**Profiling**

* HU-001: Listado de los médicos de un centro

Hemos seleccionado esta historia ya que, el pico de usuarios más pequeño en el que se produce el cuello de botella en la aplicación entre nuestras historias, se corresponde con esta.

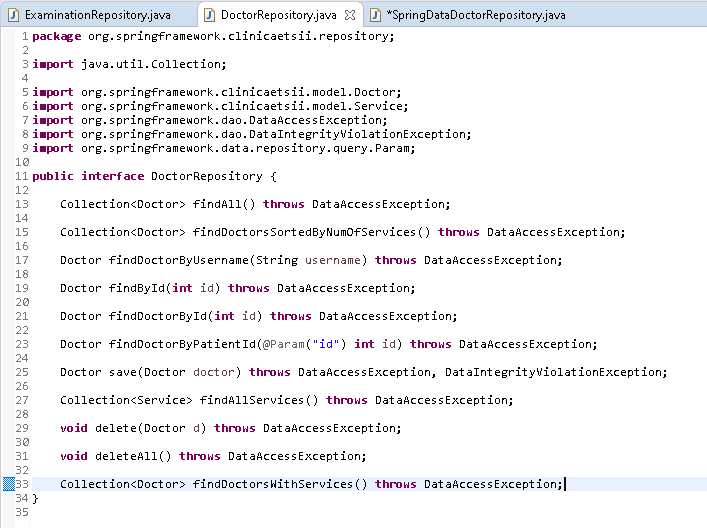
Una vez aplicada la carga de trabajo a la aplicación de unos 40 usuarios anónimos y unos 20 doctores, se obtienen los siguientes tiempos, para las queries utilizadas en el controlador DoctorController:

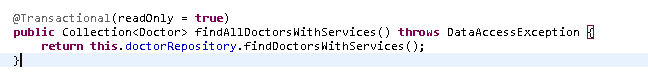
Captura de pantalla de un celular

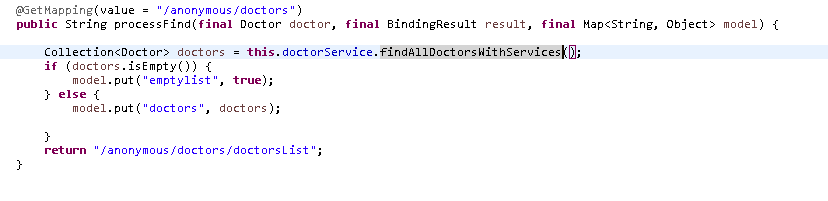
Descripción generada automáticamente

Como podemos observar, para cada uno de los doctores que se listan en la consulta, estamos consultando también los servicios que ofrecen cada uno de estos. Por lo que una de las soluciones para ahorrarnos estas consultas cada vez que consultamos los doctores de la base de datos, sería el traernos los servicios de estos cuando nos los traemos, haciendo así dos consultas en una. Para ello, tendremos que hacer una serie de cambios en diferentes partes de la aplicación:

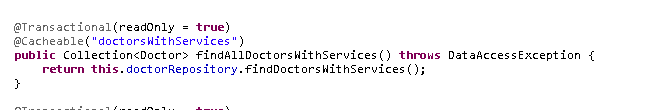
1. Crear la nueva consulta que vamos a realizar, poniendo la nueva cabecera en la clase DoctorRepository y su implementación en SpringDataDoctorRepository:



