Documento de Calidad de Código Fuente
Grupo 11 - 2021

Taller de Sistemas Empresariales
Taller de Sistemas de Información Java EE

Joaquín Cabrera

Agustina Corvo

Bruno Fernández

Gabriel Tulumello

Marcos Pulido

Marzo 2021

Tabla de Contenido

1. Introducción

En el presente documento se detalla el estudio realizado sobre la calidad del código fuente de la plataforma VacunasUY. Se realiza un análisis del código para detectar factores que puedan interferir con el correcto funcionamiento o presente vulnerabilidades de seguridad. En la segunda parte del documento, se detallan las pruebas unitarias realizadas sobre el sistema.

2. Análisis de código fuente

En esta sección se describe el análisis realizado sobre la calidad del código fuente.

2.1. Objetivo

Verificar la calidad interna del sistema realizando un análisis estático del código fuente.

2.2. Herramienta

Como herramienta se decidió utilizar Sonarqube como fue sugerida por el cuerpo docente, la cual permite realizar entre otras funcionalidades, un análisis estático del código fuente en busca de posibles patrones con errores, malas prácticas o vulnerabilidades de seguridad. Dentro de las verificaciones que realiza, se encuentran las siguientes:

- Detección de código duplicado.
- Tamaño de archivos de código.
- Tamaño de métodos.
- Falta de pruebas unitarias, falta de comentarios.
- No adecuación a estándares y convenciones de código (malas prácticas).
- Vulnerabilidades conocidas de seguridad.

Para este caso se utilizó la versión Community de la heramienta, corriendo bajo un ambiente local.

2.3. Resultados

Luego de realizado el primer análisis, se obtuvieron algunos factores a mejorar, como muestra la siguiente imagen:

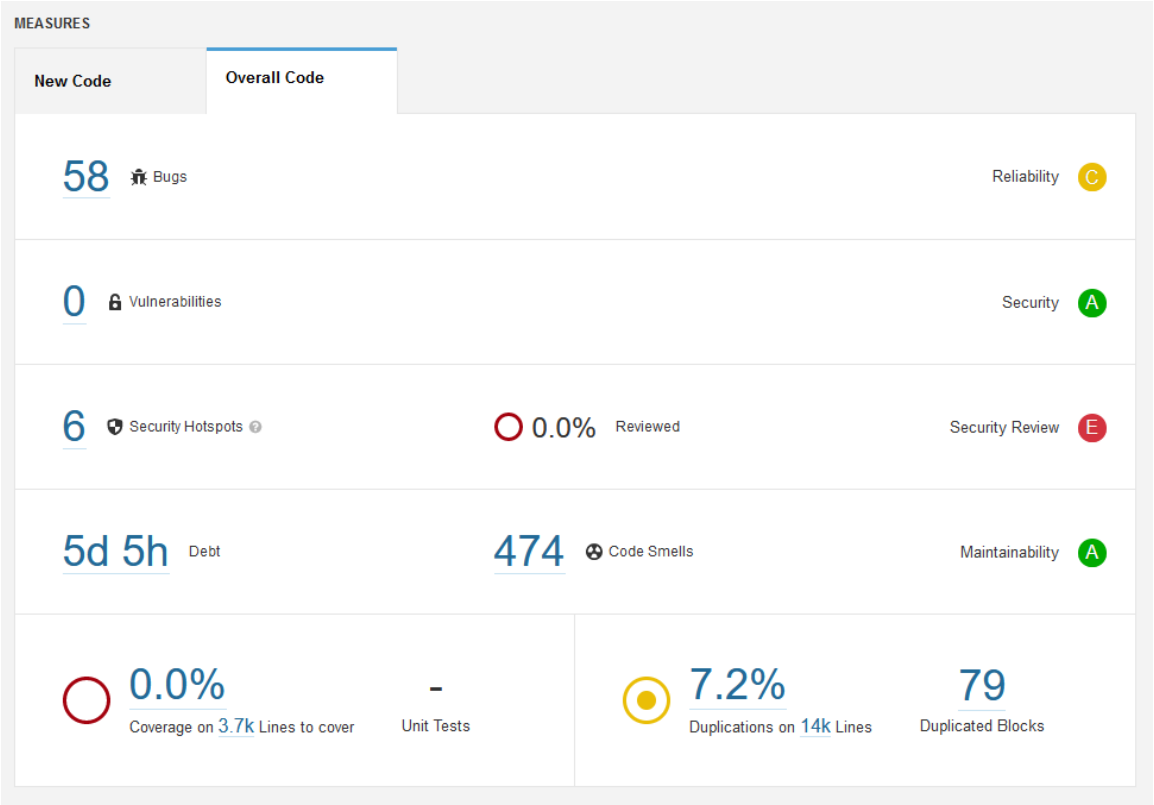

Figura 1: Resumen Análisis 1 - Sonarqube.

Se obtuvo calificación C para el apartado de Bugs, y calificación E para el apartado de seguridad.

La mayoría de los bugs analizados correspondían a comparaciones que se estaban realizando con el doble operador de igualdad, y lo correcto sería realizarlas con la operación Equals.

