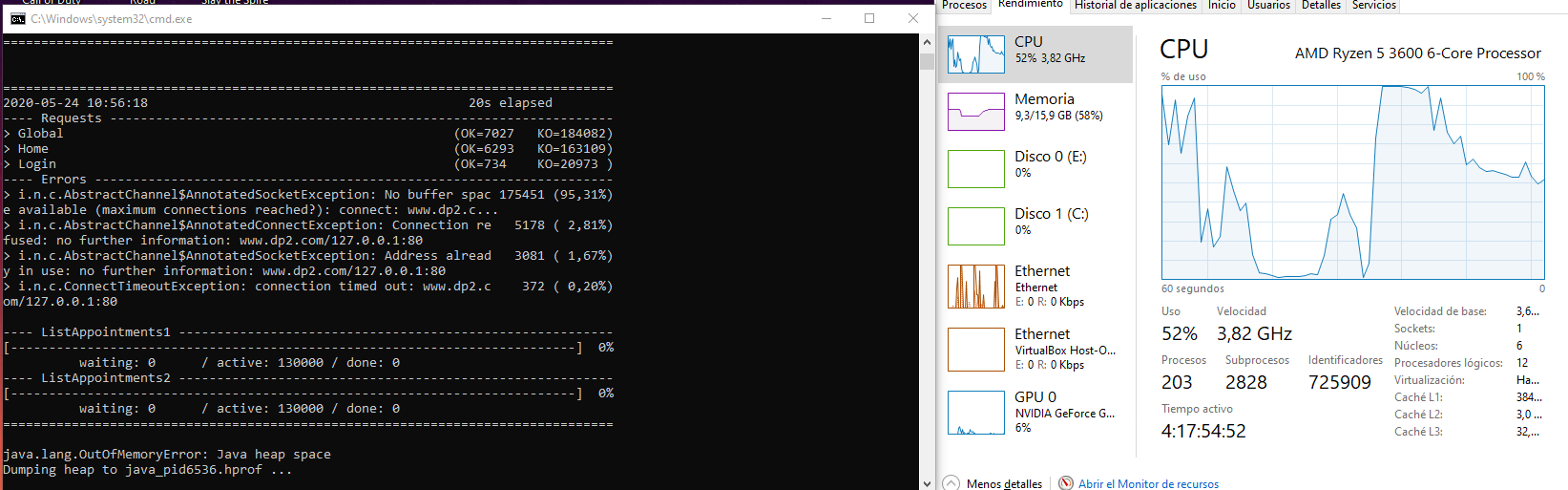
**Quiero** poder visualizar las citas del día.

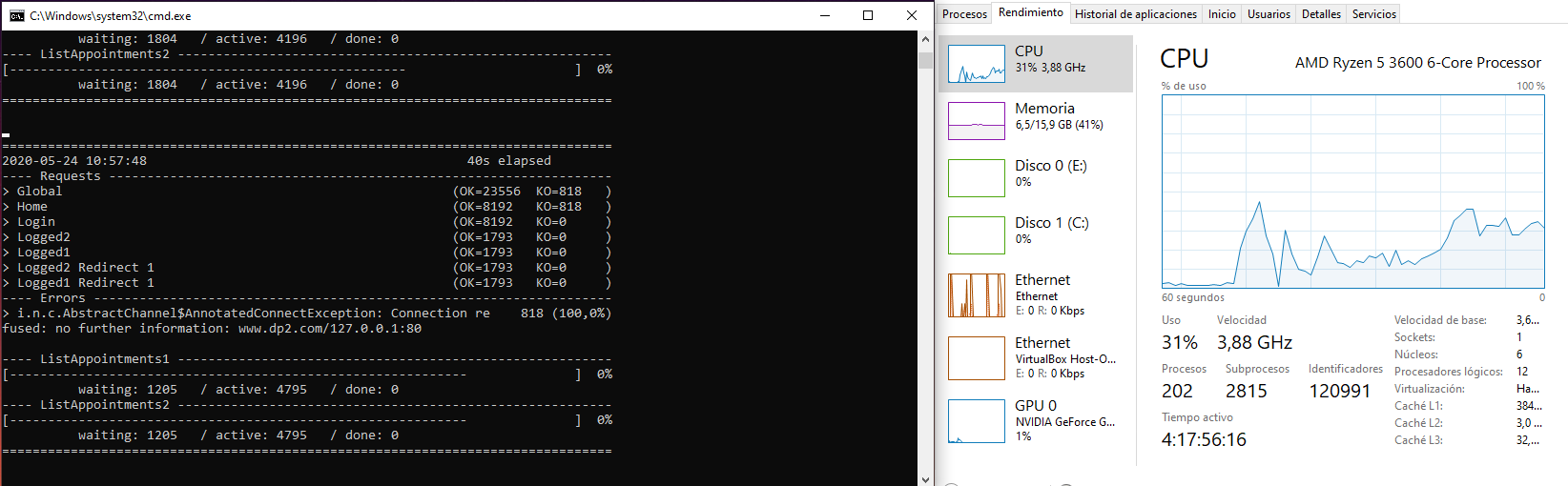
✔ Guillermo Díaz lista las citas pendientes del día de hoy y el sistema se las muestra correctamente.

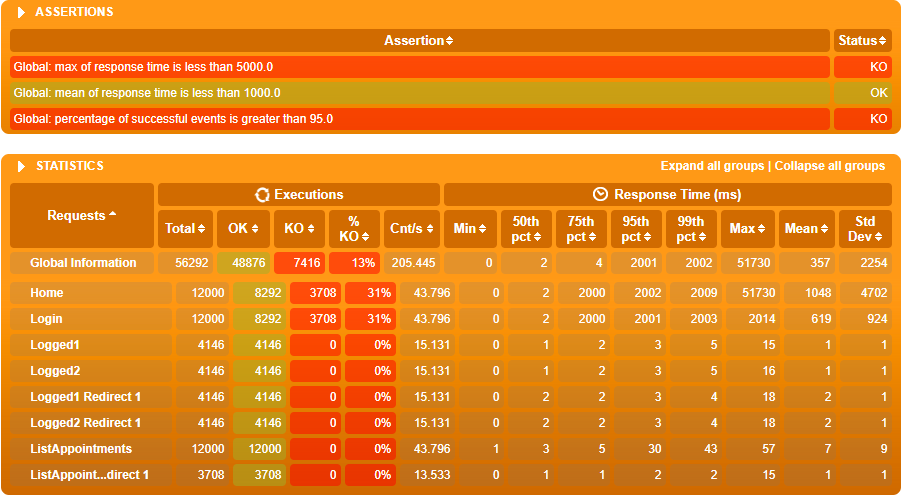
✘ Guillermo Díaz intenta listar las citas ya realizadas como próximas, pero el sistema no se lo permite.

**Entidades involucradas:** Cita, Profesional.

**Número mínimo de usuarios que no son soportados:** 260.000 en 10 segundos.

****

**Número máximo de usuarios simultáneos soportados:** 9.590 en 50 segundos.



* 1. . **HU-009: GESTIONAR CITAS**

**Como** cliente,

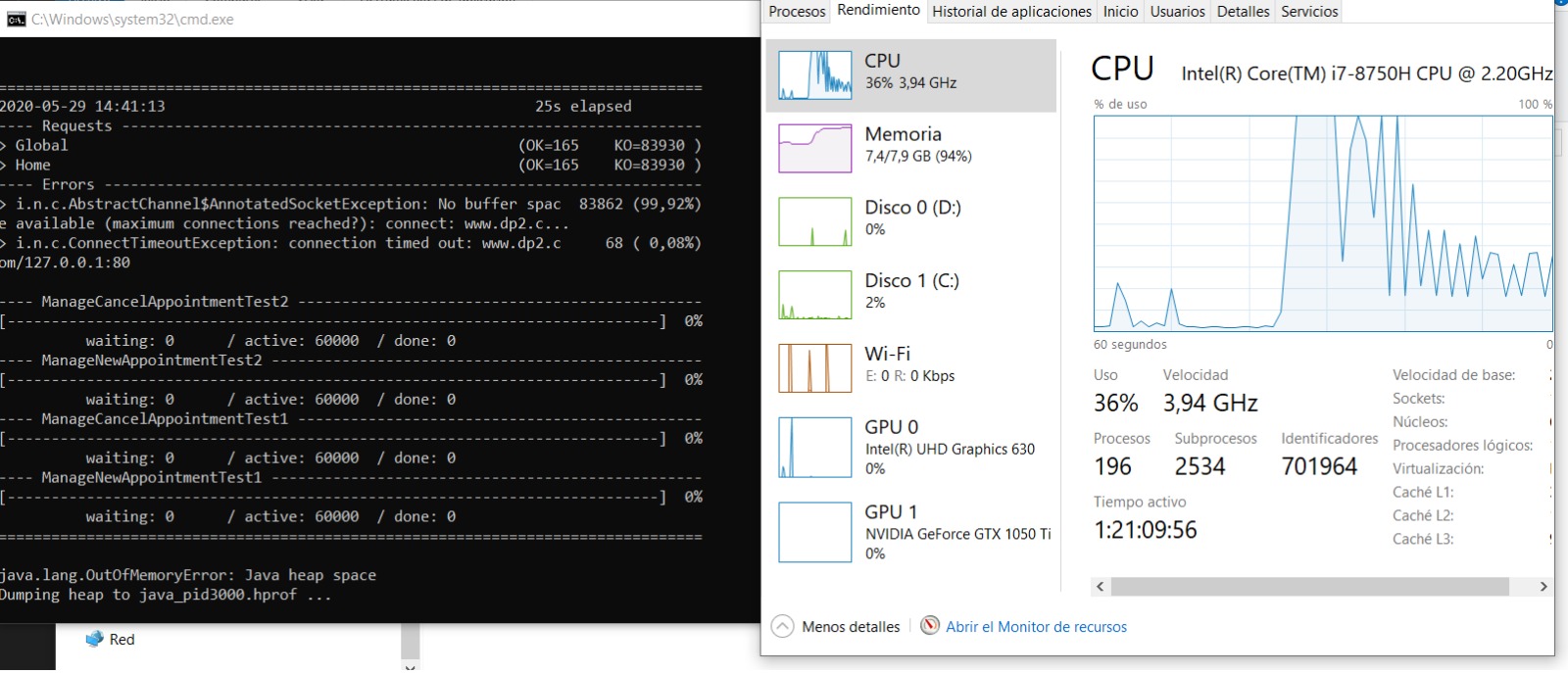
**Para** tener un mayor control sobre mis citas,

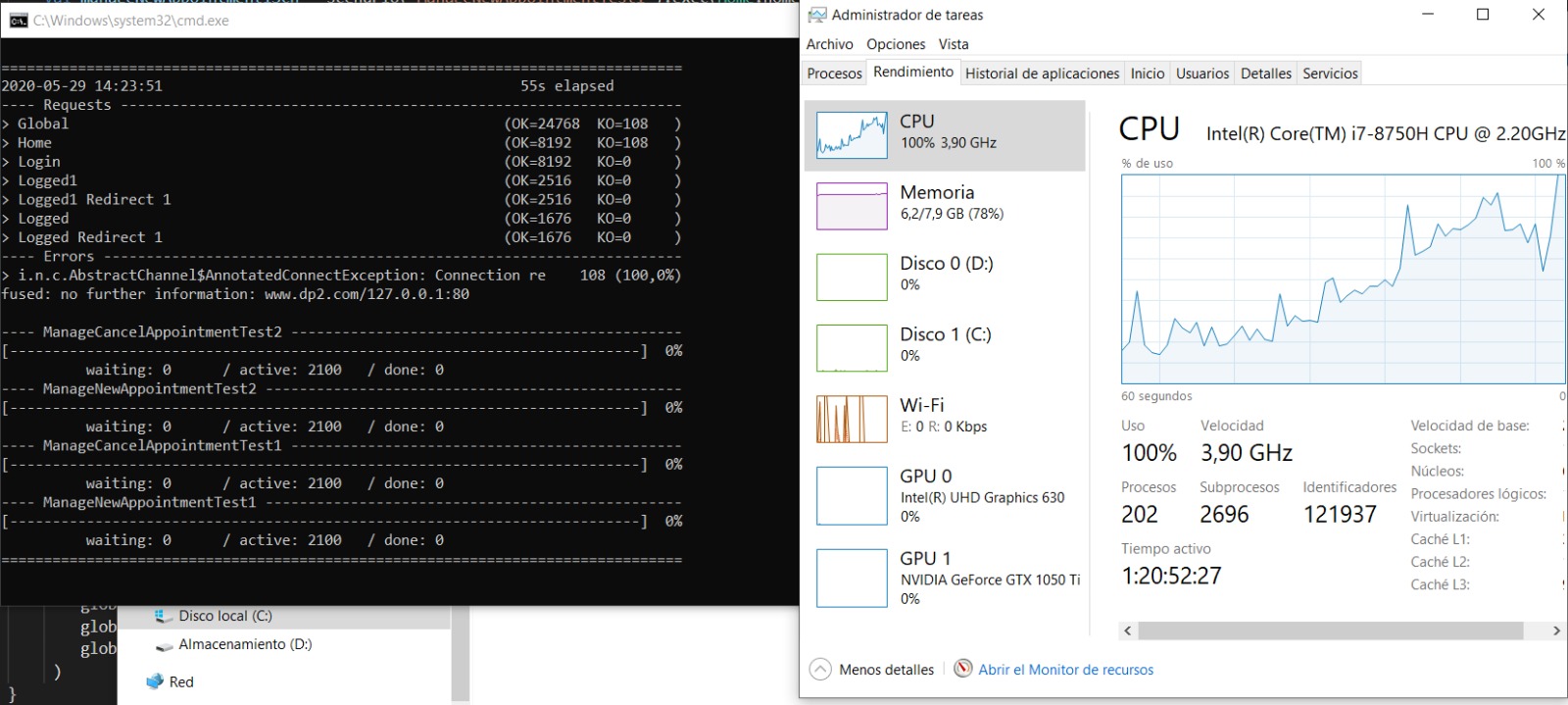
**Quiero** gestionar, incluyendo crear o cancelar citas.

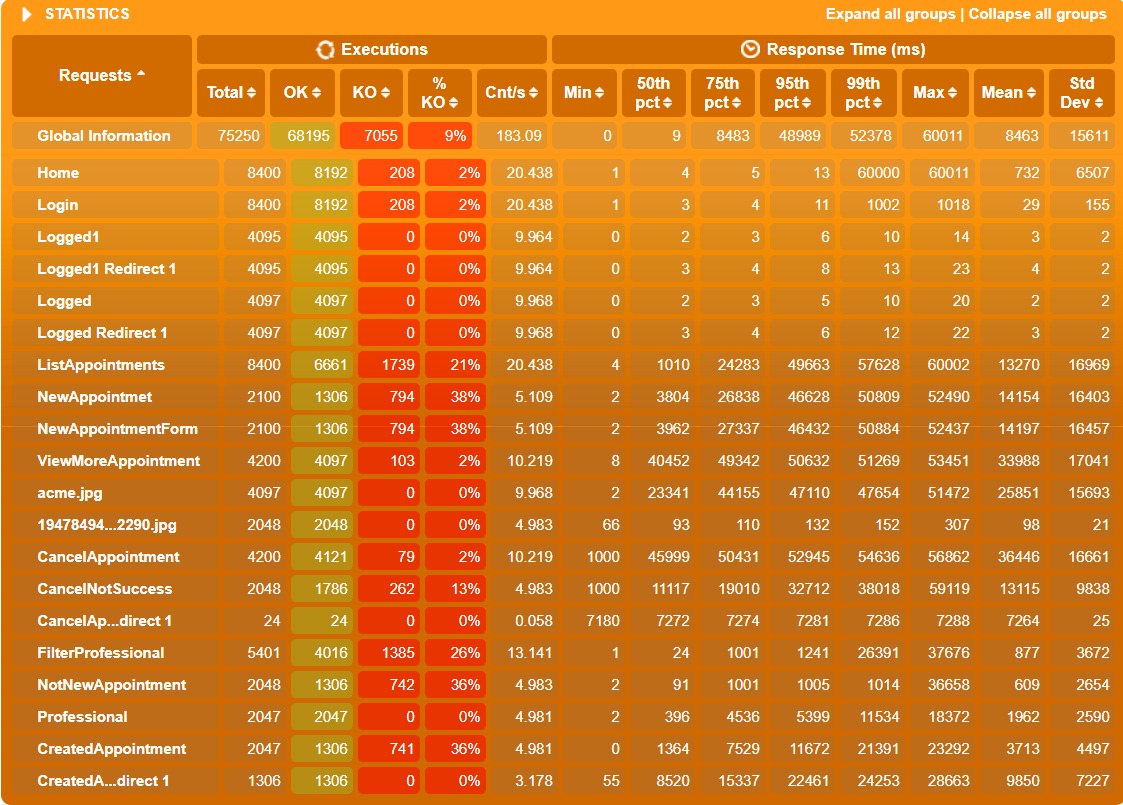
✔ Pepe Gotera cancela una cita futura, y esta se elimina correctamente.