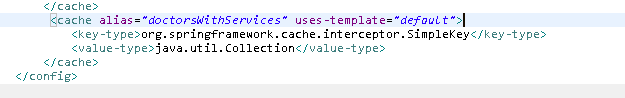


1. Llamar en su controlador correspondiente al nuevo servicio creado.

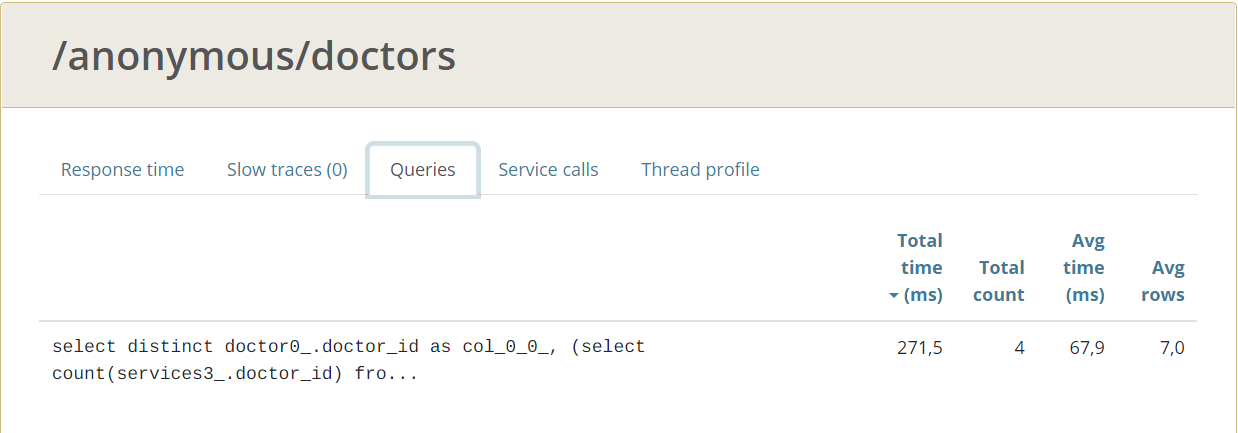
* También hemos cacheado este mismo servicio:



Y lo hemos añadido al respectivo archivo xml ehcache3:

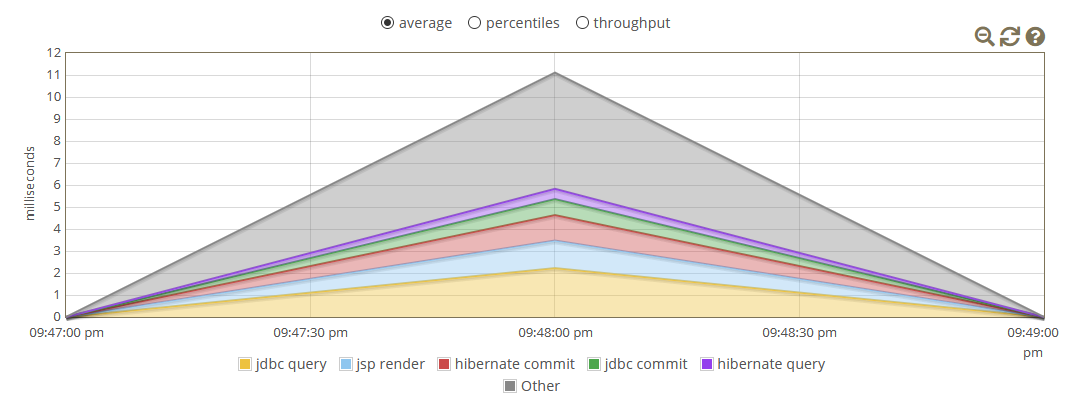


En la siguiente captura, observamos los tiempos de mejora de la consulta. Además, podemos observar que una de las queries, la asociada a la consulta de los servicios del médico ha desaparecido, ya que, en la consulta de este, nos estamos trayendo también los servicios que ofrece, mejorando así la optimización del tiempo empleado:



* HU-005 y HU-006: Cambio de médico de cabecera – Restricción del cambio médico de cabecera

## All Web Transactions

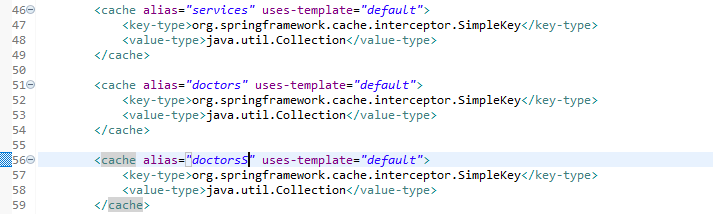


## Queries /patient/edit

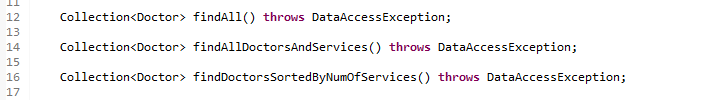


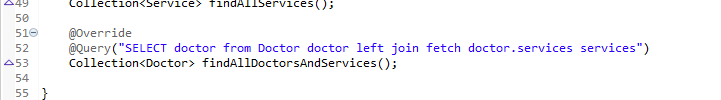
Hemos hecho los cambios convenientes en los servicios, repositorios y controladores, así como en el XML de elementos cachables.

