

Análisis de calidad

1. Introducción

El propósito de este documento es dejar constancia de los resultados de los dos escaneos del código de la historia de usuario US-241258-BuscarLinea, así como de las medidas tomadas para corregir los errores y su justificación.

2. Descripción del análisis y medidas tomadas

Antes de subir el código a *SonarCloud* se observaba que la calidad del código es de **A** en bugs (0), vulnerabilidades (0), deuda técnica (1d) y code smells (40).¹

Tras subir el código nuevo de la historia al analizador se siguen obteniendo calificaciones **A** en todos los apartados. Aún así, se generan 14 nuevos code smells, aunque sólo 6 de ellos son “reales” (el administrador de SonarCloud desactivó algunas reglas dejando algunos como “Info”). Además, un 0% del código se encuentra duplicado. Con el fin de mejorar al máximo la calidad del código y reducir la deuda técnica para el futuro se ha decidido tomar las siguientes medidas:

- **Añadir constructor privado en la nueva clase CommonUtils (mayor):** Esta clase contenía dos métodos estáticos. Como no se va a usar como objeto es necesario ocultar el constructor que se añade por defecto.
- **Eliminar línea de código comentada en CommonUtils (Mayor):** En la línea 34 de la clase existía una línea de código comentada que era necesario borrar, ya que no se usa.
- **Eliminar el atributo “mlineasTodasFragment” de la clase LineasFragments (Menor):** Se elimina ese atributo ya que no se usa.
- **Eliminar import no usado en la clase LineasFragment (Menor):** Se hacía un import de ListLineasAdapter que no se usa en ninguna parte.

Tras volver a subir el observa que los code smells han bajado a 42, siendo la deuda técnica de un día y manteniendo las tres calificaciones de **A**.

Para concluir con el análisis hay que decir que los code smells que se dejan tras esta corrección son:

- **Sobreescribir el método equals de la clase Línea (menor):** Se deja sin corregir porque se cree innecesario. El método compareTo de esa clase no usa el equals, y no se necesita en ninguna otra parte del código.
- **Eliminar el atributo mBottomBar de la clase MainActivity y declararlo como variable local (menor):** Este aviso es un error de SonarQube. Ese atributo corresponde la barra de navegación de la Activity Main, se inicializa en el onCreate y se le asigna un método manejador. Si se declarara localmente, el objeto se destruiría al acabar el método y *crashearía* la aplicación.
- **Eliminar el atributo context en la clase ListLineasPresenter (mayor):** Tras consultar al equipo de desarrollo se ha decidido no corregirlo porque se usará en versiones posteriores de la aplicación para añadir métodos manejadores. Si se quitara, se añadiría más deuda técnica para volver a refactorizar el código y dejarlo como está ahora.
- **Refactorizar un método con un if demasiado grande en la clase ListLineasAdapter (crítico):** Añade demasiada complejidad, pero tras hablarlo en su fase de desarrollo, no se encontró una manera más eficiente de hacerlo en Android.

Los otros 38 code smells están marcados como “info”. Han sido semi-desactivados por el administrador de SonarCloud, ya que la mayoría de ellos carecen de sentido (p.e. marcar métodos deprecados de Android que realmente no lo están) y por lo tanto, se entienden como *falsos positivos*.

¹ Ver informe de calidad de la historia US-241261-Ver paradas de la línea. Es al que se corresponde dicha calidad de código.