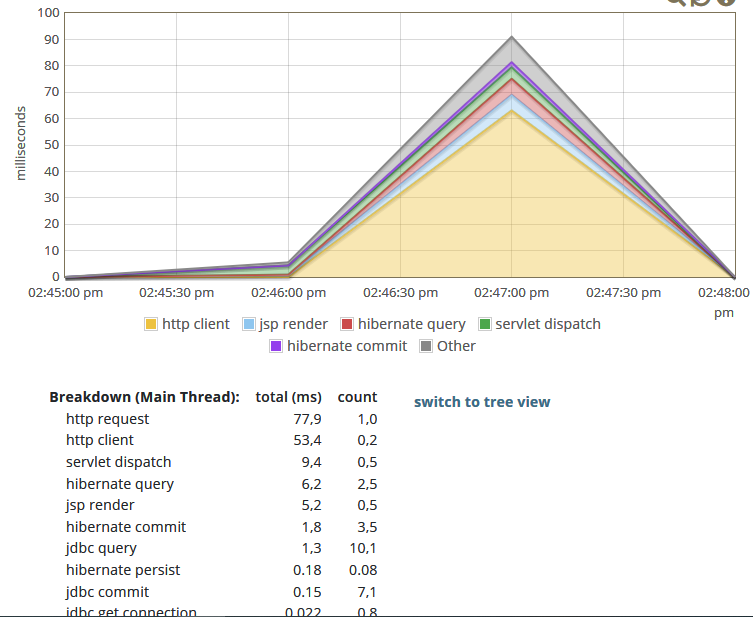
✘ Pepe Gotera intenta cancelar una cita que ya ha sido revisada por un profesional, y el sistema no se lo permite.

**Entidades involucradas:** Cita, Cliente.

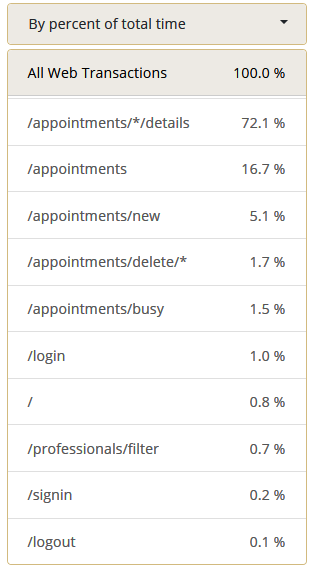
**Número mínimo de usuarios que no son soportados:** 240.000 en 10 segundos.

**Número máximo de usuarios simultáneos soportados:** 8400 en 50 segundos.

****En esta historia de usuario se ha creado una cita para posteriormente eliminarla. Tras realizar esta acción en la aplicación obtenemos estos datos en Glowroot.



Como podemos observar, lo que más tiempo ha consumido es la consulta los detalles de la cita del cliente. Tras investigar las diferentes transacciones que se han producido, se ha encontrado que en la vista /appointments/\*/details es donde se encuentra ese consumo de tiempo exagerado respecto a las demás transacciones.



Tras seleccionar dicha transacción vemos que el http client ocupa el 95% del tiempo de la transacción. Navegando por las distintas pestañas que ofrece glowroot encontramos que lo que consume tanto tiempo es la llamada al servicio de de la API Stripe, en la que se obtiene el método de pago del cliente registrado, que se usará para pagar la factura.

Tras analizar el caso negativo de esta historia de usuario no se ha obtenido ningún valor fuera de lo esperado, ya que los datos son casi idénticos al caso positivo.

* 1. . **HU-010: RECETAR MEDICAMENTOS**

**Como** médico,

**Para** poder recetar medicamentos a los pacientes y adjuntarlos en la cita,

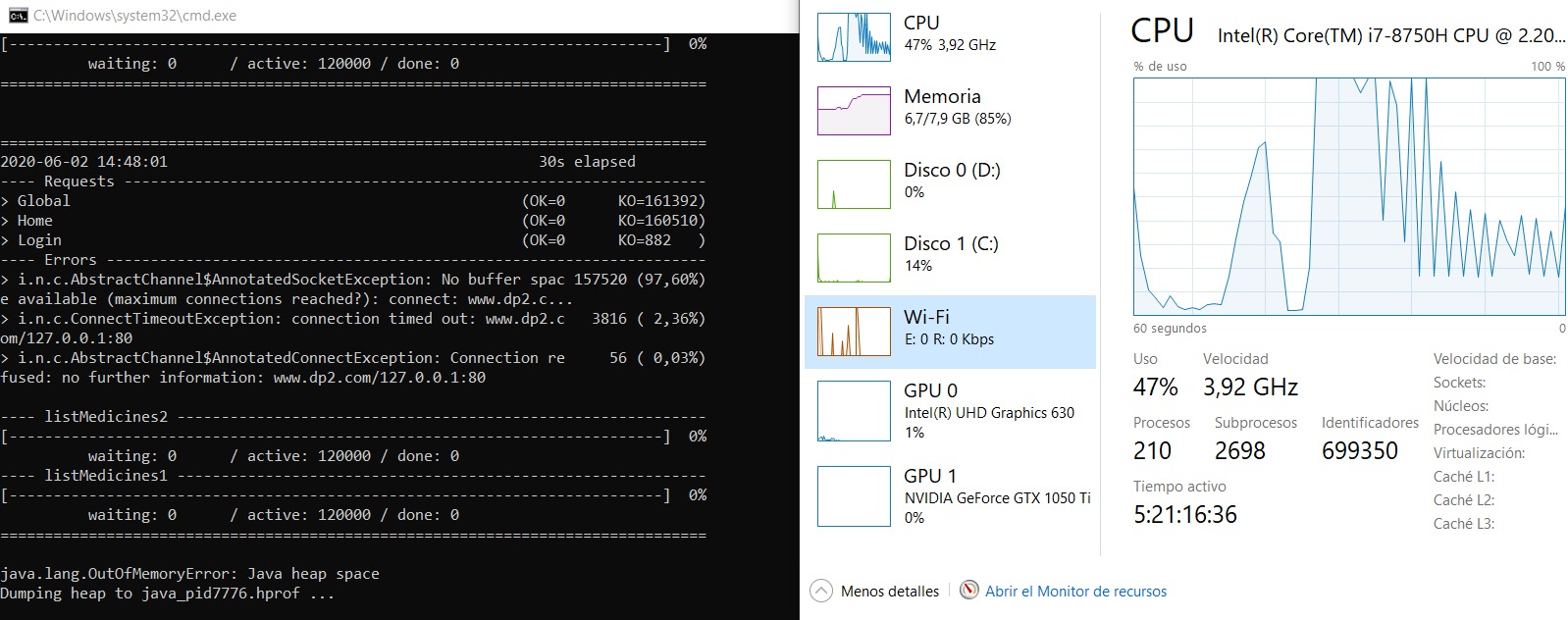
**Quiero** ver una lista de estos y poder elegir el que quiero.

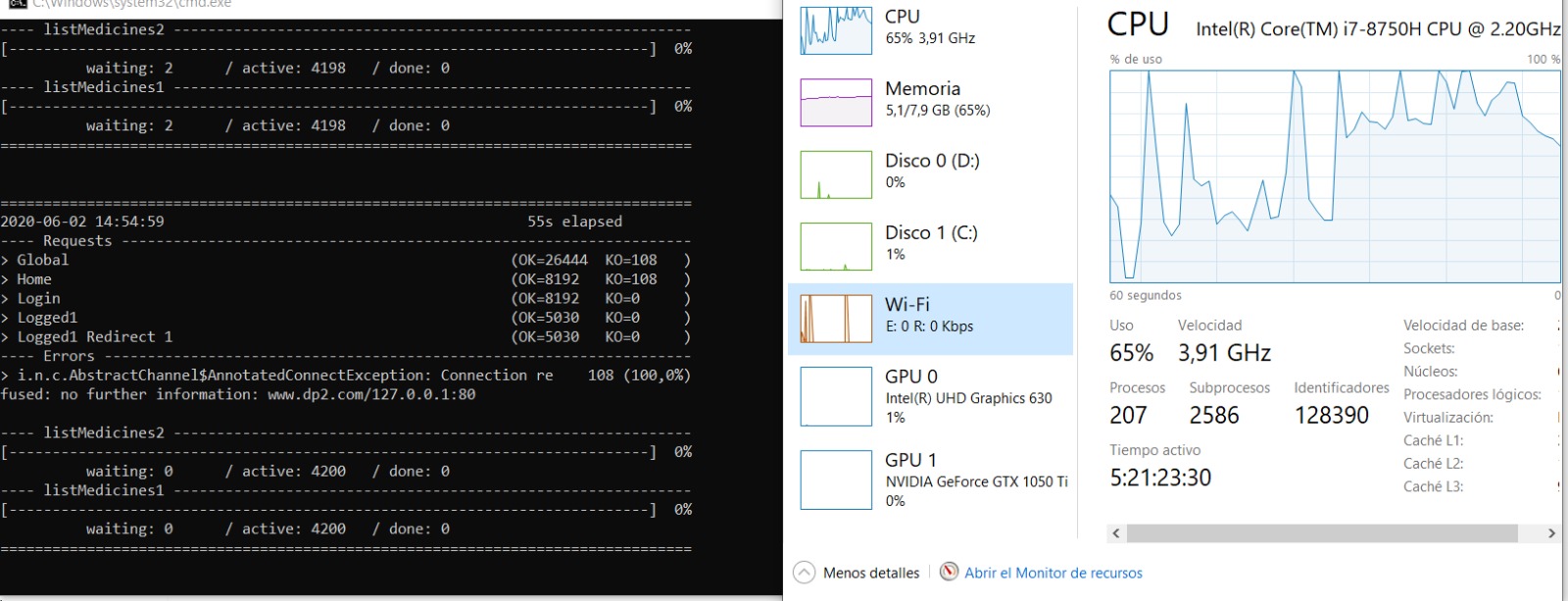
✔ Guillermo Díaz adjunta el nombre del medicamento Paracetamol al diagnóstico de la cita.

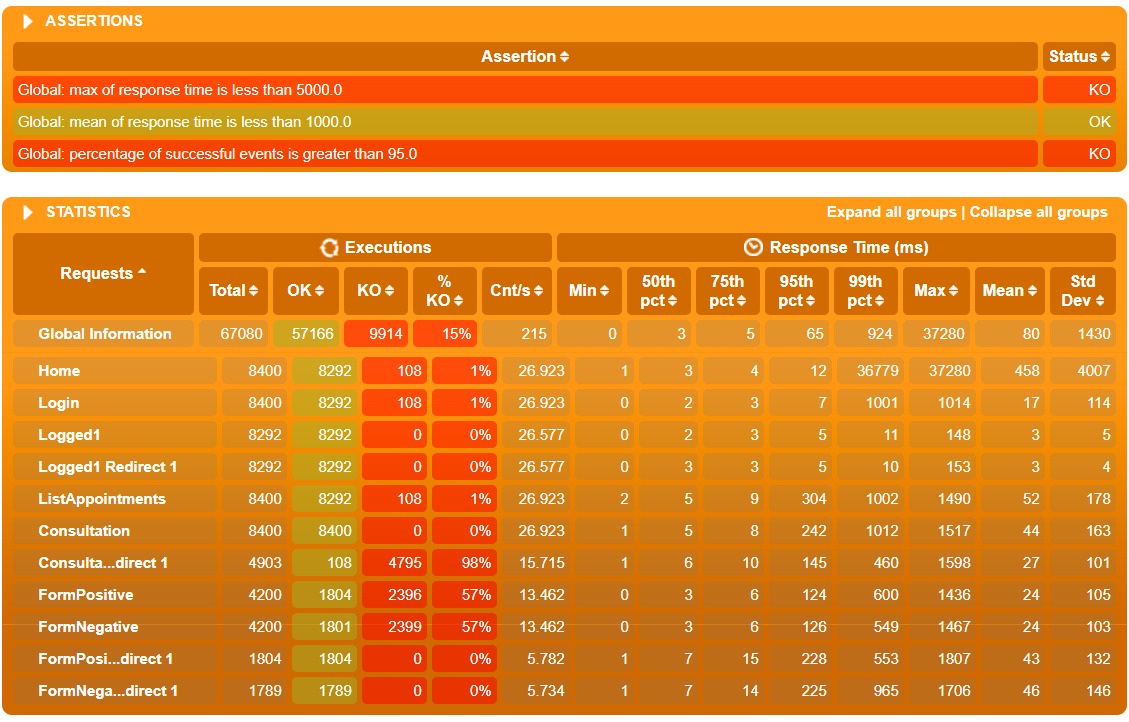
✘ El cliente Pepe Gotera se intenta autorecetar AAS-500-Mg-20-Comprimidos', pero el sistema no le deja hacerlo ya que no tiene permisos para ello.

**Entidades involucradas:** Medicina, Cita, Diagnóstico.

**Número mínimo de usuarios que no son soportados:** 240.000 en 10 segundos.

****

**Número máximo de usuarios simultáneos soportados:** 8.400 en 50 segundos.



* 1. . **HU-011: VER ENFERMEDADES ANTERIORES**

**Como** médico,

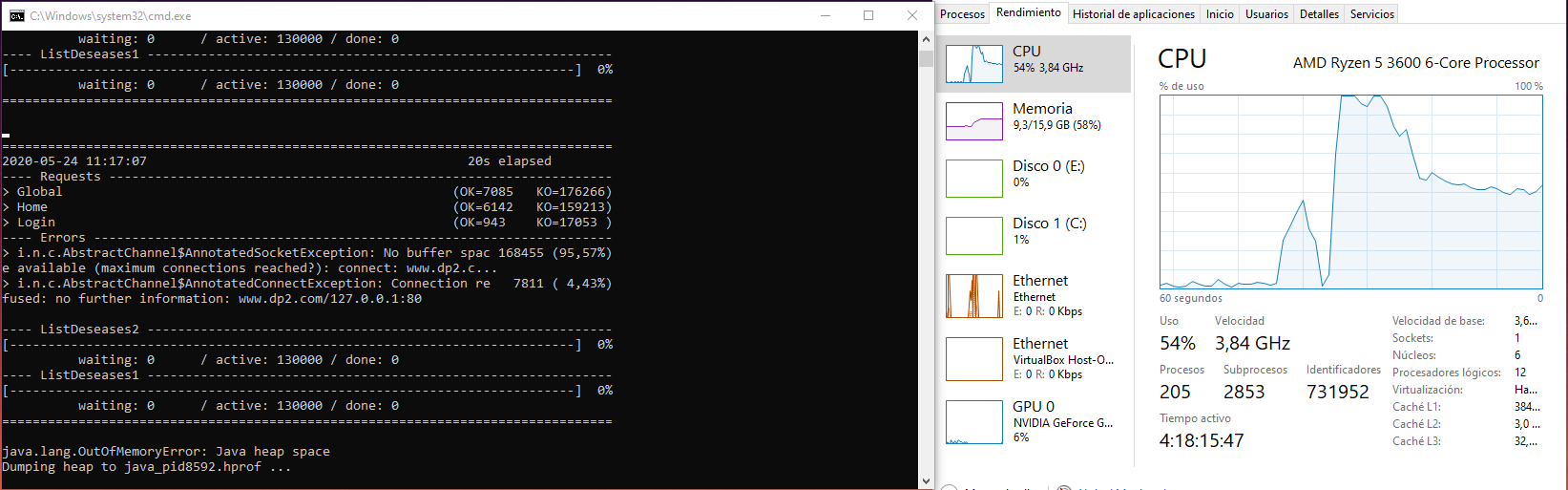
**Para** poder realizar un mejor diagnóstico y evitar que algunos medicamentos puedan empeorar la salud del paciente,

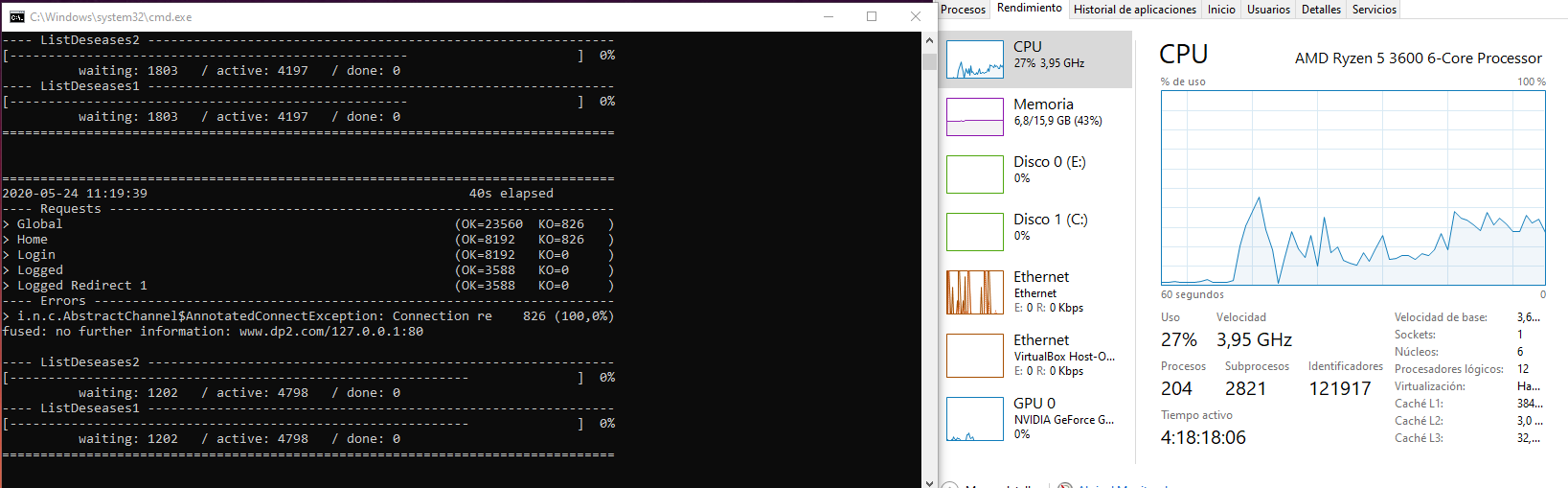
**Quiero** ver el histórico de enfermedades de cada paciente.

