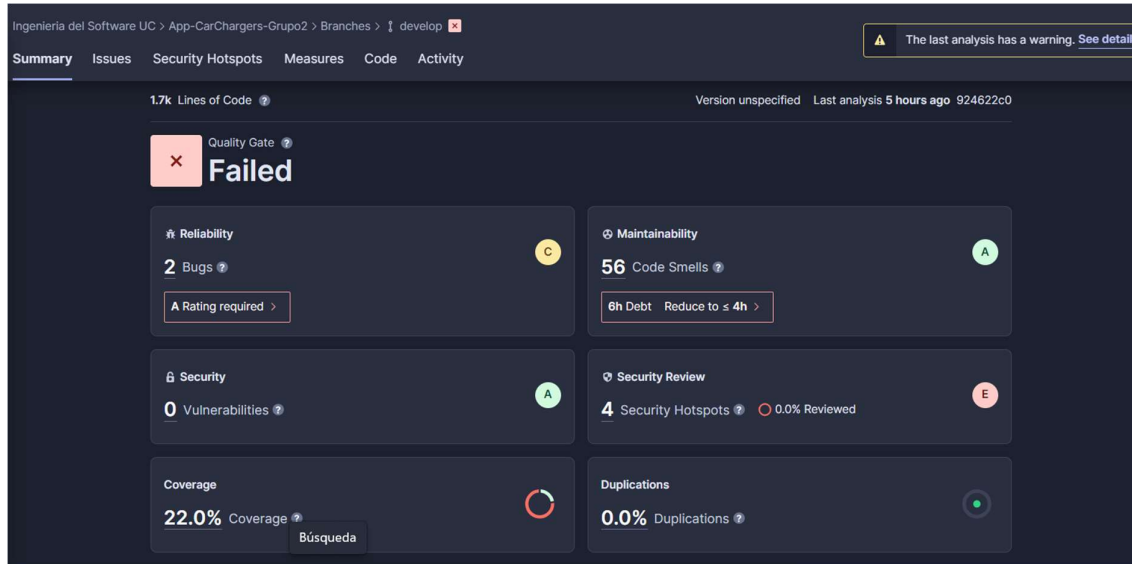


Informe de Calidad (Sprint 2)

Autores: Sergio Algorri, Isaac Berrouet, Adrián Ceballos, Pablo Gómez, Alejandro Sanchís y Carlos Silva.

ANÁLISIS 1 NOVIEMBRE 2023

CAPTURA



INCIDENCIAS

El análisis, realizado sobre la rama *develop*, no pasa los criterios de calidad de la organización debido a que hay 2 *bugs* presentes en el código, lo cual empeora notablemente la calidad del proyecto.

Además de esto, encontramos 56 *issues* de mantenibilidad (*code smells*) que suman un *effort* total de 5h 32min. De estos 56 *code smells*, 3 de ellos son de alta prioridad, mientras que los 53 restantes se reparten de manera más equitativa entre media y baja prioridad.

El primer *code smell* de alta prioridad es debido a que hay un método dentro de la clase *Favourite* que aparece vacío y es necesario añadir un comentario explicando el porqué.

El segundo es del mismo tipo que el primero, aunque esta vez en la clase *FakeCall*, ambos tienen una deuda de tan solo 5 minutos.

El último de estos tiene una deuda de 30 minutos y al igual que el segundo se encuentra en la clase *FakeCall*, aunque esta vez es debido a que hay un método *clone* que es necesario sustituir.

PLAN DE ACCIÓN

Conociendo el tiempo con el que contamos esta semana, tan solo podemos solucionar los dos *bugs* existentes y los 3 *code smells* de alta prioridad.

- 1) Dentro de la clase *FakeCall* arreglar el *code smell* que aporta 5 minutos de deuda técnica *"Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation"*. Además, en esta misma clase es necesario arreglar el *code smell* que aporta 30 minutos de deuda técnica *"Remove this 'clone' implementation; use a copy constructor or copy factory instead"*. Gracias a esto no solo tendríamos la clase *FakeCall* sin ningún fallo, sino que también se arreglarían todos los *code smell* con una prioridad alta.
- 2) Dentro de la clase *Favourite* eliminar el *code smell* mencionado anteriormente, el cual necesitará la misma solución que el existente en *FakeCall*.
- 3) Por último, ambos *bugs* se solucionan de la misma manera y en la misma clase, la cual es *DetailView*. La denominación del *bug* es *"Remove this call to 'equals'; comparisons against null always return false; consider using '== null' to check for nullity"*, aportando un total entre los dos de 30 minutos.

Comentarios:

- En este caso el plan de acciones nos sirve para solucionar posibles errores dentro del código, además de depurar en cierto modo la calidad.
- La elección de los *code smell* a arreglar viene marcada por la prioridad dada a estos. Por ello solucionamos todos los de prioridad alta para evitar posibles fallos en un futuro.