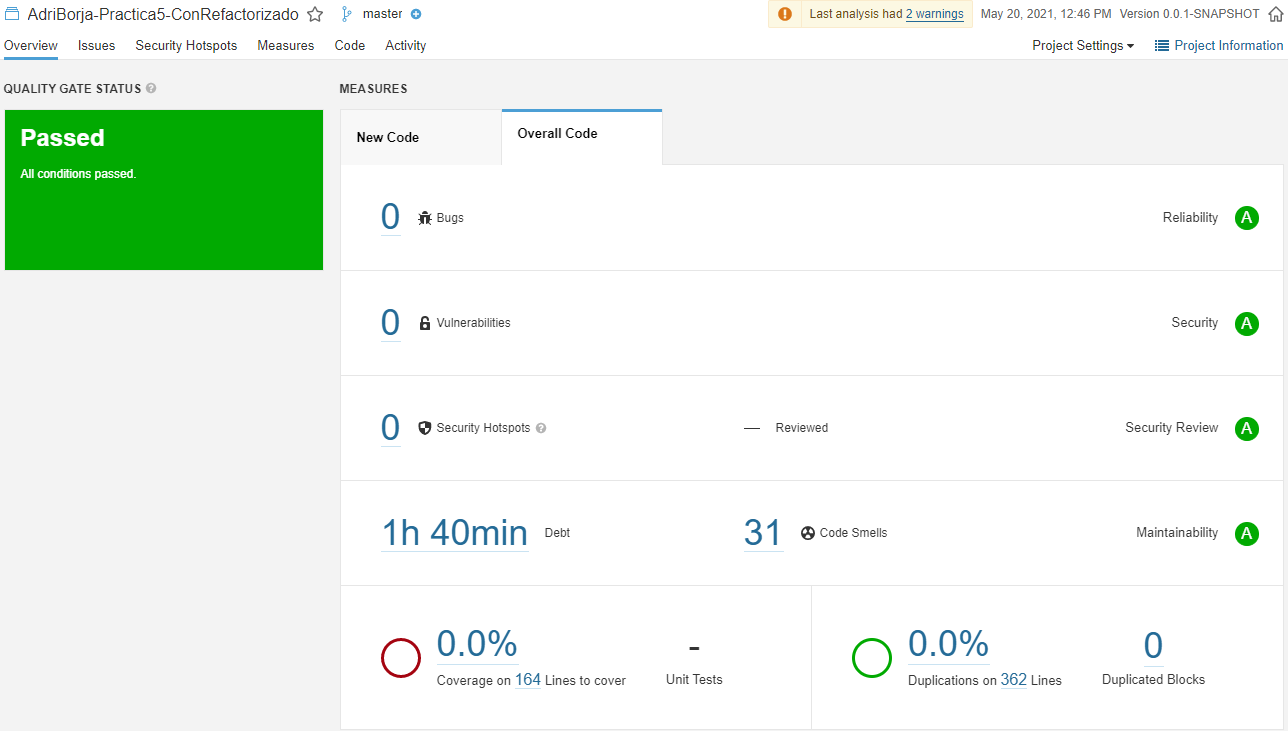
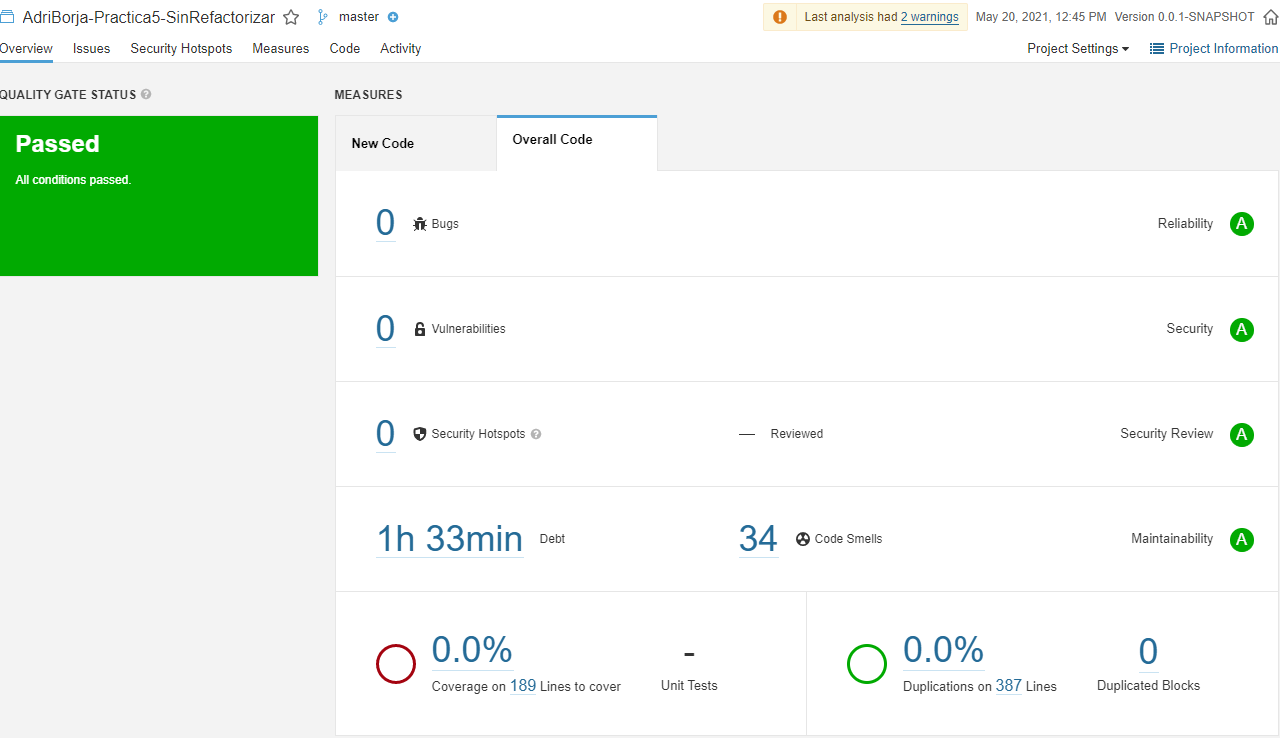
­**4.2 ANALISIS DE LA MEJORA OBTENIDA EN LA PRACTICA DE REFACTORIZACION**