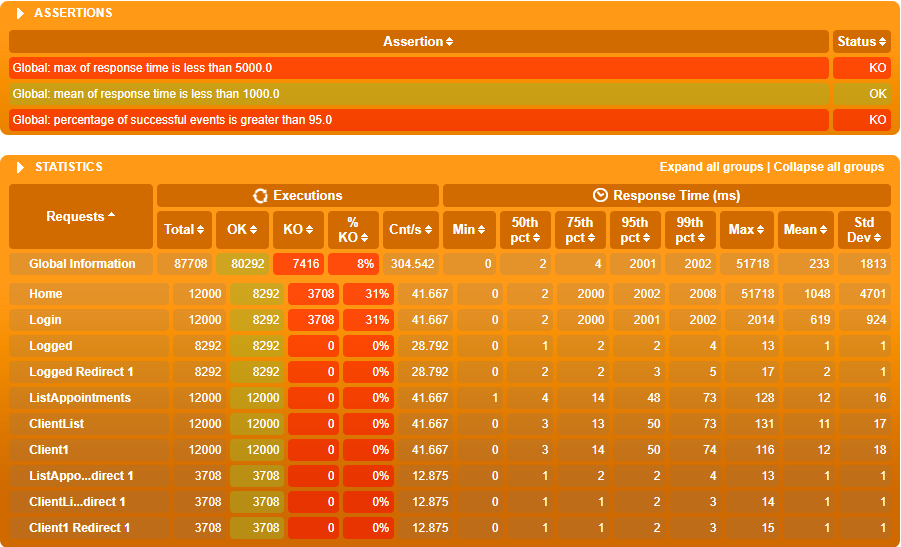
✔ Guillermo Díaz lista el historial de enfermedades que ha tenido Pepe Gotera.

✘ Guillermo Díaz intenta listar las enfermedades de Miguel Pérez, que no tiene historial, entonces el sistema le indica que no existe historial de enfermedades para ese paciente.

**Entidades involucradas:** Cliente, Enfermedad, Profesional, Diagnóstico.

**Número mínimo de usuarios que no son soportados:** 260.000 en 10 segundos.

**Número máximo de usuarios simultáneos soportados:** 9.596 en 50 segundos.



* 1. . **HU-012: VER TRATAMIENTOS ANTERIORES**

**Como** médico,

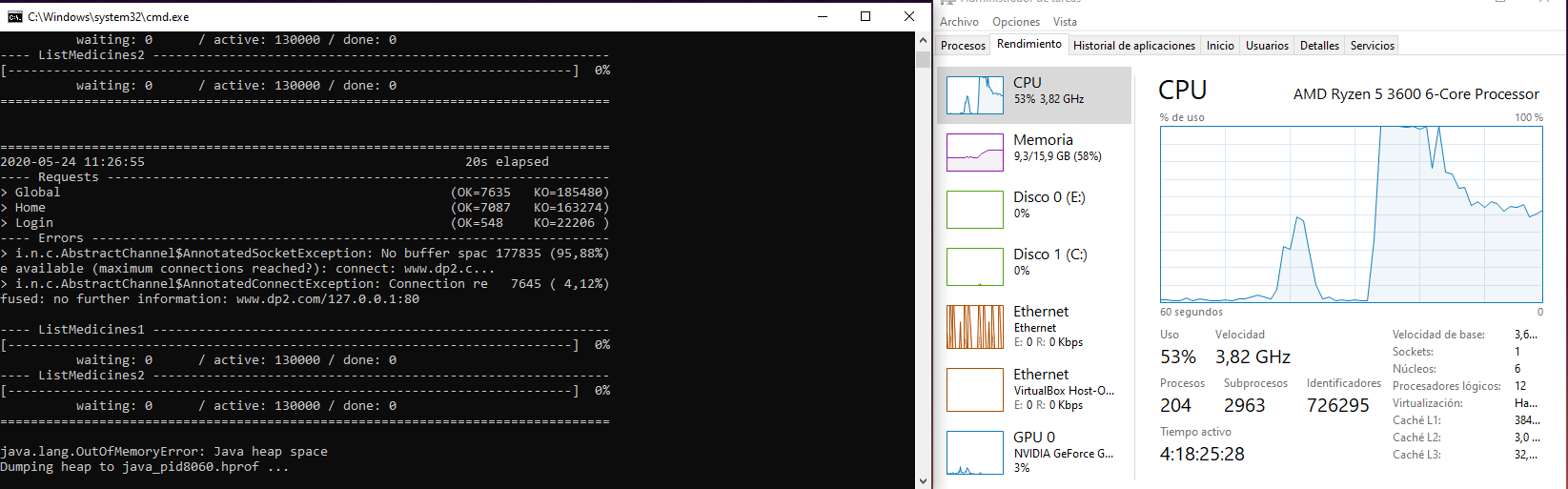
**Para** poder realizar un mejor diagnóstico y evitar poner en peligro la salud del paciente,

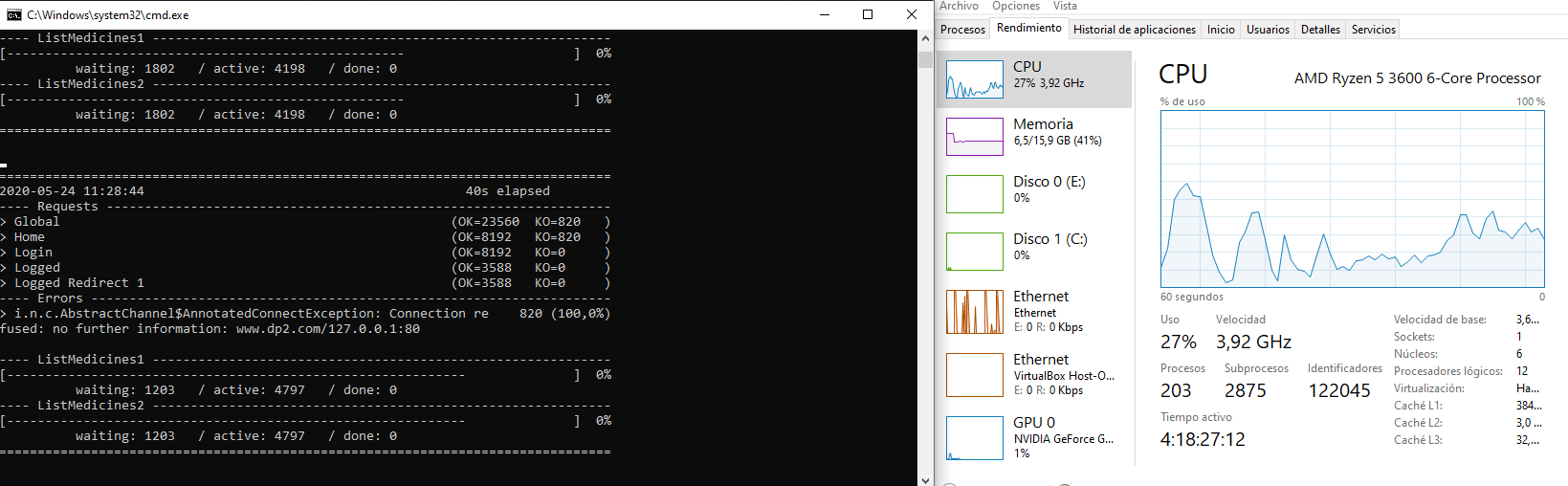
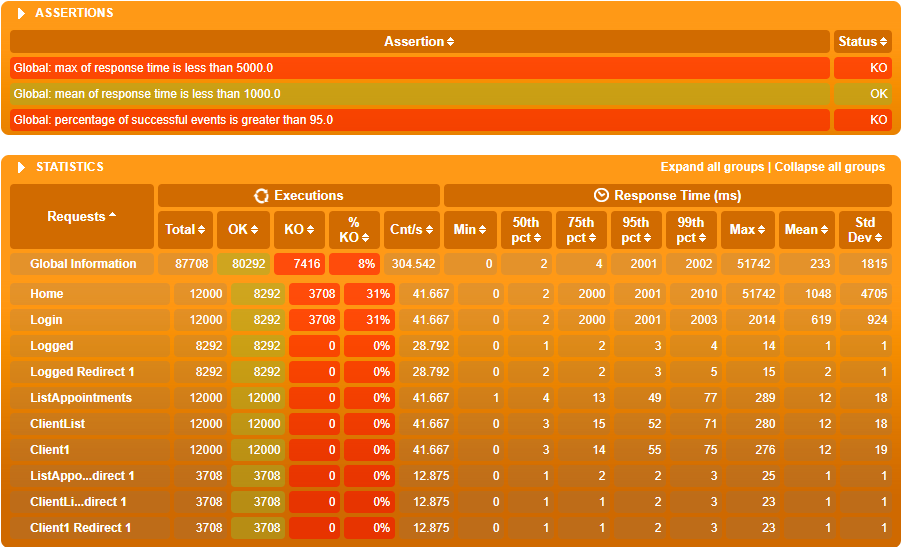
**Quiero** ver el histórico de tratamientos de cada paciente.

✔ Guillermo Díaz lista el historial de fármacos y tratamientos que ha tenido Pepe Gotera.

✘ Guillermo Díaz intenta listar los fármacos y tratamientos de Miguel Pérez, que no tiene historial, entonces el sistema le indica que no existe historial de fármacos y tratamientos para ese paciente.

**Entidades involucradas:** Cliente, Diagnóstico, Medicina.

**Número mínimo de usuarios que no son soportados:** 260.000 en 10 segundos.

**Número máximo de usuarios simultáneos soportados:** 9.594 en 50 segundos.

* 1. . **HU-013: Busqueda profesionales**

**Como** cliente,

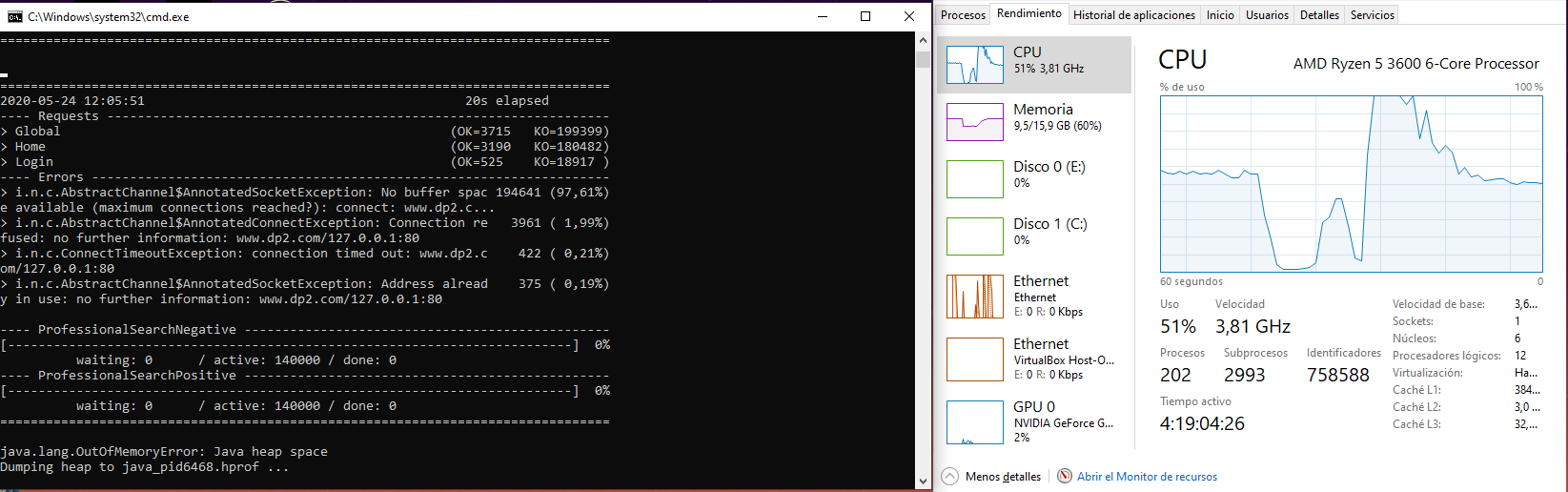
**Para** conocer los datos y especialidades de los distintos profesionales,

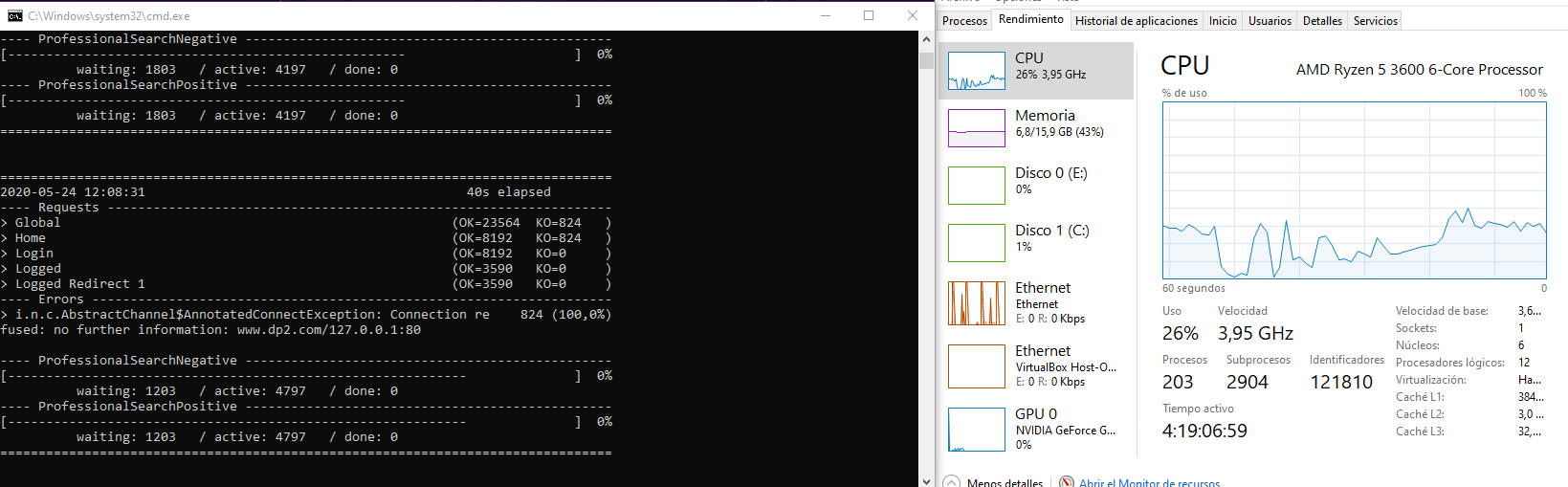
**Quiero** tener una lista con los profesionales filtrados por centro y especialidad.

