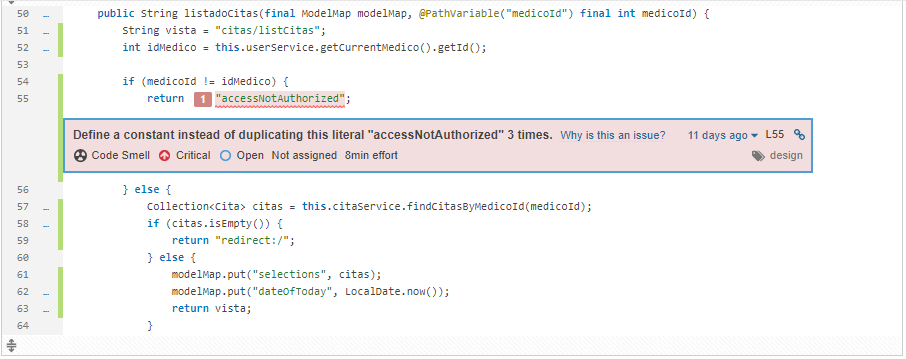
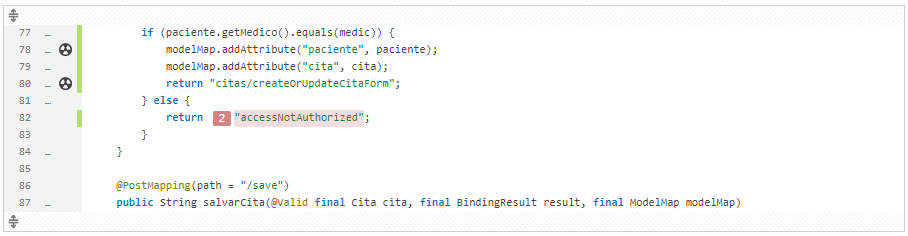
**Refactorización**

En este documento registraremos una refactorización ejecutada en nuestro código después de analizar los resultados de SonarCloud. Una vez ejecutado SonarCloud observamos que teníamos 17 malos olores de nivel crítico, así que decidimos centrarnos en su refactorización correspondiente. De este modo, el análisis llevado a cabo por SonarCloud nos mostraba que la clase CitaController poseía casi un tercio de estos malos olores críticos tal como podemos ver en la siguiente imagen, por lo que decidimos solucionarlos.



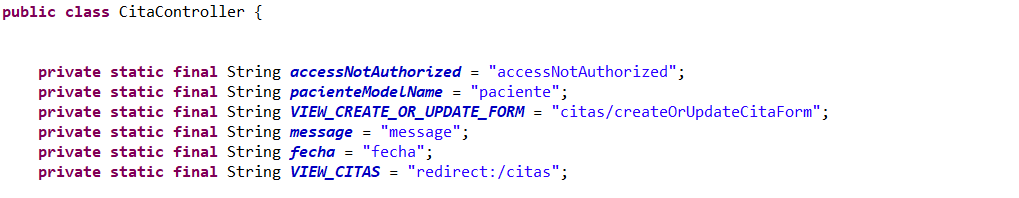
Como podemos observar, estos malos olores críticos son en su totalidad de literales duplicados que se invocan en múltiples ocasiones, por lo que resulta una buena práctica convertirlos en constantes individuales que se invoquen cuando sea necesario. Así pues, observemos un ejemplo otorgado por SonarCloud donde se aprecian estos literales duplicados:







En estas imágenes, SonarCloud nos hace observar la múltiple repetición de un literal que podría ser declarado como constante. Este tipo de mal olor crítico se repite hasta un total de seis veces en esta clase, por lo que en pos de refactorizar nuestro código, procedemos a solucionar estos seis malos olores declarando los parámetros indicados como constantes. De este modo, una vez declaradas las constantes que producen estos malos olores, el código añadido es el siguiente:



Con este fragmento añadido y la correspondiente sustitución de los métodos que usan estos parámetros, los malos olores críticos de esta clase deberían desaparecer. Para comprobarlo, volvemos a ejecutar SonarCloud y el resultado es el siguiente:



Efectivamente, observamos que los malos olores han desaparecido después de la refactorización del código, con lo que podemos concluir que la refactorización ha sido realizada de forma efectiva.g