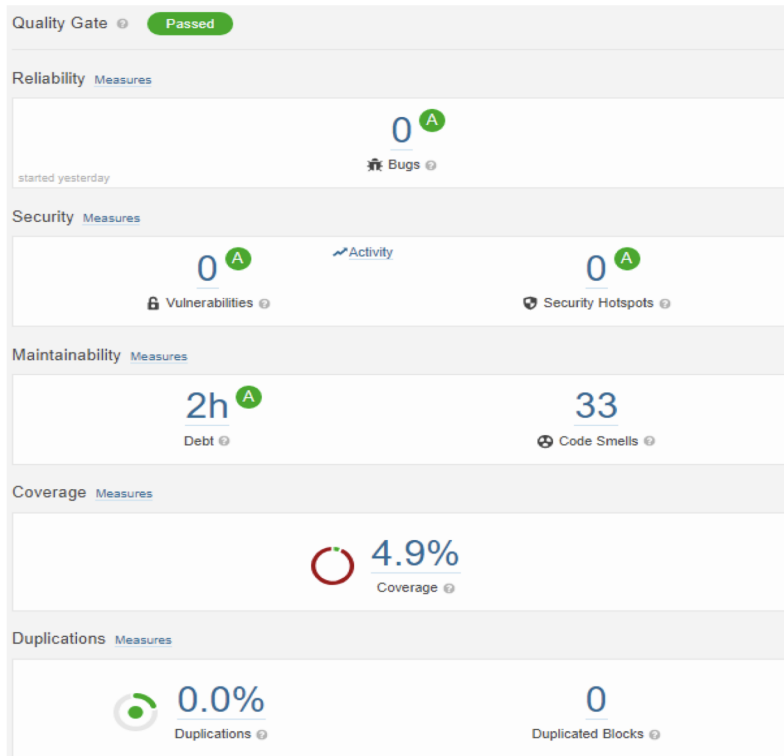


Informe de Calidad (Sprint 1)

Autores: Gonzalo Pauchard y Javier Barquin

ANÁLISIS 15 OCTUBRE 2021 - Gonzalo Pauchard

CAPTURA



INCIDENCIAS

El análisis pasa correctamente los criterios de calidad de la organización. Por otro lado, se encontraron 33 *issues*, 1 *issue* que indica agregar un constructor privado para esconder uno público, 2 *issues* que indican reordenar modificadores por “recomendación de java”, 1 *issue* que indica reemplazar el tipo de especificación en una llamada a un constructor, 1 *issue* que indica usar un acceso del tipo “static” y 30 *issues* que indican remover código que no se está utilizando.

PLAN DE ACCIÓN

- 1) Dentro de la clase *EventsRepository*, arreglar únicamente el *code smell* que aporta 30 minutos de deuda técnica “Add a private constructor to hide the implicit public one” y dejar los otros dos sin resolver.
- 2) Dentro del método *filtrarPorPalabrasClave*, arreglar el único *code smell* que aporta 1 minuto de deuda técnica, ya que es simplemente cambiar `new ArrayList<Event>()` por `new ArrayList<>`, con esto el método indicado quedaría sin ningún *issue*.
- 3) Dentro de la clase *EventArrayAdapter* arreglar únicamente el *code smell* “Use static access with “android.content.Context” for “LAYOUT_INFLATER_SERVICE””, este aporta 20 minutos de deuda técnica al tener que cambiar el acceso a un objeto no derivado.

Informe de Calidad (Sprint 1)