Según el análisis realizado por el software de calidad Sonar, este es el cuadro general de nuestro proyecto REFACTORIZADO

****

Este proyecto tiene una complejidad ciclomática de 63 y una complejidad cognitiva de 14.

Según el análisis realizado por el software de calidad Sonar, este es el cuadro general de nuestro proyecto SIN REFACTORIZADO



Este proyecto tiene una complejidad ciclomática de 70 y una complejidad cognitiva de 24.

Como podemos ver, la deuda técnica del proyecto REFACTORIZADO es de 1h40m, mientras que la deuda técnica del proyecto SIN REFACTORIZADO es de 1h33m, sin embargo, la complejidad del proyecto REFACTORIZADO es bastante inferior a la complejidad presentada en el proyecto SIN REFACTORIZADO. Las refactorizaciones, han disminuido considerablemente la complejidad a costa de un escaso aumento de la deuda técnica.

**4.3 ANALISIS DE CALIDAD DEL CODIGO REFACTORIZADO**

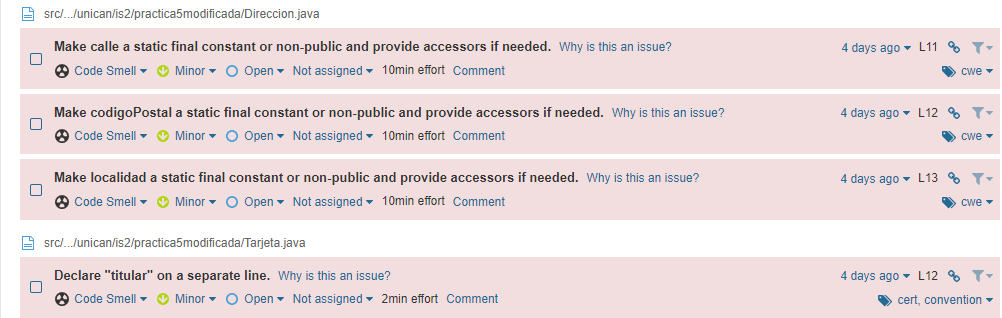
Sonar nos ha marcado 3 fallos mayores y 12 fallos menores. Sonar también nos marcaba los errores de las clases de Test, pero no han sido incluidos para centrarnos mas en el código de negocio.