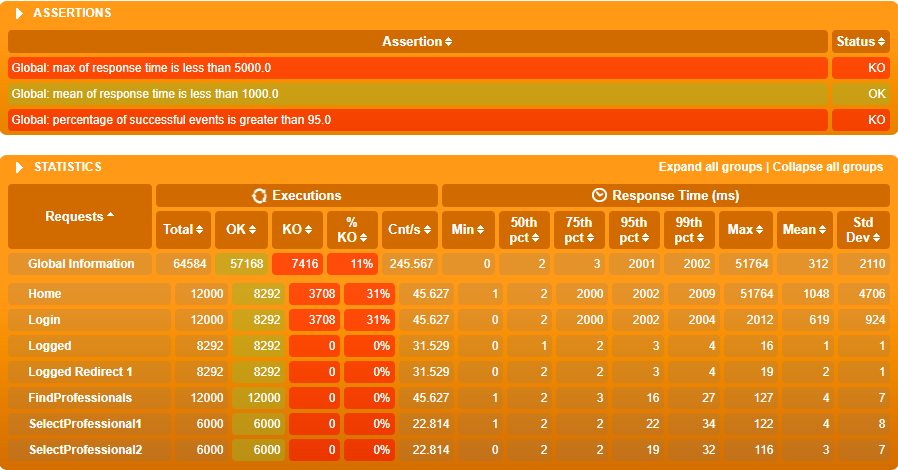
✔ Pepe Gotera busca profesionales que trabajen en Sevilla y estén especializados en dermatología.

✘ Elena Nito al intentar buscar por profesionales, pero se olvida de elegir el campo de especialidad, saltando un error.

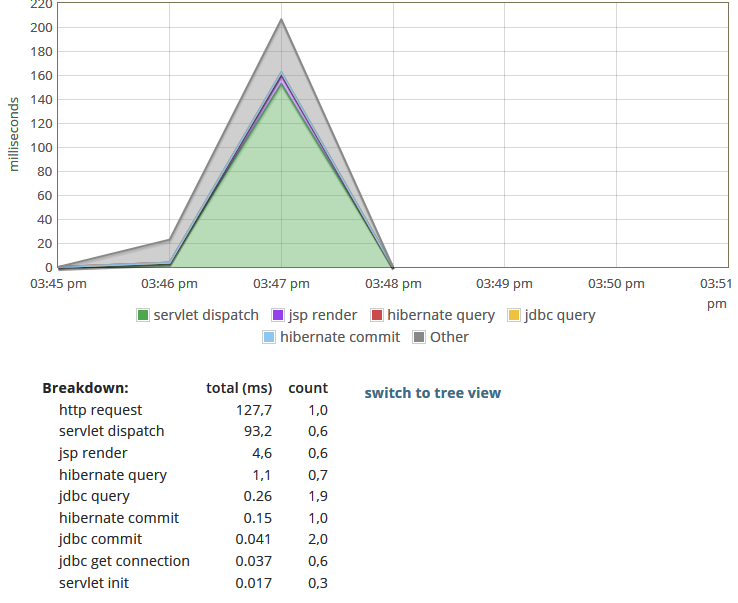
**Entidades involucradas:** Profesional, Centro, Especialidad.

**Número mínimo de usuarios que no son soportados:** 280.000 en 10 segundos.

**Número máximo de usuarios simultáneos soportados:** 9.594 en 50 segundos.



Analizando el test de rendimiento de la búsqueda de profesionales, a simple vista observamos que la mayor parte del tiempo el test de rendimiento de búsqueda de profesionales lo usa en el servlet dispatch, como podemos observar en la imagen de más abajo.



El hecho de que el servlet use tanto tiempo es normal, ya que en la acción de buscar profesionales obtenemos una vista en la que podemos introducir la especialidad y centro que deseemos y una tabla con los profesionales que cumplan los requisitos. Esto se refleja también en el jsp render.

Analizando las transacciones, se obtienen los datos descritos más abajo. Los cuales tienen sentido, ya que la mayor parte del tiempo nos encontramos en la vista de profesionales.



Como conclusión parece que la historia de usuario está creada correctamente.

Tras analizar el caso negativo vemos que se obtienen valores parecidos ya que la única diferencia respecto al positivo es que en vez de una tabla obtenemos mensajes de error.

* 1. . **HU-014: FACTURAS**

**Como** administrador del sistema,

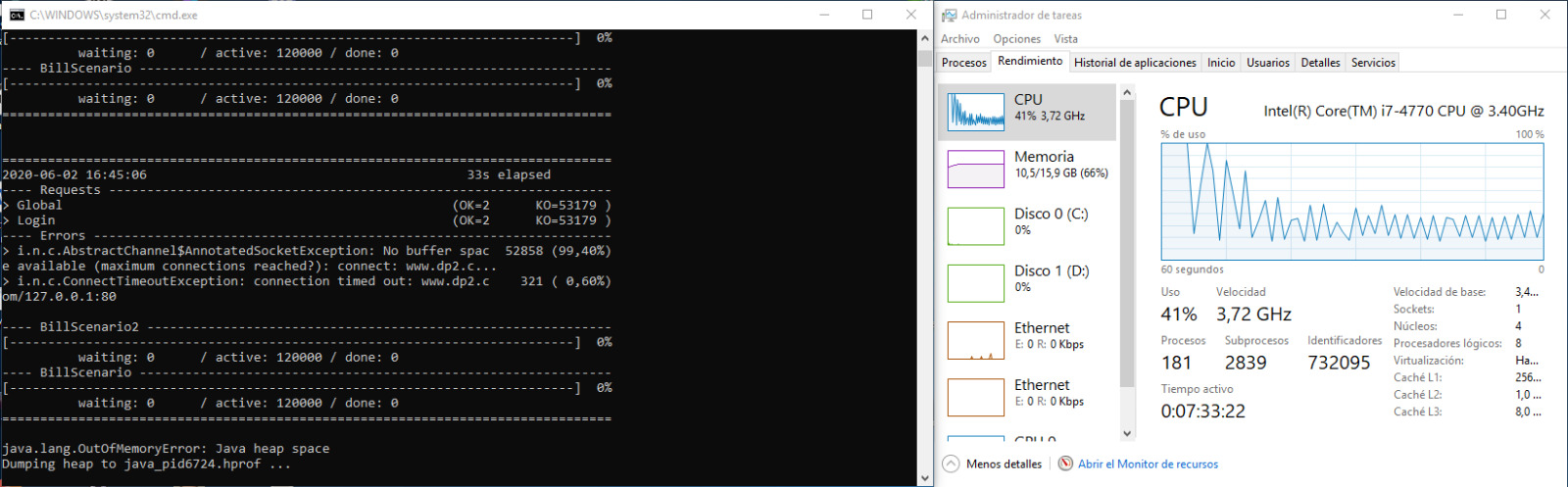
**Para** cobrar la factura de manera más cómoda y rápida,

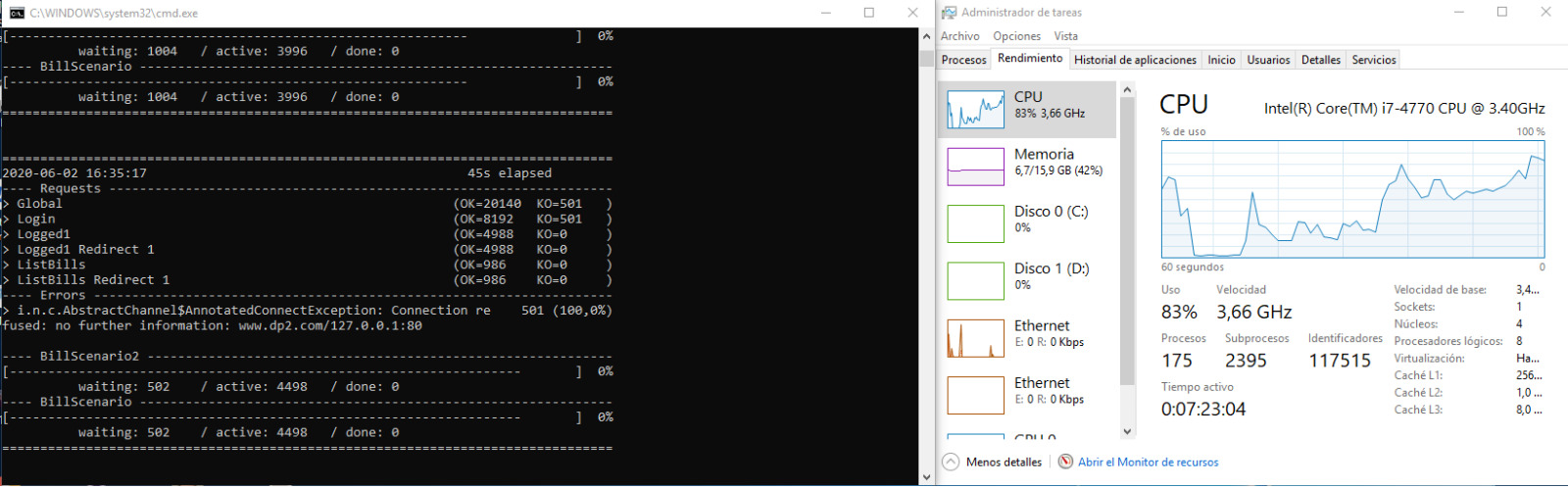
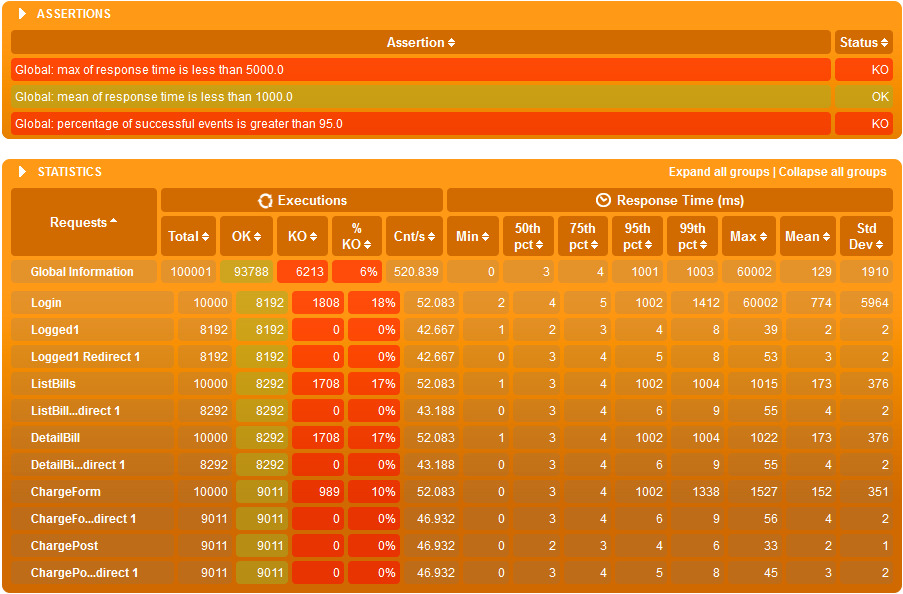
**Quiero** que el sistema calcule el pago final y emita una factura.

✔ El administrador introduce el precio de 50 euros a cobrar y el método de pago por donde se va a cobrar y el sistema le calcula lo que falta por pagar.

✘ El administrador introduce ceros en los datos del precio de la factura y el sistema le muestra el mensaje de error de que los datos introducidos no pueden ser 0.

**Entidades involucradas:** Factura, Cita, Cliente, Método de pago, Transacción.

**Número mínimo de usuarios que no son soportados:** 240.000 en 10 segundos.

**Número máximo de usuarios simultáneos soportados:** 8.996 en 50 segundos.

* 1. . **HU-015: PAGOS CON TARJETA**

**Como** cliente,

**Para** poder pagar con mayor comodidad,

**Quiero** tener la posibilidad de realizar los pagos mediante tarjeta de crédito.

✔ Pepe Gotera introduce una tarjeta de crédito VISA como método de pago, introduciendo los datos de la misma.

✘ Pepe Gotera intenta realizar el pago mediante tarjeta de crédito introduciendo datos inexistentes.

**Entidades involucradas:** Método de pago, Cliente.

**No se puede hacer el análisis de rendimiento de dicha historia de usuario porque involucra a la API externa.**

* 1. . **HU-016: Diagnóstico MÉDICo**

**Como** profesional,

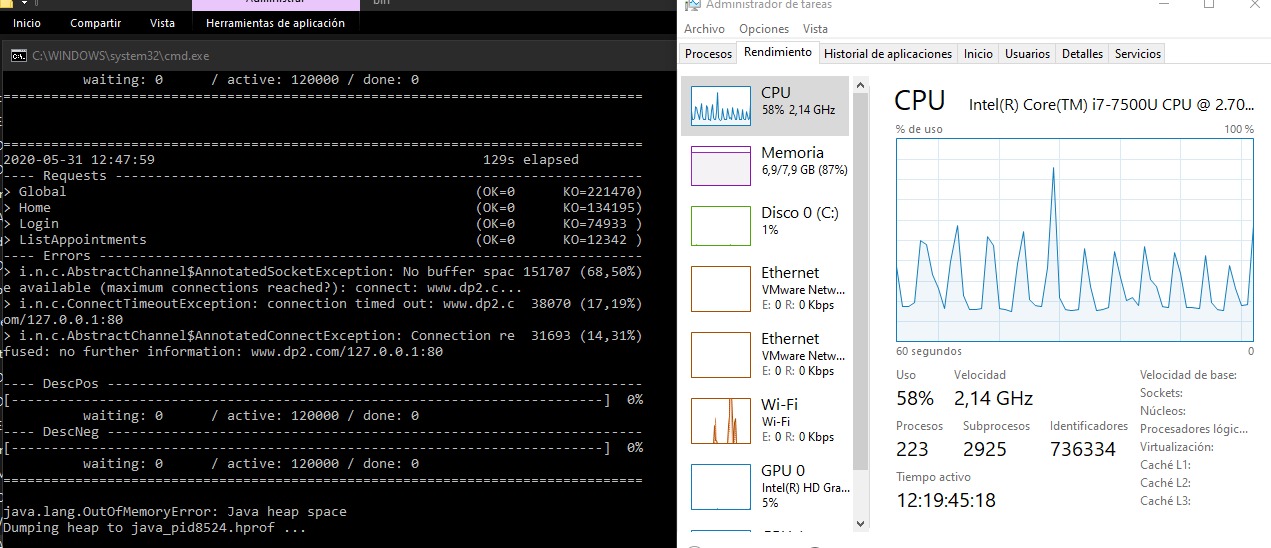
**Para** realizar un diagnóstico de la cita seleccionada,