En el XML mencionado, se han añadido las siguientes líneas:

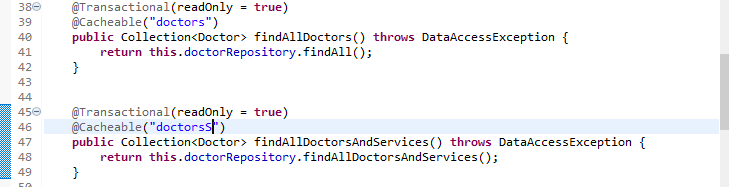


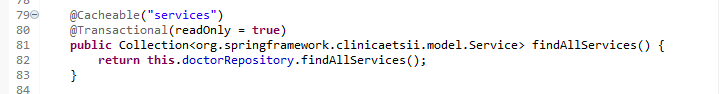
Hemos modificado la clase DoctorRepository, con su correspondiente query.



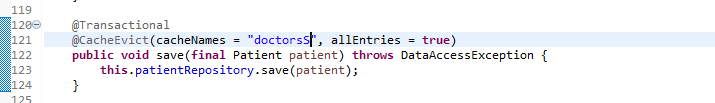


Hemos añadido las anotaciones necesarias en los métodos dentro de la clase DoctorService.

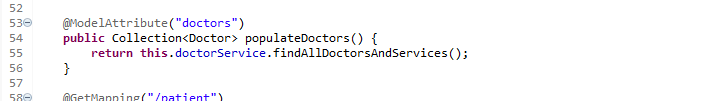




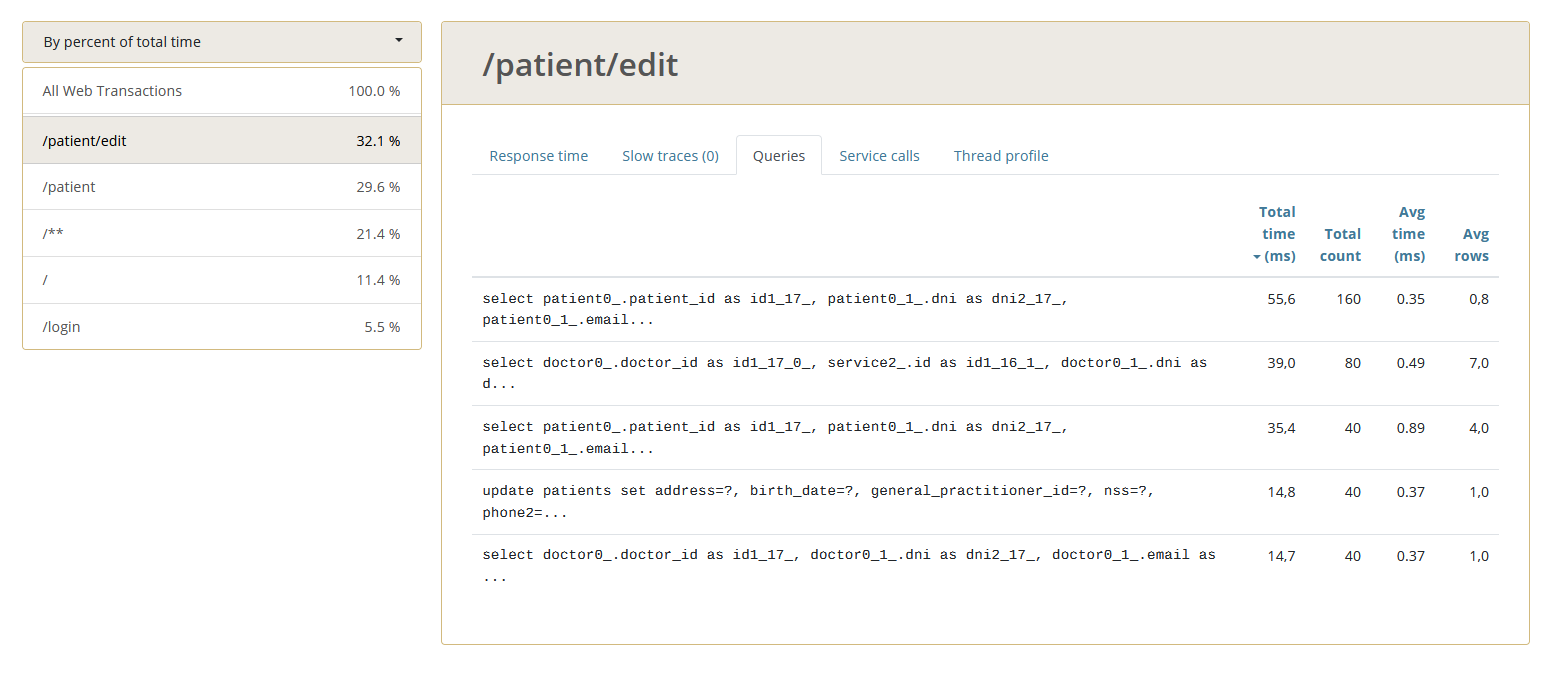
En la clase PatientService, hemos añadido la anotación indicada para vaciar la caché.