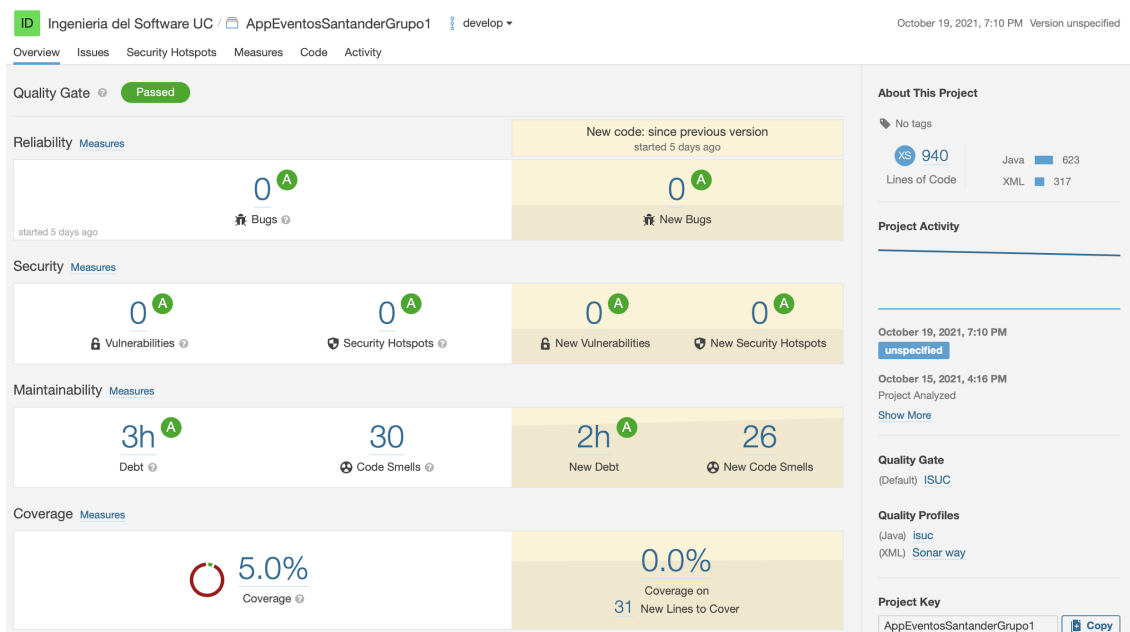
Autores: Gonzalo Pauchard y Javier Barquin

Comentarios:

- Si bien el código pasa los criterios de calidad, consideramos que estos *issues* podrían traer ciertos problemas de compilación más adelante y de todas maneras, pueden ser solucionados de manera sencilla, esto nos permitiría mejorar la calidad del código como tal.

ANÁLISIS 19 OCTUBRE 2021 - Javier Barquin

CAPTURA



INCIDENCIAS

El análisis pasa los criterios de calidad de la organización debido a que la calificación en confiabilidad (*reliability*) es A. Tampoco se encuentran vulnerabilidades nuevas.

Por otro lado, encontramos 30 incidencias (*issues*) (26 nuevas) de mantenibilidad (*code smells*) en clases del modelo de datos: en *EventsActivityUITest.java* **una** única incidencia que suma 20 minutos, en *EventsPresenter.java* **cuatro** incidencias que suman 8 minutos, en *EventsActivity.java* **ocho** incidencias que suman 58 minutos; en las clases *EventsPresenterTest.java* y en *EventsPresenterITest.java* encontramos **ocho** incidencias que suman 34 minutos en cada una, y en *IEventsContract.java* **una** indecencia que suma 2 minutos.

PLAN DE ACCIÓN

- 1) Dentro de la clase *EventsPresenter.java* solamente habría que eliminar los *imports* que no se usan.
- 2) Dentro de la clase *EventsActivity.java*, además de eliminar los *imports* que no se usan, corregir la línea 154 eliminando el *cast* innecesario de (*NetworkInfo*). Además, en la

Informe de Calidad (Sprint 1)

Autores: Gonzalo Pauchard y Javier Barquin

línea 157 arreglar el *bug* para que en lugar de utilizar dos sentencias *return*, se retorne en una única línea con la expresión directamente: `return (info != null)`.

- 3) Dentro de la clase `IEventsContract.java` eliminar el *import* que no se usa.
- 4) En las clases `EventsPresenterITest.java` y `EventsPresenterTest.java`, intercambiar en varias líneas los dos argumentos de las comprobaciones para que estén en orden correcto: valor esperado, valor real.

Comentarios:

- Tras el primer informe de calidad se puede ver que se bajó la deuda técnica de solo 2h a 1h, ya que en esta segunda integración se han añadido 2h (3h en total). Puede deberse a que se hace uso de métodos obsoletos como `NetworkInfo`, que dado a que aún funciona y no hemos encontrado alternativa no hemos podido solventarlo aún.
- Por otro lado, tras el segundo informe se solventaron 29 *code smells*, pero puede haber sido un fallo no repasar profundamente con los demás integrantes del grupo el informe y la posterior solución de problemas de calidad ya que se han repetido algunos errores de la pasada integración.
- En caso de que pase directamente el criterio de calidad podemos establecer un plan de acciones preventivo a la hora de realizar las siguientes implementaciones que elimine alguna de las incidencias más recurrentes que tengamos.