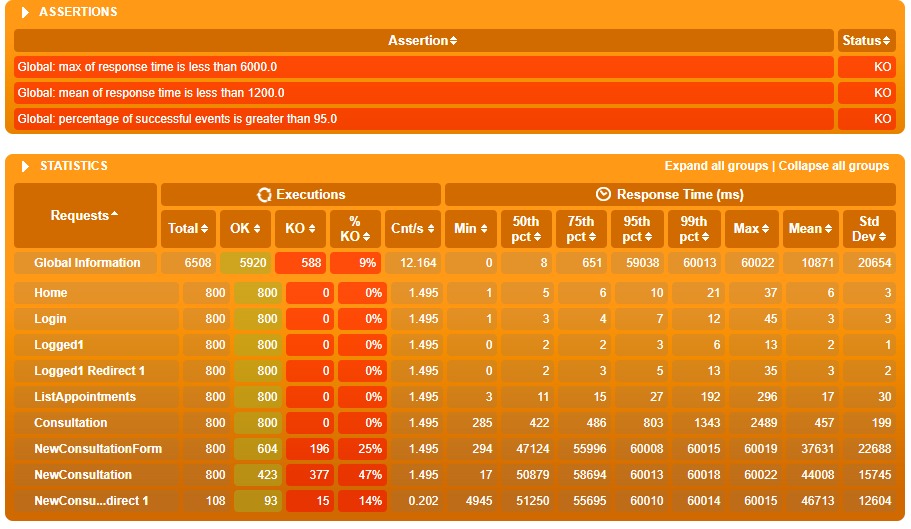
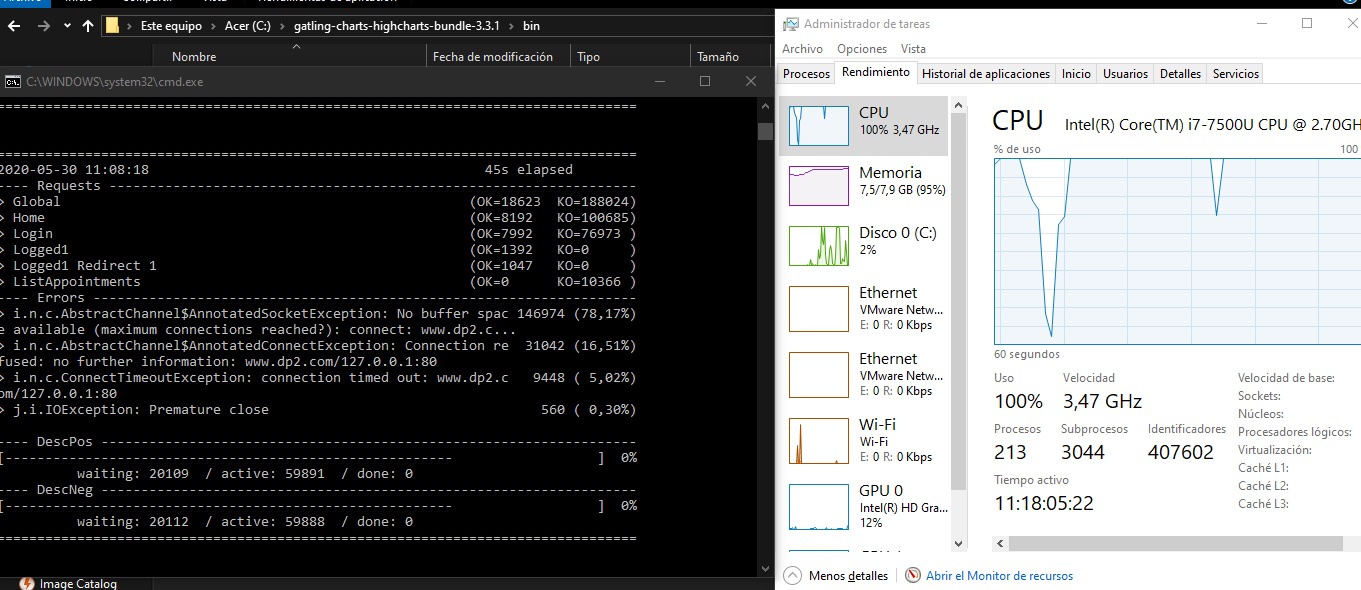
**Quiero** poder guardar un diagnóstico para la cita correspondiente, seleccionando el tratamiento.

✔ Guillermo Díaz selecciona la/s enfermedad/es, medicina/s y una descripción para el cliente Pepe Gotera y finaliza la consulta.

✘ Julio Maldonado quiere realizar un diagnóstico para una cita de Pepe Gotera, pero se le olvida rellenar la descripción.

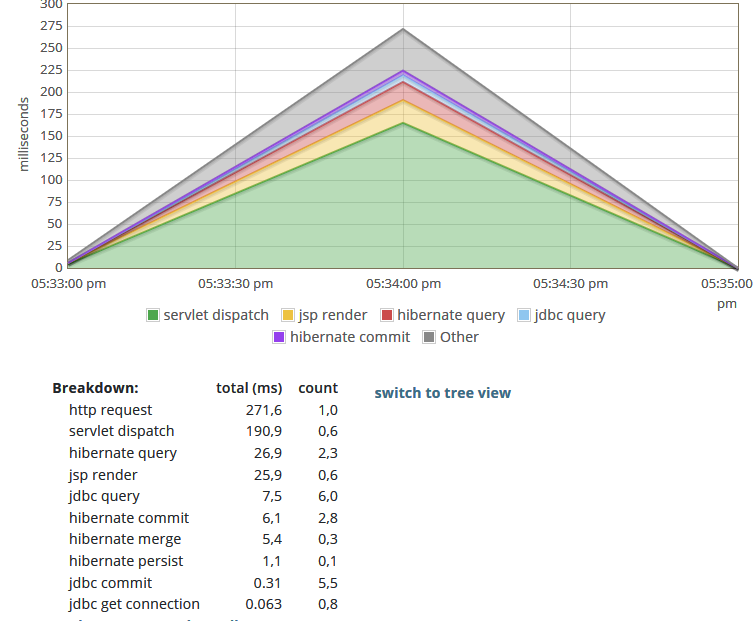
**Entidades involucradas:** Diagnóstico, Cita.

**Número mínimo de usuarios que no son soportados:** 240.000 en 5 segundos.

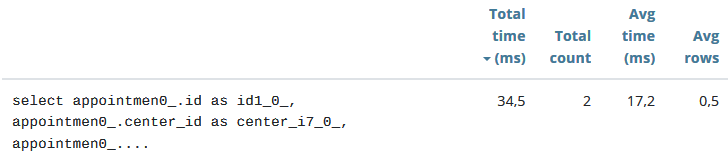
******Número máximo de usuarios simultáneos soportados:** 770 en 60 segundos.

Esta historia trata sobre la creación de un diagnóstico por parte de un profesional.

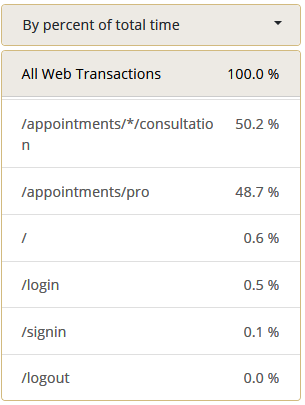
Si observamos la gráfica con los tiempos obtenidos en la captura vemos que los tiempos parecen correctos, ya que es normal que el servlet dispatch sea tan alto debido a que la vista de consulta tiene bastantes elementos.



Resulta curioso que el valor de hibernate query sea tan alto. Tras acceder a la pestaña query, vemos que la razón por la que se gasta tanto tiempo en esta propiedad es la query en la que se obtiene la cita.



Analizando los porcentajes de las transacciones tiene bastante sentido que los de consulta tengan el valor tan alto, ya que la mayoría de las operaciones tienen lugar en esas vistas.



Respecto al caso negativo la gráfica es bastante parecida, ya que en caso de que haya error se vuelve a la vista de consulta.

* 1. . **HU-017: PANEL DE CONTROL**

**Como** administrador del sistema,

**Para** un mejor control del centro de salud.

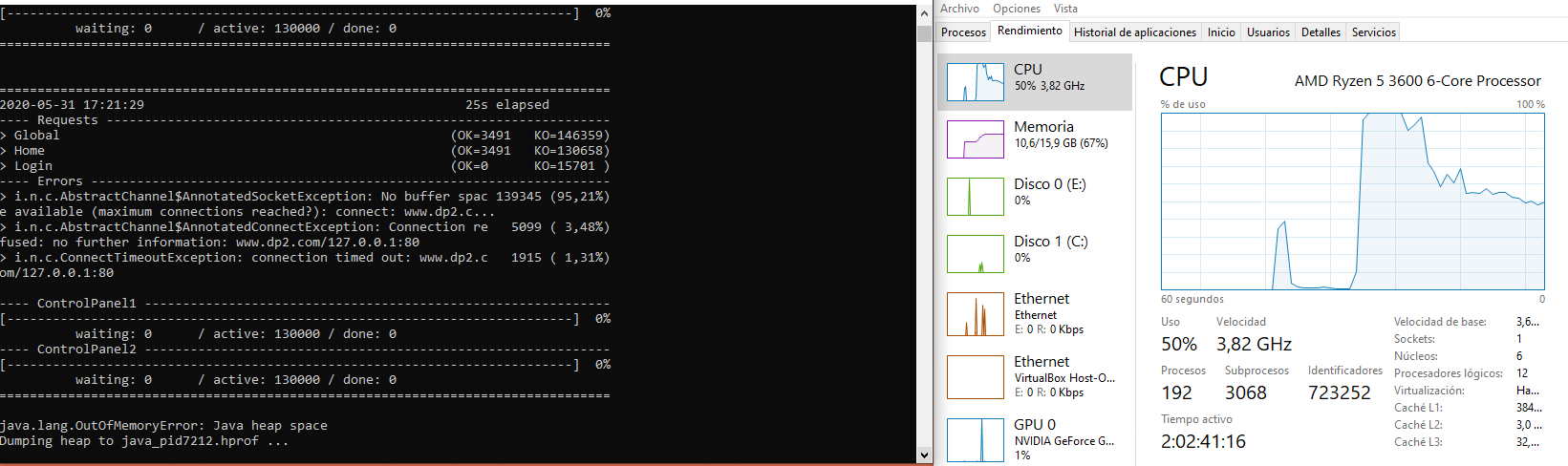
**Quiero** tener un panel de control donde pueda navegar por los distintos clientes, profesionales y citas y ver sus detalles, pudiendo también crearlos, actualizarlos o eliminarlos. Además, también me gustaría tener una pestaña donde pueda ver estadísticas, tales como las citas agrupadas por estado, el precio acumulado de todas las citas y el precio acumulado por día.

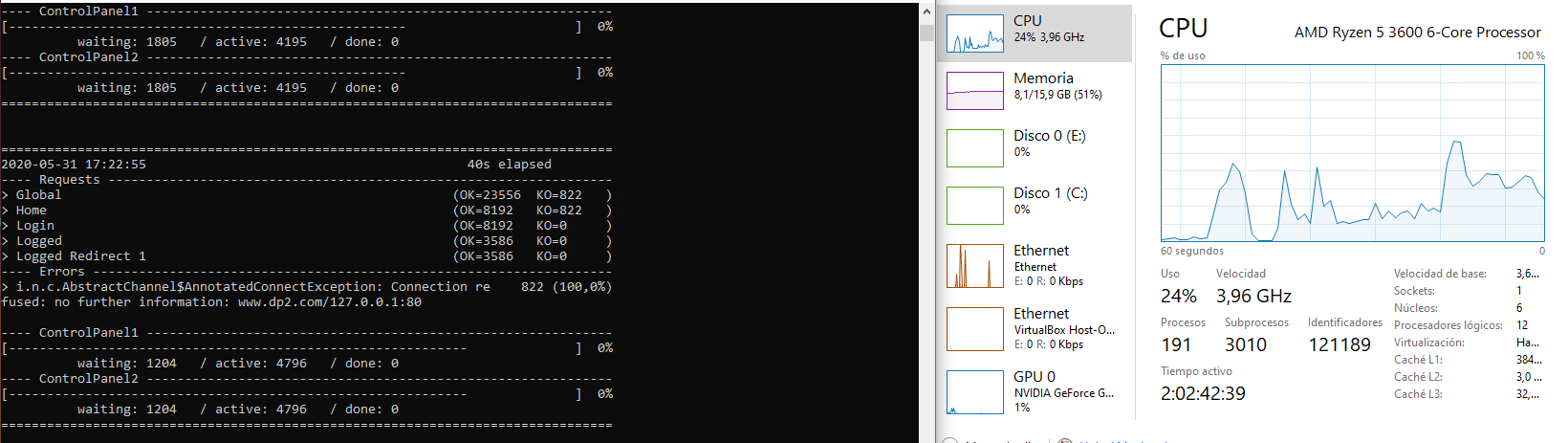
✔ El administrador crea un profesional nuevo distinto.

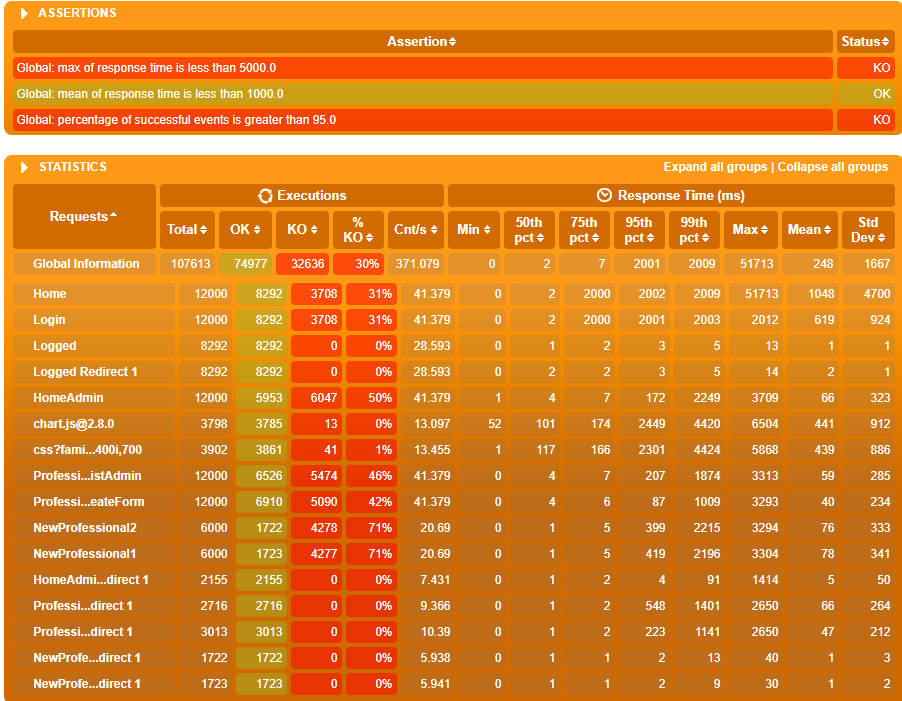
✘ El administrador intenta crear un profesional con el formato de email incorrecto y el sistema le muestra el mensaje de que debe elegir un formato correcto.