Y en la clase PatientPatientController, hemos modificado lo siguiente:

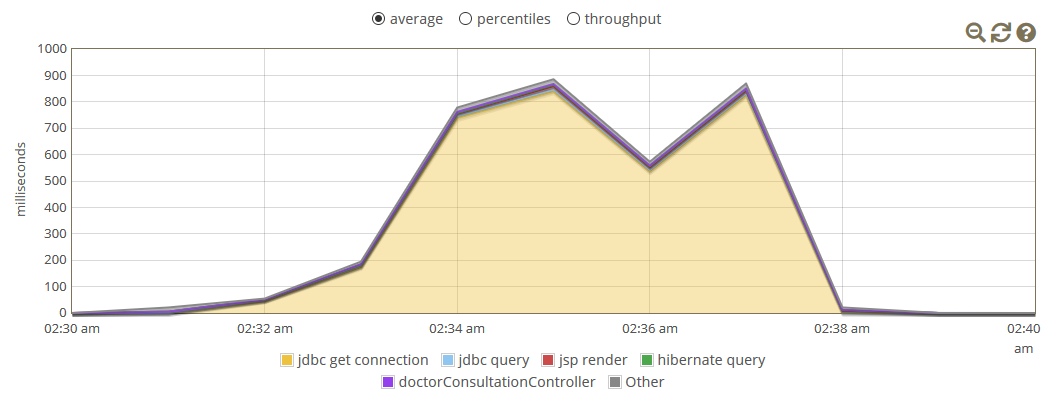


Aunque las consultas no sean muy costosas, cachearlas es una buena práctica, ya que son frecuentes y el resultado de éstas no varía.

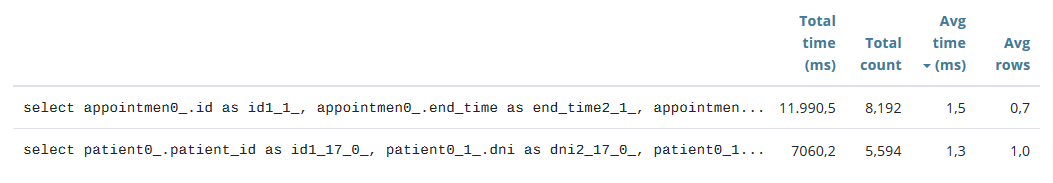


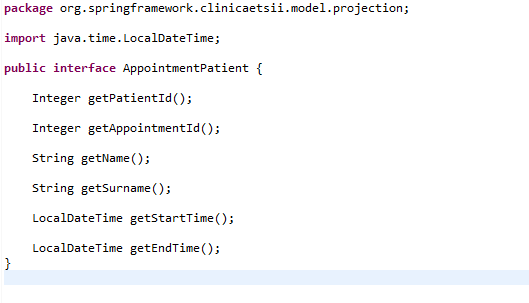
* HU-016: Creación de una nueva consulta a partir de una cita