Respecto a los errores de seguridad, la mayoría fueron causados por librerías que ya no son confiables. Por ejemplo, se utilizó la librería "Math.random"para generar la hora aleatoria para una agenda. La misma tuvo que ser sustituida por una implementación más segura, donde se utilizó "SecureRandom".

Una vez realizadas las correcciones en el código, se lanzó un nuevo análisis, dado esta vez calificación A en los 4 factores.

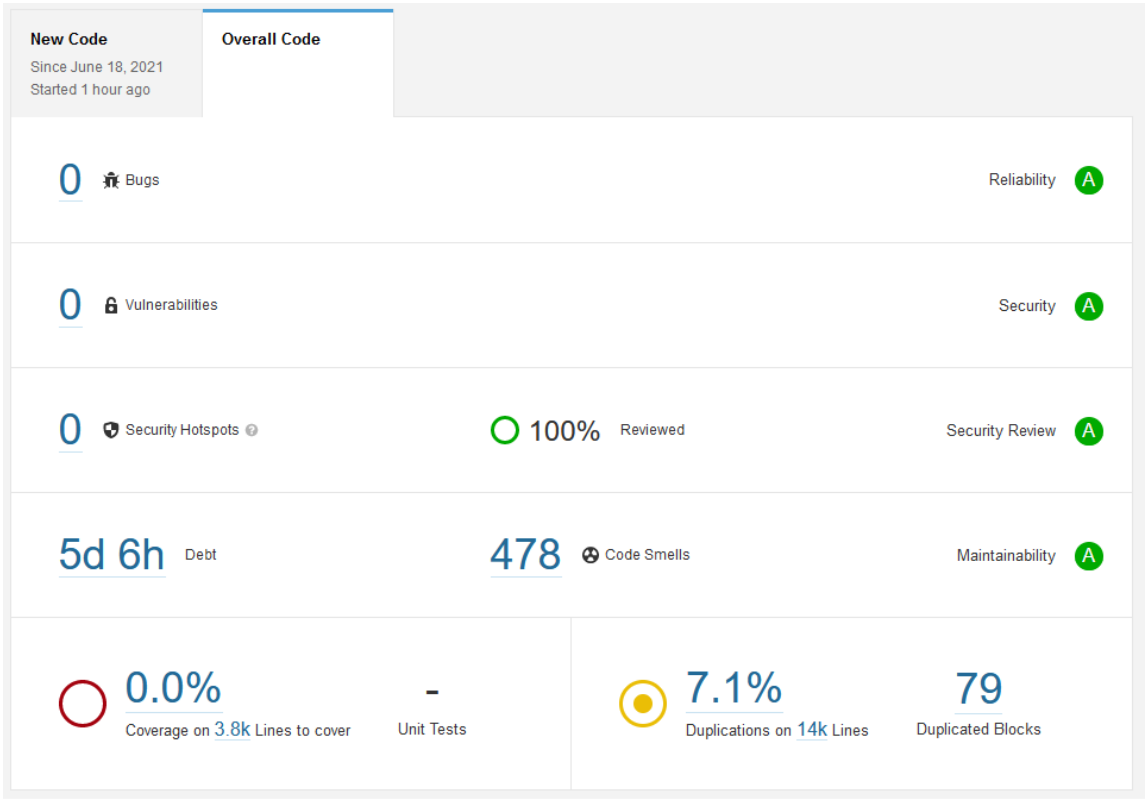

Figura 2: Resumen Análisis 2 - Sonarqube.

Queda como trabajo a futuro realizar una refactorización en el código para disminuir los bloques duplicados y mejorar en ese aspecto.

3. Pruebas unitarias

En esta sección se detallan las pruebas unitarias realizadas sobre el sistema.

3.1. Objetivo

Verificar el correcto funcionamiento de las funciones y métodos correspondientes a la lógica de negocio de la aplicación, cubriendo el 80 % de las funcionalidades en esta capa.

3.2. Herramientas

Como herramientas se decidió utilizar JUnit para realizar las pruebas unitarias y Mockito, a sugerencia del tutor, para simular los componentes que son utilizados por la lógica de negocio.

3.3. Resultados

Se logró cubrir el 80.2 % de la lógica de negocio con pruebas unitarias para asegurar su correcto funcionamiento.

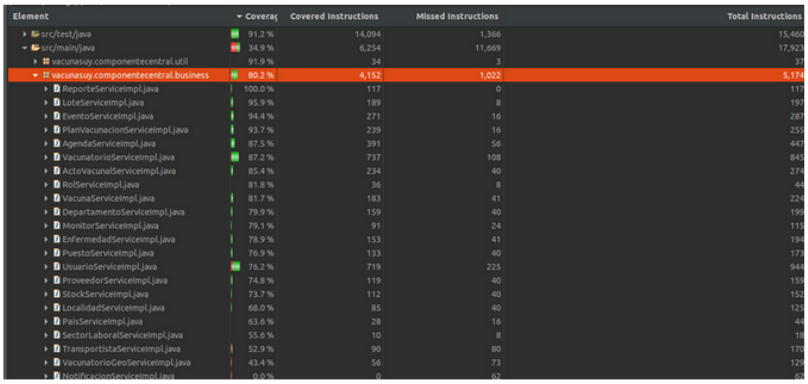

Figura 3: Resultados de coverage - JUnit/Mockito.