**Entidades involucradas:** Cliente, Profesional, Cita, Factura, Método de pago.

**Número mínimo de usuarios que no son soportados:** 260.000 en 10 segundos.



**Número máximo de usuarios simultáneos soportados:** 9.592 en 50 segundos.



* 1. . **HU-018: ELECCIÓN DE CONTRASEÑA**

**Como** administrador,

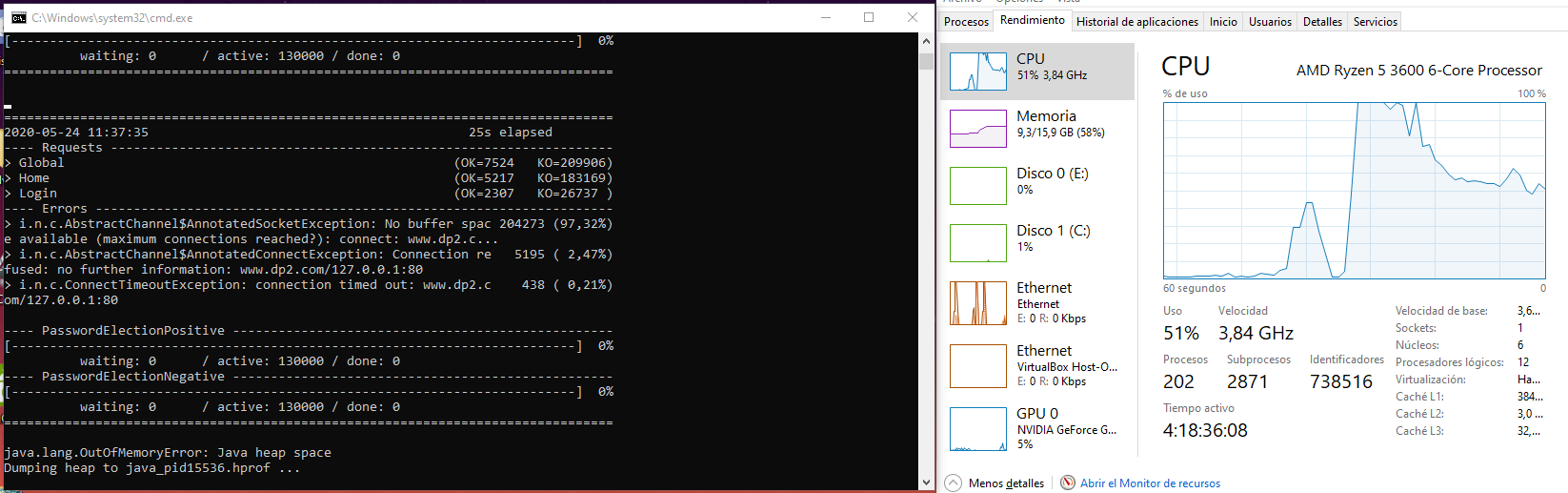
**Para** mejorar la seguridad de los clientes,

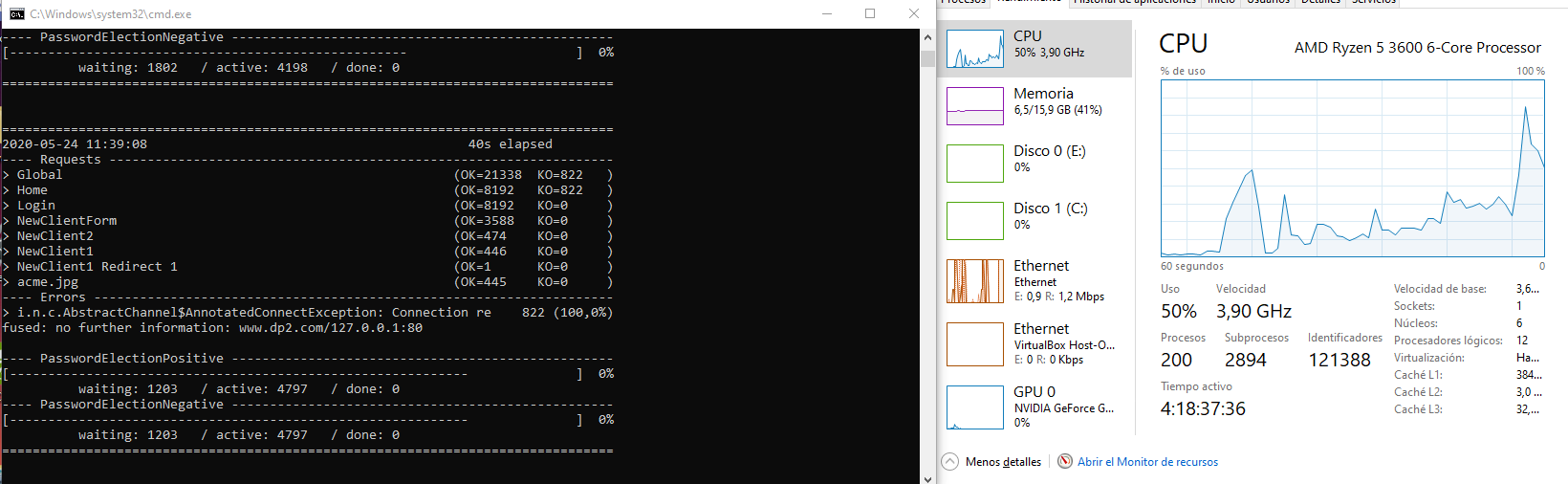
**Quiero** que las contraseñas de los clientes que se vayan a registrar tengan entre 6 y 15 caracteres.

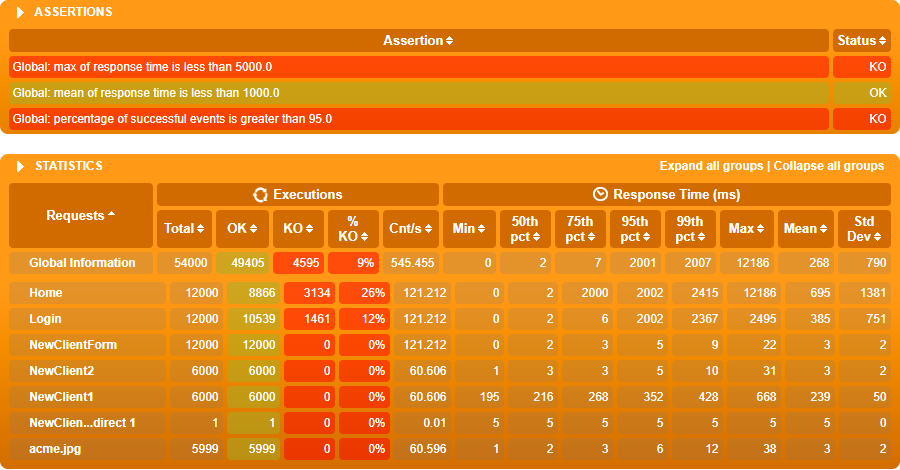
✔ Manu Reina escribe una contraseña de 8 caracteres y rellena todos los campos correctamente.

✘ Rubén Doblas intenta registrarse con una contraseña de 4 caracteres, pero se le muestra un mensaje de error.

**Entidades involucradas:** Cliente.

**Número mínimo de usuarios que no son soportados:** 260.000 en 10 segundos.

**Número máximo de usuarios simultáneos soportados:** 9.594 en 50 segundos.



* 1. . **HU-019: LISTAR CITAS CLIENTE**

**Como** cliente,

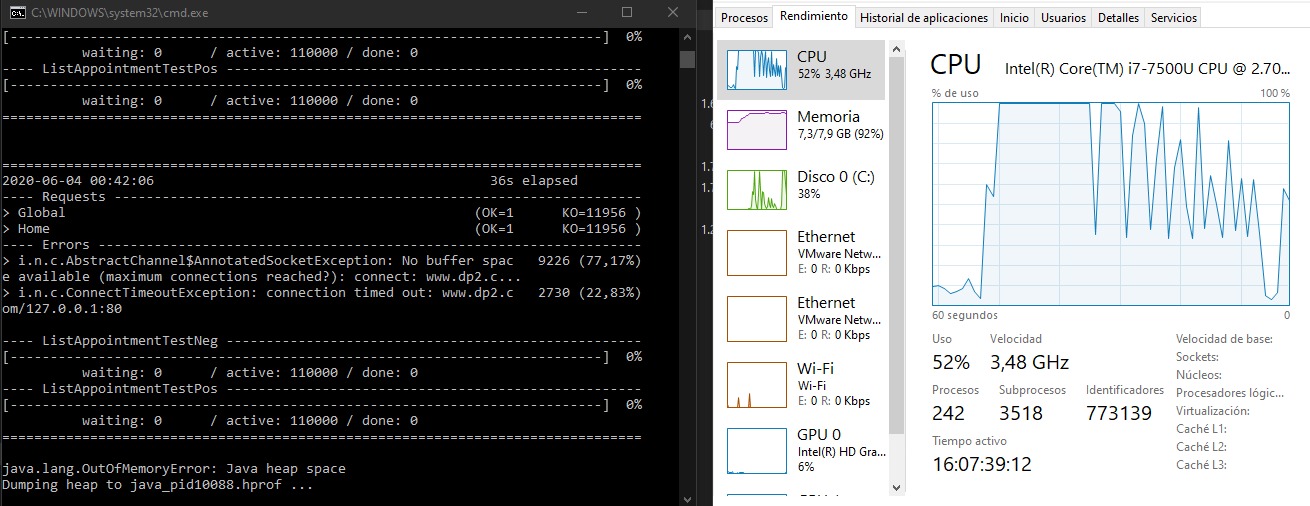
**Para** conocer las citas anteriores y futuras,

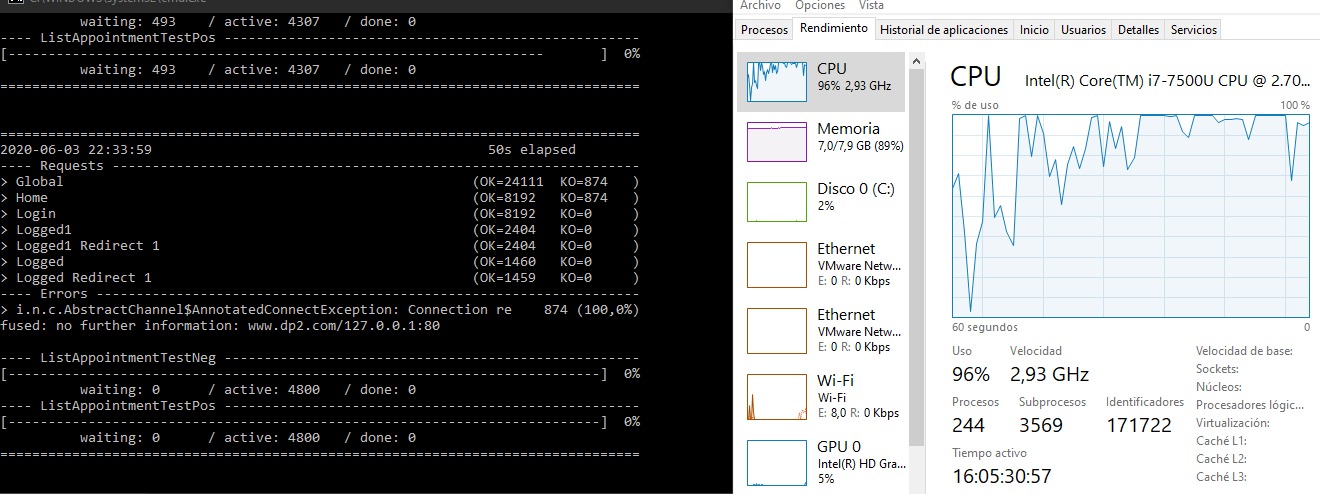
**Quiero** poder acceder al listado de todas mis citas registradas.

✔ Pepe Gotera visualiza el listado con todas las citas que ha registrado.

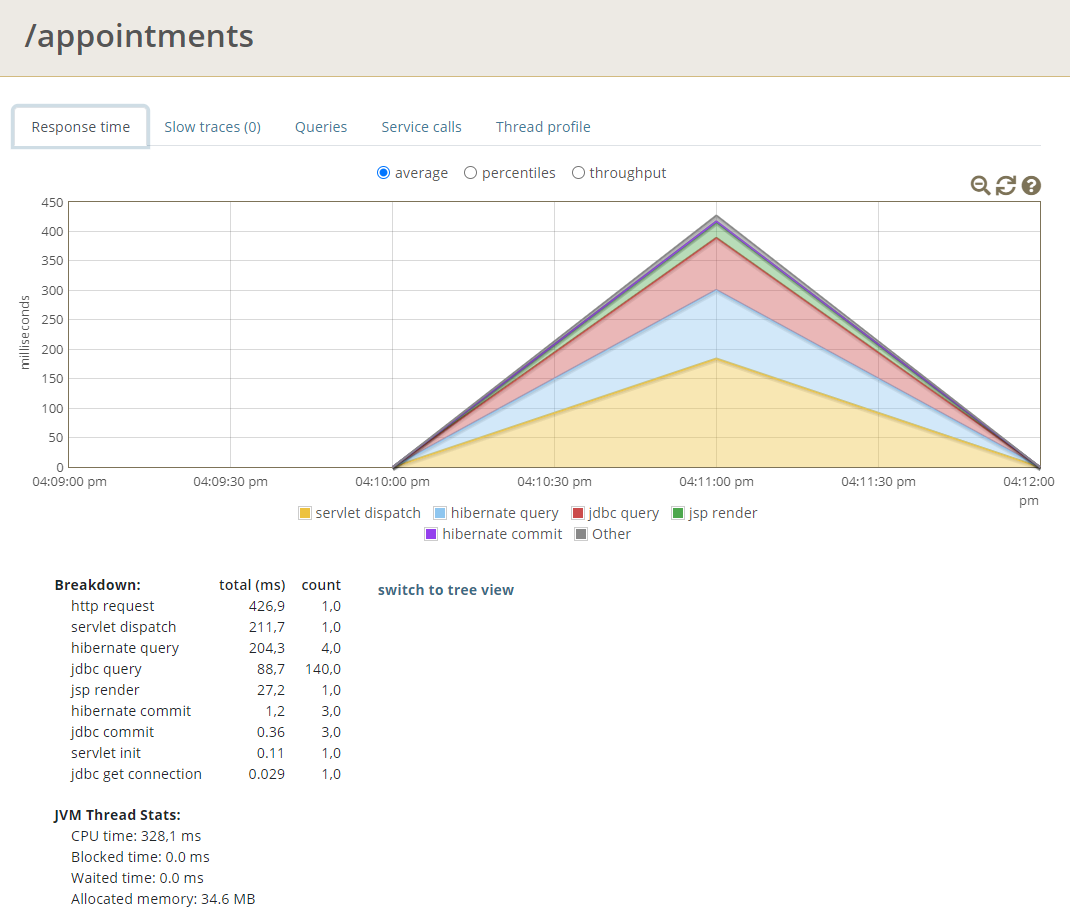
✘ Miguel Pérez intenta acceder al listado de sus citas, pero al no haber registrado ninguna, el sistema le indica que no hay citas.

**Entidades involucradas:** Cliente, Cita.

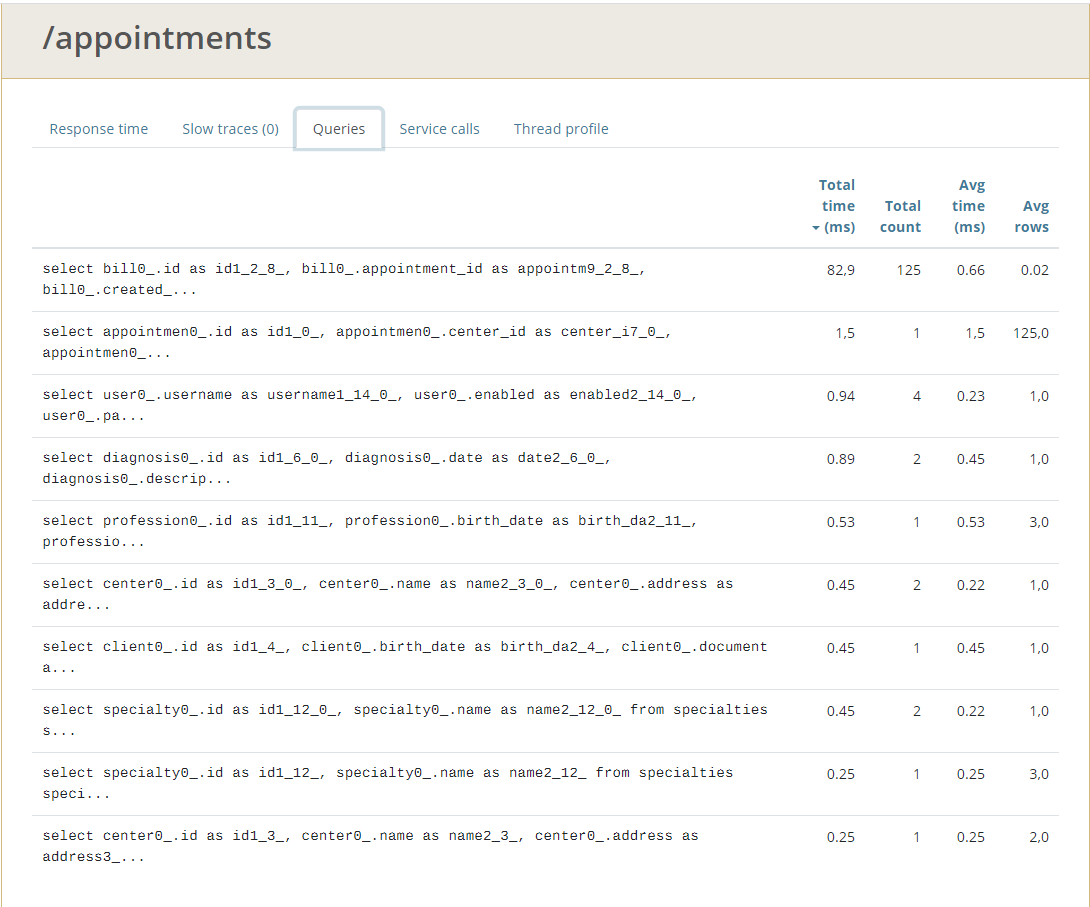
**Número mínimo de usuarios que no son soportados:** 220.000 en 10 segundos.