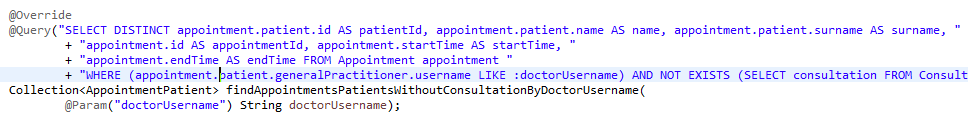
## All Web Transactions



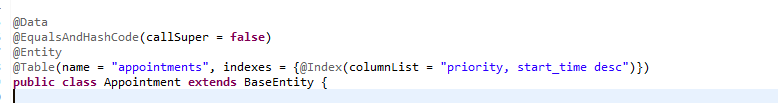
## Queries /doctor/appointments





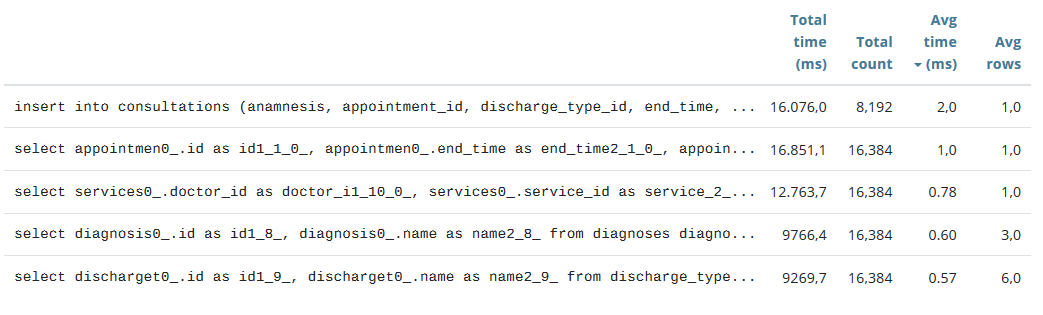


Hemos hecho los cambios covenientes en el servicio, controlador y vista.

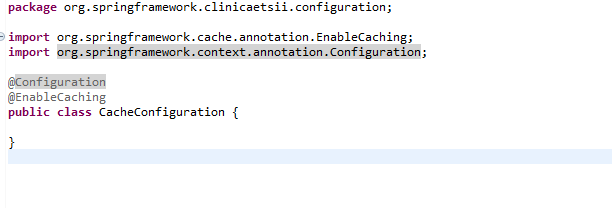
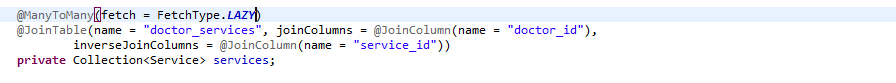


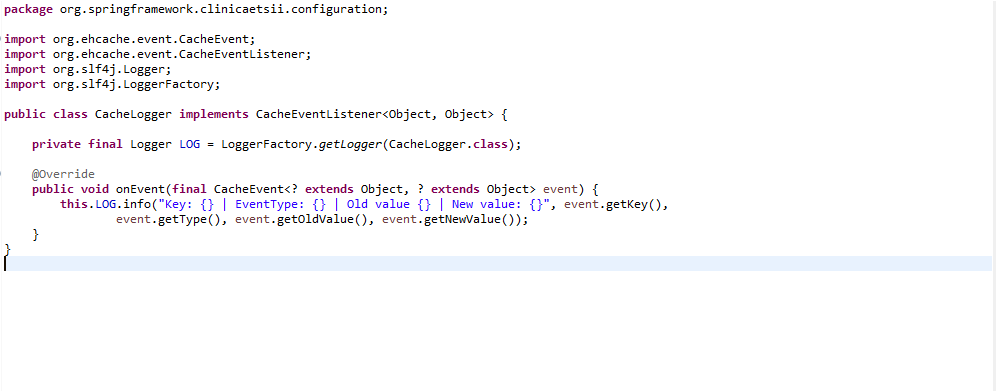
Hemos añadido las columnas utilizadas en la condición de la query como índice de la tabla para acelerar la consulta.

## /doctor/patients/\*/consultations/new



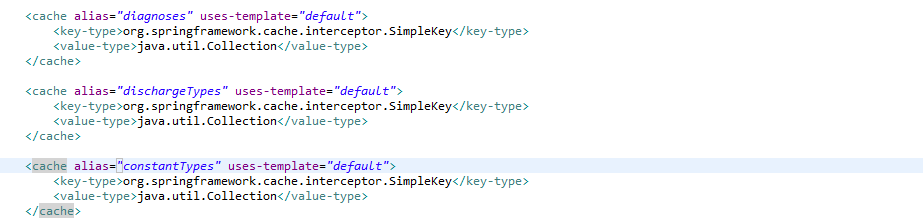
Cambiamos el FetchType para que cada vez que hagamos esas consultas no se traiga todos los

datos si no son necesarios.

