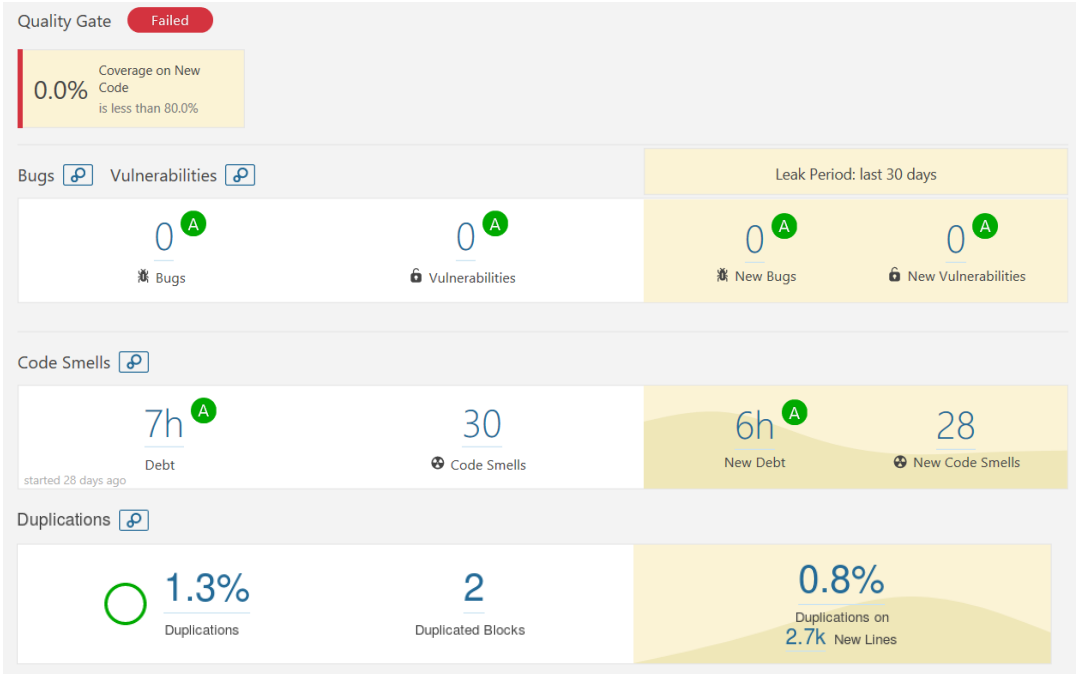


**Autor:** Óscar Alario Pelaz  
**Historia de usuario:** InformarPrecios

**Análisis del proyecto**

Análisis inicial:



El análisis inicial no pasa la quality gate debido a una norma errónea sobre la cobertura de nuevo código. Aun así, el proyecto aprueba con una calificación de A los apartados principales analizados por Sonar (Reliability/Bugs, Security/Vulnerabilities y Maintainability/Code Smells).

El principal problema del proyecto son los 30 code smells, con su respectiva deuda técnica, ya que el código duplicado es solamente de un 1.3%. Aun así, ni los code smells ni la deuda técnica son demasiado elevados, especialmente si los comparamos con valores obtenidos en sprints anteriores en los que se superaban los 110 code smells y los 2 días de deuda técnica.

Deuda técnica inicial:



Toda la deuda técnica es de color verde, y por tanto de la menor gravedad posible, por lo que no hay que preocuparse demasiado por ella. Las clases con mayor deuda son ParadasFragment, LineasFragment y EstimacionesFragment.

Pese a todo, para intentar mejorar aún más la calidad del código se han tratado algunos code smells, priorizando siempre la corrección de los code smells cuya severity es mayor. En este caso, de los treinta code smells, trece tienen severidad info, ocho tienen severidad menor, seis son mayor, tres son critical y no hay ninguno que sea blocker.

Los tres code smells con severidad critical no han sido corregidos debido a que son code smells surgidos de la alta complejidad de algunos métodos del proyecto, los cuales tienen que tener dicha complejidad por las operaciones que realizan.