**Número máximo de usuarios simultáneos soportados:** 9.600 en 50 segundos.

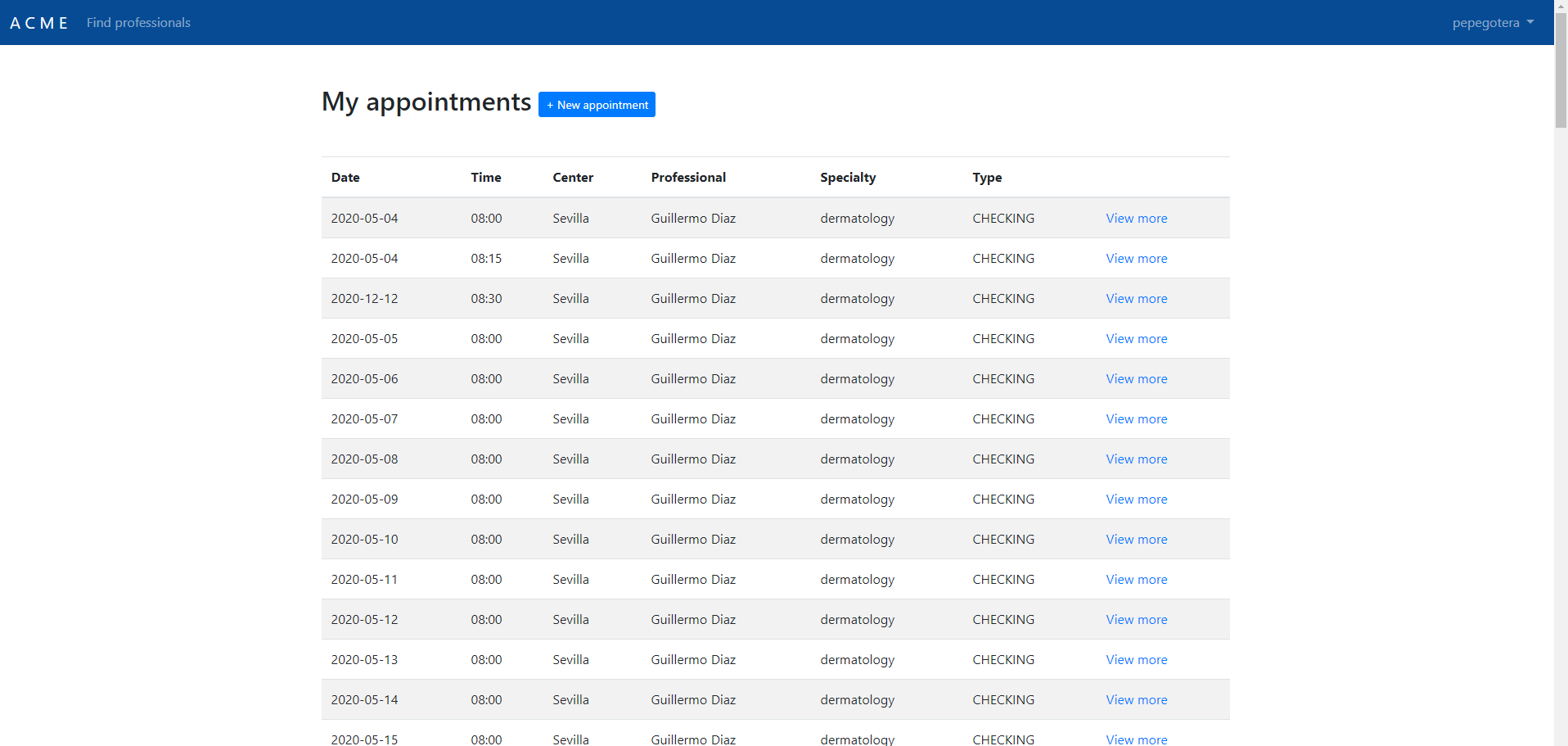
En cuanto al profiling de esta historia de ususario, se han sacado los siguientes resultados:



Como se puede observar, se tarda mucho tiempo en procesar las jdbc querys, y esto se debe por la siguiente razón:



Se tarda 82,9 ms en hacer un total de 125 querys, algo innecesario y que empeora el rendimiento considerablemente. Para que se vea mejor de lo que hablo, aquí está la vista en la aplicación, una muestra de la lista de citas de un cliente:



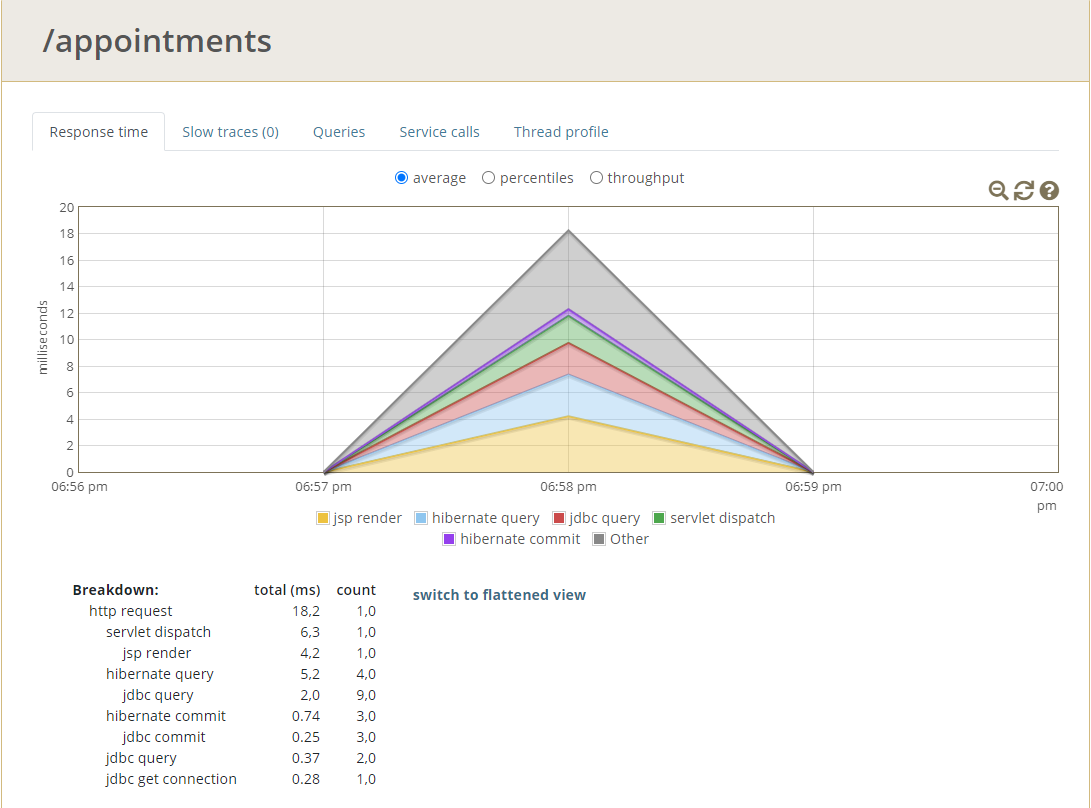
Las conclusiones que se pueden sacar de esto son obvias: hace demasiadas llamadas innecesarias. También, no tiene mucho sentido que en el listado de las citas salgan todos los atributos de esta ya que al darle al botón de “view more” vuelven a salir, a parte de otros atributos involucrados.

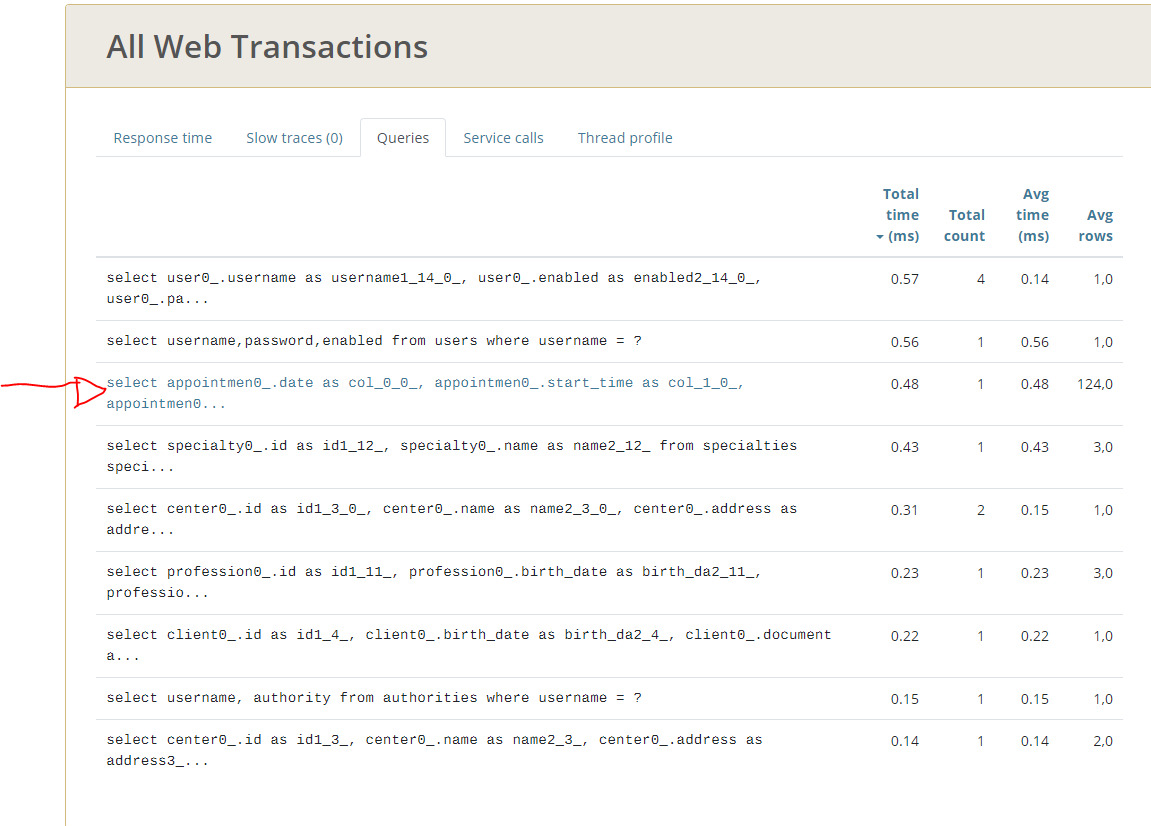
Para mejorar el rendimiento de esto, he hecho una refactorización de projection, para coger solo los atributos mínimos que me interesan y mostrarlos en el listado. He creado una interfaz ListAppointmentClient con los get de estos atributos y he modificado la query de encontrar citas por cliente de la siguiente forma:

@Query("SELECT a.date AS date, a.startTime AS startTime, a.id AS id "

+ "FROM Appointment a WHERE a.client.id = :id ORDER BY a.id ORDER BY a.date")

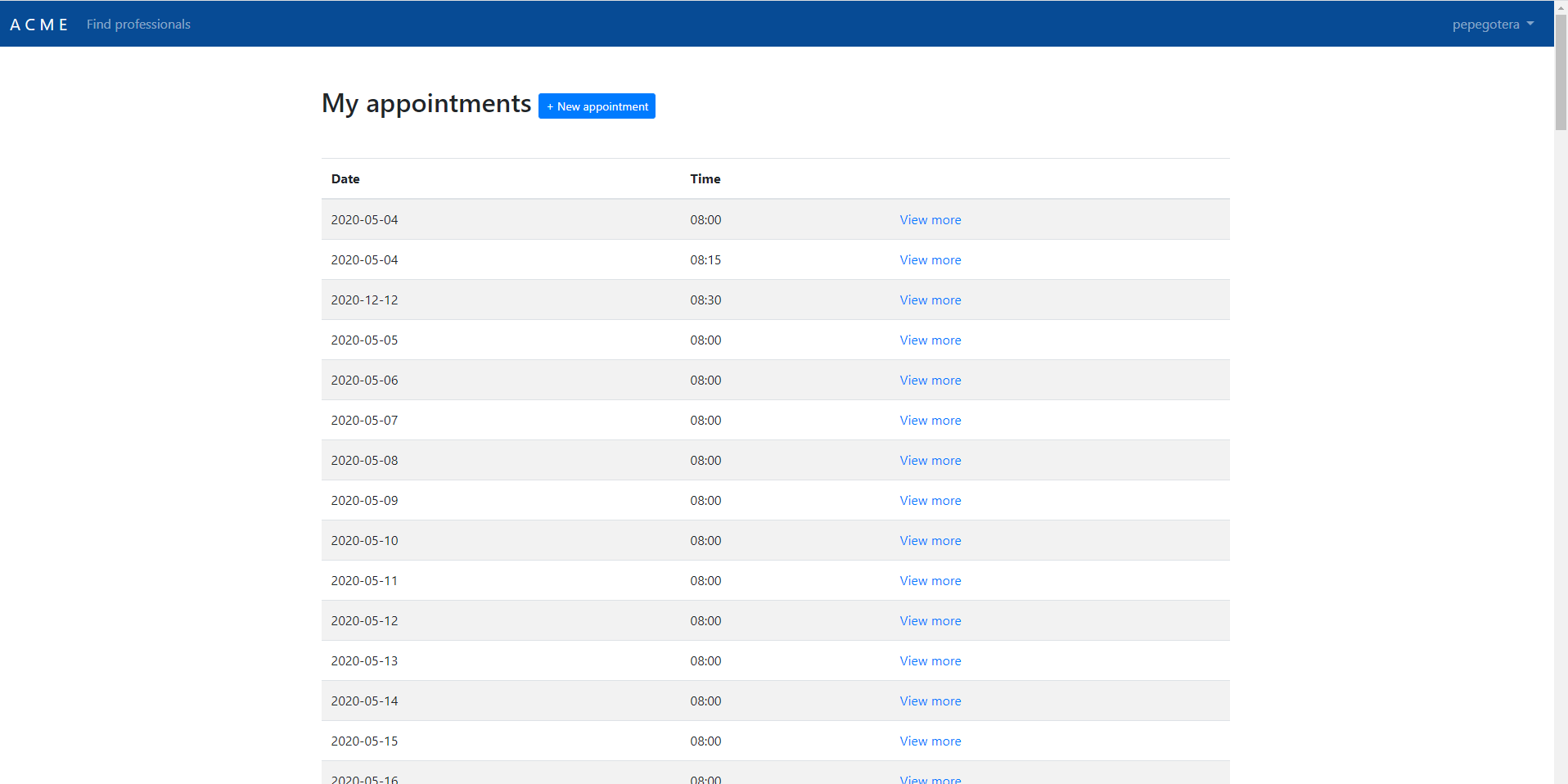
De esta forma, ya no se procesan tantas querys, por lo que el rendimiento mejora considerablemente, tal y como se muestran en las siguientes capturas al volver a hacer el profiling:





Como se puede observar, el tiempo total se ha reducido de 82,9 ms a 0,48, y la cuenta total del procesamiento de esa query se ha reducido de 125 a 1.

Así quedaría la vista con la refactorización realizada:



* 1. . **HU-020: MÉTODOS DE PAGO Repetidos**

**Como** administrador,

**Para** evitar problemas a la hora de pagar,

**Quiero** que el sistema impida que un cliente pueda añadir dos veces el mismo método de pago.

✔ Raúl Pérez incluye sus tarjetas de crédito introduciendo los datos correspondientes a estas.

✘ Raúl Pérez intenta añadir de nuevo una tarjeta ya añadida anteriormente.