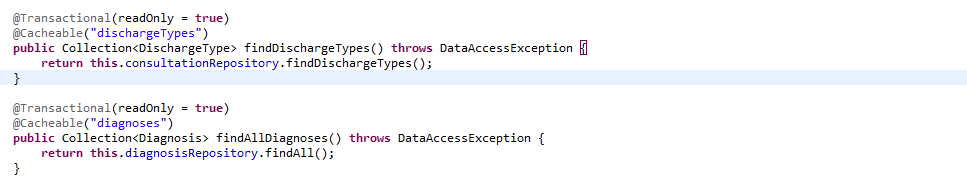


Hemos hecho los cambios convenientes en los servicios, repositorios y controladores, así como en el XML de elementos cacheables.

En el XML mencionado, se han añadido las siguientes líneas:



Hemos añadido las anotaciones necesarias en los métodos dentro de la clase DoctorService.



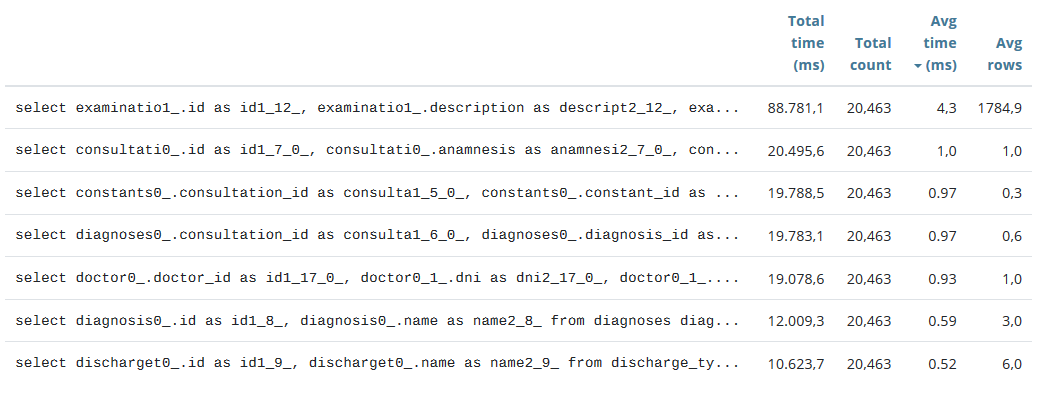
Hemos añadido las anotaciones necesarias en los métodos dentro de la clase ConstantService.



Hemos añadido una cache y tres funciones de los servicios.

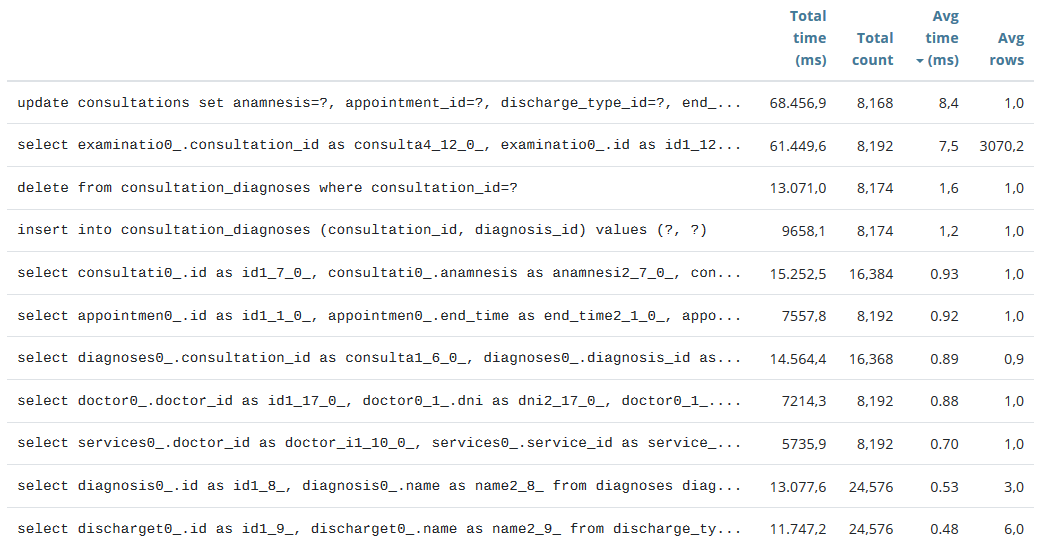
Aunque las consultas no sean muy costosas es bueno cachearlas pues son muy repetitivas y el resultado es siempre el mismo, a parte se utilizan mucho.