Los fallos mayores marcados por sonar son los siguientes:



Los fallos menores marcados por sonar son los siguientes:





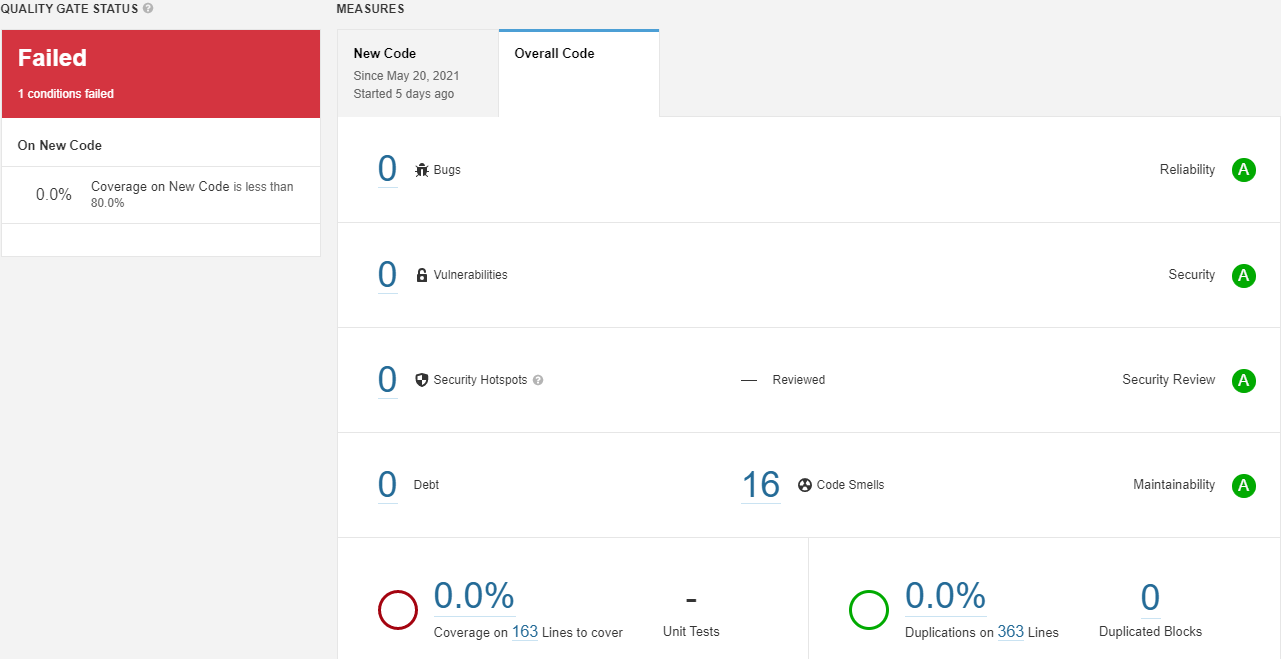
Como se puede apreciar, la mayoría de los fallos son los mismos pero repetidos en distintas clases, como se puede apreciar claramente en los fallos de clase ‘Dirección’ y ‘Cliente’.

**4.4 MEJORA DE CALIDAD**

La prioridad que hemos seguido a la hora de arreglar los fallos es la siguiente:

1. Fallos mayores
   1. Cambios en la visibilidad del constructor
   2. Renombrado de la variable
2. Fallos menores
   1. Realizamos los cambios que no requerían cambiar una variable de publica a protected clase a clase
   2. Realizamos los cambios que requerían cambiar una variable de publica a protected clase a clase

Aunque sonar nos daba una deuda técnica de 1h40m, en realidad el coste de realizar todos los cambios no superaban los 10 minutos, por lo que decidimos solucionar todos los fallos que nos daba sonar. Tras implementar las mejores, los resultados de calidad obtenidos son:



Como se puede comprobar, hemos reducido la deuda técnica a 0, y los 16 ‘code smells’ corresponden a los test, pero no los hemos solucionado porque nos hemos centrado en el código de negocio.

Se ve que sale un fallo arriba a la izquierda pero no tiene importancia en el proyecto, ya que se da porque no se llega al 80% de la cobertura.