**Entidades involucradas:** Cliente, Método de Pago.

**No se puede hacer el análisis de rendimiento de dicha historia de usuario porque involucra a la API externa.**

* 1. . **HU-021: precio consulta**

**Como** profesional,

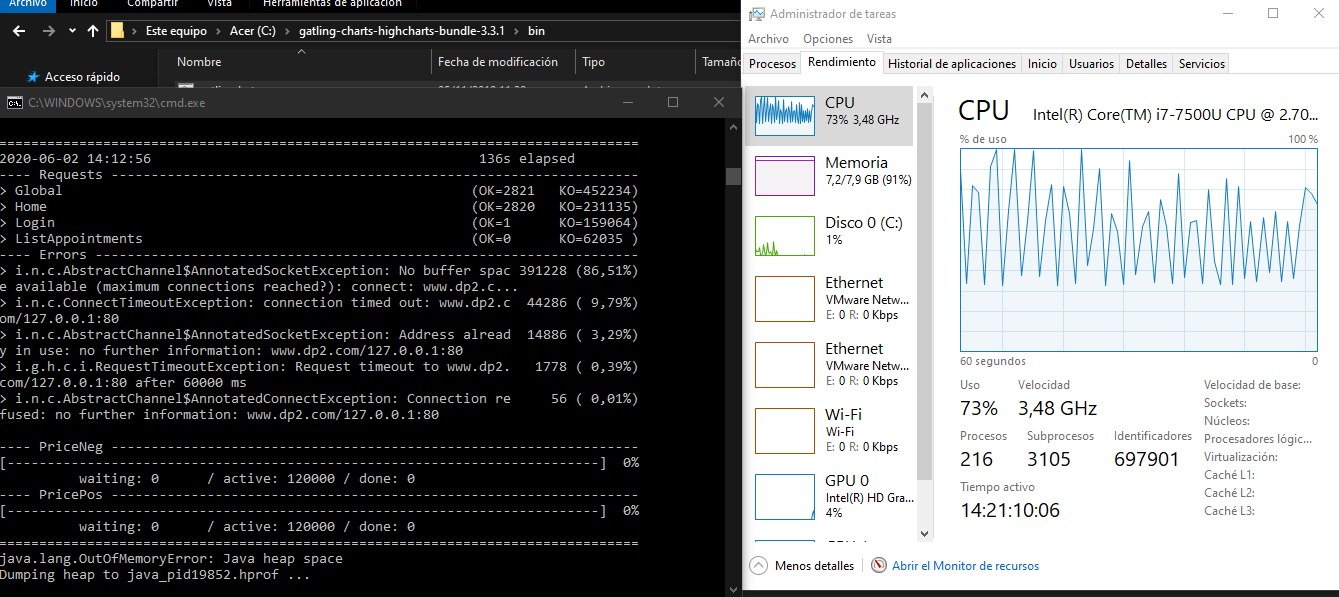
**Para** optimizar el proceso de pago,

**Quiero** poder fijar el precio de los servicios ofrecidos por mi trabajo.

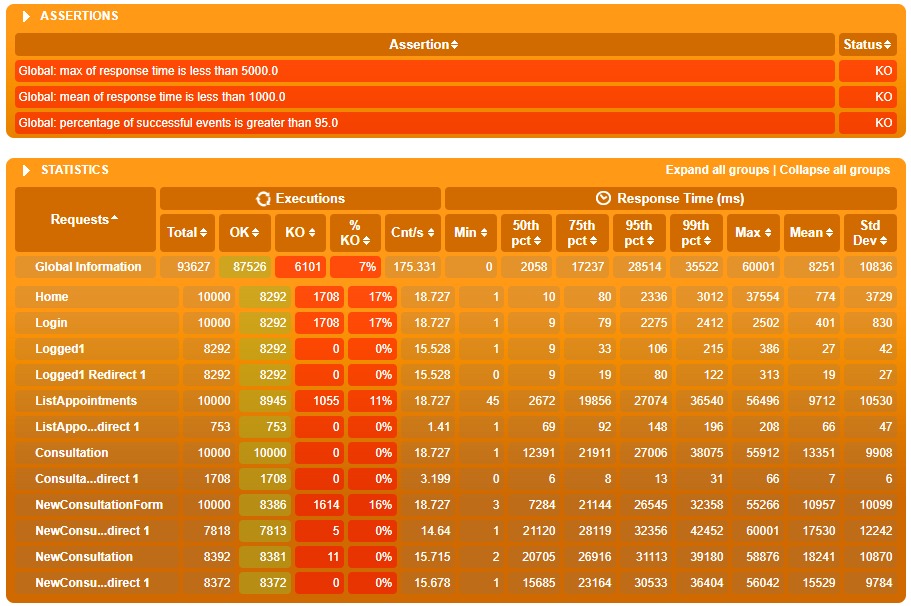
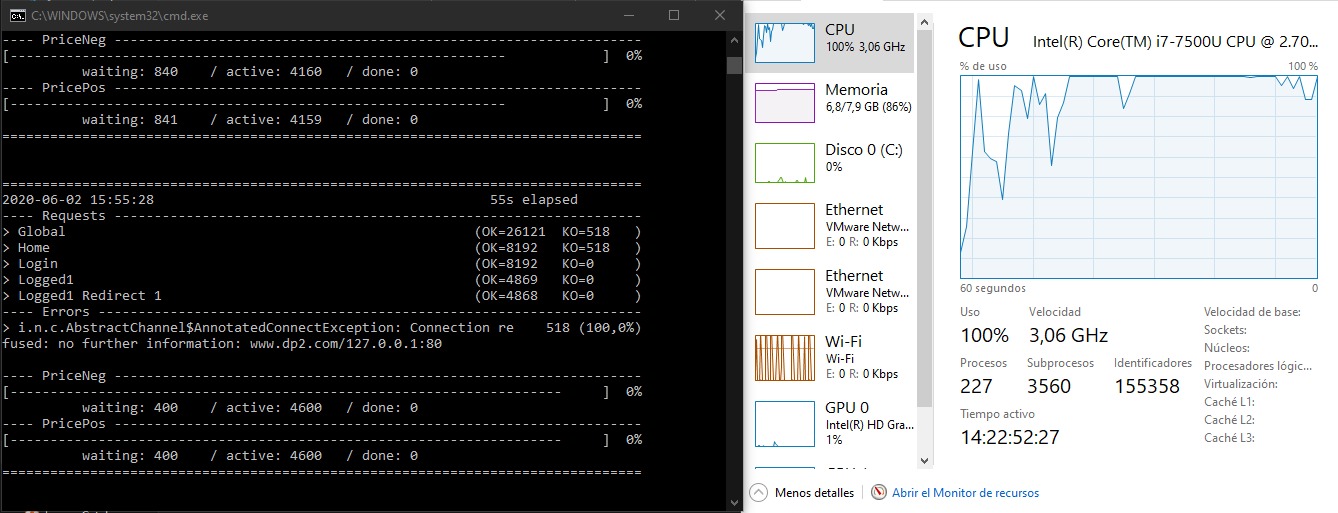
✔ Guillermo Díaz fija un precio de 10 euros por la consulta.

✘ Julio Maldonado fija un precio de la consulta de -5 euros de forma errónea, por lo que el sistema le impide finalizar la creación de la misma.

**Entidades involucradas:** Cliente, Cita, Factura.

**Número mínimo de usuarios que no son soportados:** 240.000 en 10 segundos.

**Número máximo de usuarios simultáneos soportados:** 9.200 en 50 segundos.

****

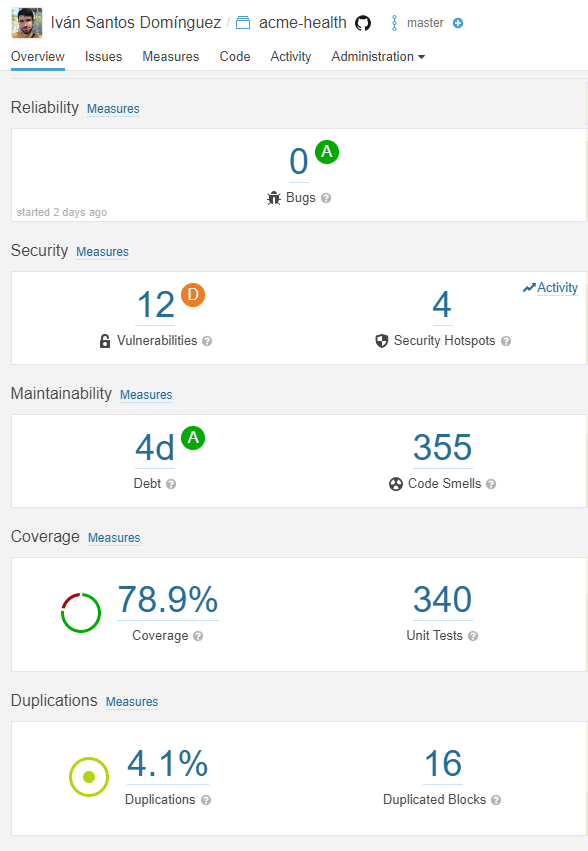
## 5. Refactorizaciones con SonarCloud

El análisis previo realizado con SonarCloud nos mostró los siguientes resultados:



A continuación, se procederá a identificar los malos olores y bugs que nos propusimos a arreglar y explicaciones de cómo lo conseguimos.

5.1. R-001, 002 & 003:



## 6. Justificación de cómo han ido consiguiendo cada uno de los requisitos del nivel señalado

6.1. Sprint 1

En el documento creado, se añadió la descripción a grandes rasgos de la aplicación web, una versión inicial de las historias de usuario y dos escenarios para cada una, tanto positivo como negativo. Además, se creó un documento aparte con el uml dividido en trozos de las distintas entidades que componían nuestra aplicación.

En GitHub, creamos el proyecto y añadimos las distintas tareas (cada historia de usuario correspondía a una tarea), las repartimos a las parejas y las planificamos para los siguientes sprints (creamos un milestone por cada sprint, para que así hubiese una mejor organización).

6.2. Sprint 2

En cuanto a la cronología que se ha llevado a cabo para alcanzar los niveles propuestos, primero se ha ido implementando funcionalidad a la vez que las pruebas, cada pareja ha ido rotando sus roles en las distintas tareas (uno de los miembros implementa una parte de la funcionalidad y el otro la prueba, y viceversa). Cuando ya se implementó y probó el 60% de las historias de usuario que nos propusimos hacer para este sprint, se realizaron las pruebas automatizadas con Travis CI satisfactoriamente. La implementación de la API REST se ha dejado para más adelante por recomendación del profesor, que argumentó que se iba a subir un vídeo explicando cómo hacerlo.

6.3. Sprint 3

Tras la implementación del 60% de las historias y sus correspondientes pruebas unitarias, continuamos implementando las historias de usuario hasta llegar al 90% en la finalización del Sprint 3. Al mismo tiempo, se realizaban las pruebas unitarias de los métodos correspondientes y se crearon pruebas end to end para todos los controladores.

Junto con las demás implementaciones, se integró el servicio externo seleccionado en el sprint 2 (Stripe), mediante su librería, realizando así las pruebas de integración de sus métodos, contenidas en los ficheros: AdminControllerTest, StripeServiceTest, TransactionServiceTest, PaymentMethodServiceTest y PaymentControllerTest (sin contar los E2E).

Mientras algunos miembros implementaban algunas historias, los demás realizaban las pruebas de interfaz de usuario de las creadas anteriormente, y viceversa. Como método para creación de dichas pruebas se usó el Katalon Recorder, creando un test para un escenario negativo y otro para un caso positivo. Al finalizar el Sprint contábamos con un total de 36 pruebas de interfaz para 18 historias de usuario.

Como se ha comentado anteriormente, las parejas se han ido turnando en la realización de las tareas con el objetivo de que todos los miembros tengan contacto con las distintas herramientas del trabajo, por lo que no ha habido ninguna pareja que haya realizado el 100% de un apartado. Es cierto que la pareja formada por José y Paco se ha centrado más en la implementación de historias y la API, y la formada por Iván y Alberto ha trabajado más con los tests, pero no con una diferencia abismal.

6.4. Sprint 4

José Gamaza, con la ayuda de Alberto Murillo se han encargado de realizar la implementación de las dos últimas historias de usuario. Mientras, Iván Santos se encargó de realizar las pruebas de rendimiento para 13 de las 21 historias de usuario. Una vez terminada la implementación, entre todos nos pusimos a realizar las pruebas de los anteriores sprints para toda la nueva funcionalidad añadida, para así llevar al día el proyecto.

Tras esto, José Gamaza realizó una prueba de rendimiento, Francisco Rosa tres y Alberto Murillo dos, dejando así hechas todas las pruebas de este tipo (en total cuentan 19 de 21 historias de usuario, pero las dos restantes al estar involucrada la API no se podían hacer).

Luego, en cuanto a profiling, Francisco Rosa lo realizó para tres historias de usuario e Iván Santos para una, haciendo además para esta la refactorización correspondiente para la mejora de rendimiento.

Por último, Iván Santos se encargó de realizar el análisis de SonarCloud, pero como ya había hecho una refactorización, dejó a los demás miembros del grupo ocuparse de las de diseño (arreglo de bugs y malos olores). Por lo que José Gamaza arregló los 4 bugs que aparecieron, Francisco Rosa arregló 2 malos olores y Alberto Murillo, 6.

## 7. Análisis retrospectivo de los sprints

7.1. Sprint 1

El grupo entero se mostró activo en cuanto a la planificación, ya que consideramos que era lo más importante de todo el proyecto, dado que una mala planificación desde el principio supone una carga de horas mucho mayor conforme va pasando el tiempo.

**Tiempo dedicado estimado:** todo el proceso lo pudimos completar en horario de clase, ya que las clases prácticas consistían en trabajar en grupo. Además, la comunicación presencial fue un factor clave en cuanto a eficiencia y organización.

7.2. Sprint 2

En general, la realización del sprint por parte de cada miembro del grupo en particular y en pareja ha sido muy satisfactoria. Gracias a las historias de usuario realizadas anteriormente cada pareja ha tenido de forma muy clara lo que tenían que realizar y a su vez se han repartido las subtareas en cuanto a partes de la funcionalidad. No hemos tenido ningún problema a la hora de mergear nuestras ramas a master. Esto significa que ha habido una separación e independencia clara en cuanto a las tareas que han ido haciendo los miembros del grupo, y cuando ha habido alguna dependencia se han organizado bien para que no haya ningún problema.

El tiempo dedicado estimado es el siguiente:

|  |  |
| --- | --- |
| Gamaza Díaz, José | 52 h |
| Murillo Jaenes, Alberto | 46 h |
| Rosa Rodríguez, Francisco Javier | 48 h |
| Santos Domínguez, Iván | 50 h |

7.3. Sprint 3

Como grupo la conclusión a la que se ha llegado es que este sprint ha sido el más duro al que nos hemos sometido, ya que no solo tuvimos que implementar bastantes historias de usuario y tests unitarios, sino que también nos enfrentamos con la integración de la API y nuevas pruebas, que, aunque no fueron demasiado complejas, nos consumieron bastante tiempo, lo que llevó a estrés al unirse con las exigencias de las demás asignaturas y la situación que vivimos con el confinamiento.

Aparte de por el tiempo, no hubo ningún problema demasiado recalcable, ya que gracias a las explicaciones subidas a la enseñanza virtual conseguimos realizar todas las tareas hasta el nivel 9.

Centrándonos en las parejas ha habido gran coordinación, ya que ha habido buena comunicación y organización. Cada miembro del proyecto tenía su propia rama, aparte de algunas que se hicieron solo para tareas concretas, como las pruebas de interfaz. Cuando cada pareja terminaba de implementar la parte correspondiente de su historia de usuario o una completa, la subía al repositorio principal para que la otra pareja pudiera hacer los tests correspondientes o continuara implementado historias.

|  |  |
| --- | --- |
| Gamaza Díaz, José | 40 h |
| Murillo Jaenes, Alberto | 52 h |
| Rosa Rodríguez, Francisco Javier | 60 h |
| Santos Domínguez, Iván | 58 h |

7.4. Sprint 4

En este sprint se ha observado especialmente una mejora de conocimientos y eficiencia general, ya que a esas alturas todos los miembros del grupo habían tocado de todo (tanto implementación como todo tipo de pruebas). Las pruebas de los anteriores sprints que había que hacer de la nueva funcionalidad no llevó a penas tiempo por esa misma razón, y las nuevas pruebas a realizar de ese mismo sprint resultaron algo tediosas, pero para nada difíciles, por lo que la moralidad del equipo se vio realmente aumentada. El flujo de trabajo fue constante. Hubo un importante problema relacionado con las pruebas de rendimiento, pero con la ayuda del profesor se pudo solucionar. A parte de eso, como se ha dicho anteriormente, no hubo grandes impedimentos. Por último, comentar que esta recta final del proyecto nos ha parecido bastante interesante, sobretodo con el tema de profiling y análisis de SonarCloud, ya que hemos podido ver y analizar distintas métricas medidas en nuestro propio proyecto. Y gracias a esto hemos podido mejorar considerablemente el rendimiento de nuestra aplicación.

En definitiva, estamos satisfechos de haber cursado esta asignatura ya que pensamos que lo que hemos aprendido en ella nos va a resultar bastante útil en el futuro, ya que los temas tratados durante este cuatrimestre se dan cada día en cualquier empresa.

|  |  |
| --- | --- |
| Gamaza Díaz, José | 80 h |
| Murillo Jaenes, Alberto | 54 h |
| Rosa Rodríguez, Francisco Javier | 50 h |
| Santos Domínguez, Iván | 58 h |