**/doctor/patients/\*/consultations/\***

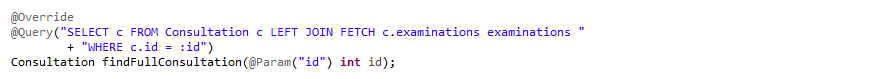


Para este tipo de consulta también se ha utilizado la implementación de la caché antes mencionada.

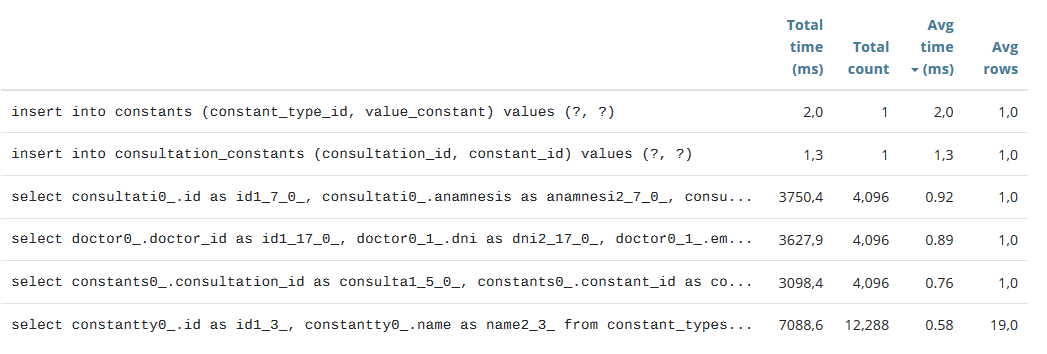
## /doctor/patients/\*/consultations/\*/edit



Para este tipo de consulta también se ha utilizado la implementación de la caché antes mencionada.

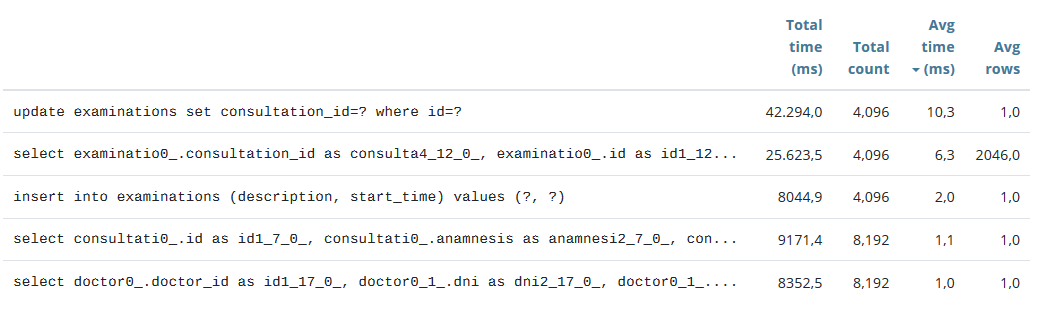


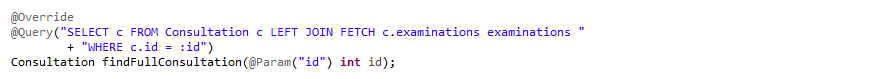
## /doctor/patients/\*/consultations/\*/constants/new



Para este tipo de consulta también se ha utilizado la implementación de la caché antes mencionada.

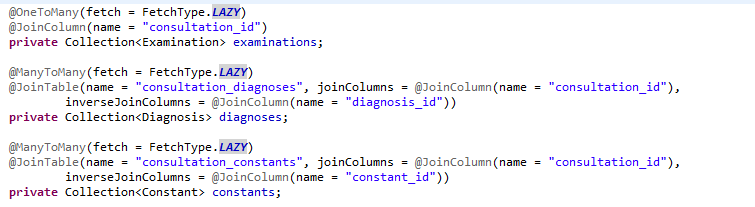
## /doctor/patients/\*/consultations/\*/examinations/new





Cambiamos el FetchType para que cada vez que hagamos esas consultas no se traiga todos los

datos si no son necesarios.



Para evitar consultar examinations, diagnoses y constants siempre que no sea necesario.