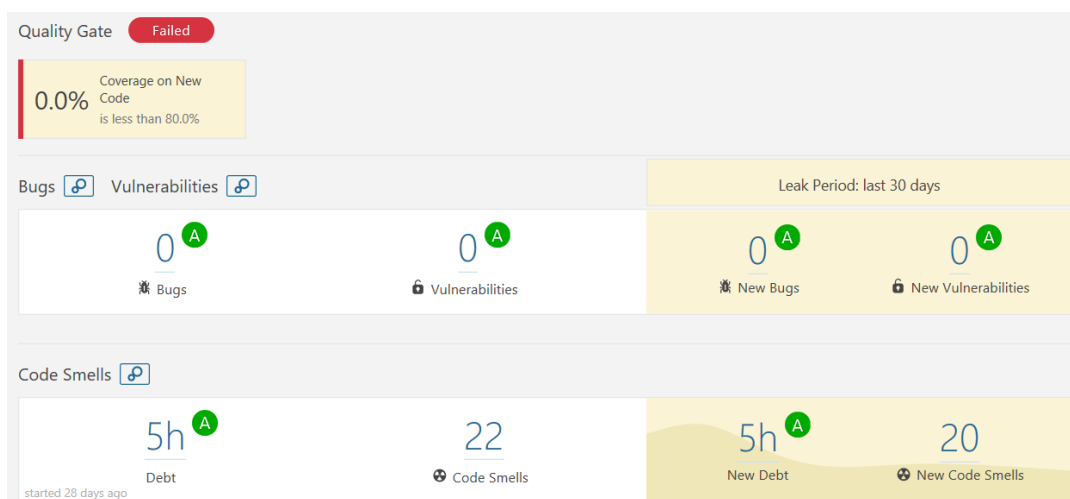
De los seis majors, he corregido tres: dos eran debidos a que había dos bloques de código comentado en la clase MainActivity, por lo que para corregirlos simplemente he eliminado dichos bloques, y el otro se debía a que había un atributo en la clase ParserJSON que no se utilizaba. Éste también ha sido eliminado.

En cuanto a los ocho code smells con severidad menor, cuatro han sido los corregidos: los dos primeros se producían porque en las clases ListLineasPresenter y ListParadasPresenter había un atributo de tipo string al que se le asignaba directamente la URI de la base de datos. Para solucionarlo, he creado un nuevo string en el fichero strings.xml con el valor de la URI y después dicho string lo he utilizado para crear el atributo. Los otros dos se correspondían con la encadenación de varios strings en las mismas clases mencionadas antes. La solución ha sido utilizar la clase StringBuilder para encadenarlos.

Finalmente, de los trece code smells con severidad info, solamente he corregido uno ya que los otros son debido a la utilización de la clase deprecated ProgressDialog, la cual es necesaria para el correcto funcionamiento del proyecto. El code smell corregido se producía en la clase TarifasFragment y era debido a que se hacía un cast a Button innecesario. Para arreglarlo simplemente he borrado dicho cast.

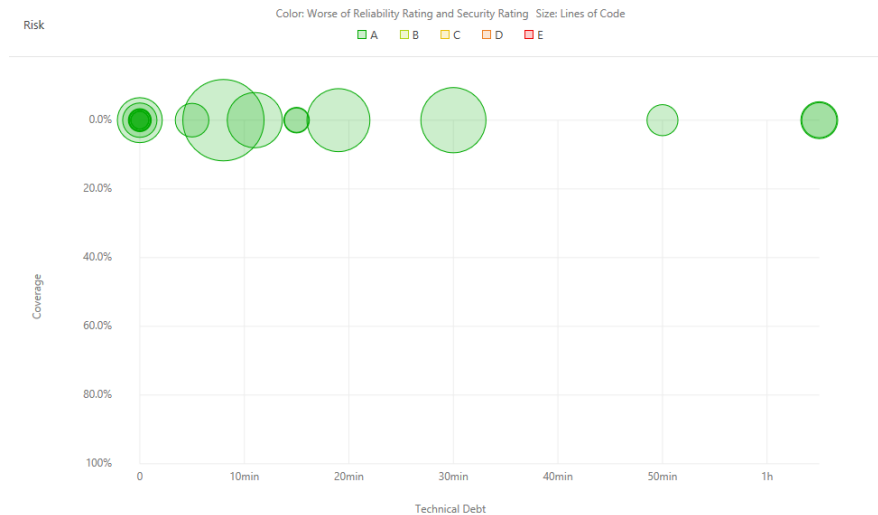
Además de los ya comentados, parte de los code smells restantes no se han solucionado ya que su arreglo produciría otros problemas aún mayores. Es por ello por lo que se esperará a otro momento más adecuado para solucionarlos.

Análisis final:





## Deuda técnica final:



Como era de esperar, el proyecto sigue cumpliendo los requisitos pedidos. Se han reducido en ocho los code smells, se ha bajado la deuda técnica de siete a cinco horas y el código duplicado se mantiene en un porcentaje muy bueno (1.3%). La distribución de la deuda técnica es muy similar a la situación inicial ya que las clases más afectadas por ella siguen siendo `ParadasFragment`, `LineasFragment` y `EstimacionesFragment`.