

Informe de Calidad: Historia de usuario “US-248546-Añadir comentario a parada”



Resumen

A continuación, se realizará el análisis de calidad del código que da funcionalidad a la historia de usuario “US-248546-Añadir comentario a parada”.

1. Introducción

El documento contiene el análisis de calidad realizado a la implementación de la historia de usuario “US-248546-Añadir comentario a parada”. En este informe se detallarán los análisis realizados, los errores encontrados y las diferentes medidas correctoras ejecutadas sobre el proyecto.

2. Análisis inicial

La situación del código al inicio del análisis mostraba una puntuación A en las tres medidas, Reliability, Security, Maintainability, respectivamente.

A pesar de obtener la máxima calificación en los 3 apartados anteriormente mencionados, aún existe margen de mejora de la calidad del código. Esto se debe a que en el apartado de Maintainability se han generado cincuenta y un nuevos code smells que generan un total de seis horas de nueva deuda técnica. En los apartados restantes se considera que se ha alcanzado la calidad óptima debido a que no se han encontrado nuevos bugs en el apartado de Reliability y tampoco nuevas vulnerabilidades en el apartado de Security, por tanto, el margen de mejora es nulo en estos dos casos.

La forma en la que se han conseguido mejorar dichos resultados será explicado en los siguientes apartados.

En esta nueva historia de usuario existe código duplicado en la clase ParserJson. Se decide no corregir dicho error ya que permiten una mejor distinción de cuando se necesita leer una parada o bien una parada de línea.

3. Solución a los errores encontrados

Como se ha comentado anteriormente, el margen de mejora de la calidad del código del proyecto se basa en tratar de resolver al máximo el número de code smells generados.

Para establecer un cierto orden, se han corregido en un primer instante los errores críticos, después los errores mayores y por último los errores menores. Es decir, el orden establecido para corregir los errores ha sido en base a la importancia de éstos para la calidad del código. Los code smells cuya gravedad se marquen como “info” no serán tratados al considerarse poco relativos.

- Por tanto, los errores de carácter crítico que se han encontrado, así como la estrategia de mejora establecida, son los siguientes:
 - **ParserJson:** se establecerán dos constantes para los textos “UTF-8” y “resources” ya que se utilizan en repetidas ocasiones.

Los errores críticos generados con esta historia de usuario quedarían solucionados.

- A continuación, realizaremos el mismo proceso con los errores mayores:
 - **ParserJson:** se sustituirán dos imports que hacen referencia a las dos clases existentes en el paquete dao por un solo import que incluya el paquete dao sin detallar las clases.
 - **ListLineasPresenter:** se indica que el atributo de tipo Context nunca llega a ser utilizado más allá de su asignación. En el futuro este atributo tendrá funcionalidad por lo que se elige no solucionar este error. Supondría una pérdida de tiempo en un futuro buscar la forma de conseguir ese atributo cuando ya ha sido realizado previamente dicho trabajo.
 - **DetallesLineaActivity:** se encuentra una parte de código comentada, la cual será eliminada.
 - **MainActivity:** se encuentra una parte de código comentada, la cual será eliminada.
 - **ParadasFragment:** se encuentra un atributo que no realiza ninguna función en el proyecto, este deberá ser eliminado. Se encuentra una parte de código comentada, la cual será eliminada.

Los errores mayores generados con esta historia de usuario quedarían solucionados.

- A continuación, realizaremos el mismo proceso con los errores menores:
 - **DetallesParadaActivity:** se eliminarán los imports que no son utilizados en el código de la clase.
 - **Database:** se produce una mala declaración de un array, se modificará siguiendo la estética establecida.
 - **DetallesParadaPresenter:** se eliminarán los imports que no son utilizados en el código de la clase.
 - **ListParadasLineaPresenter:** se eliminarán los imports que no son utilizados en el código de la clase.
 - **ListParadasLineaAdapter:** se eliminarán los imports que no son utilizados en el código de la clase.
 - **LineasFragment:** existe el elemento ';' al finalizar una línea vacía, este deberá ser eliminado.
 - **ParadasFragment:** existe el elemento ';' al finalizar una línea vacía, este deberá ser eliminado.
 - **ParadasTodasFragment:** se eliminarán los imports que no son utilizados en el código de la clase.

Los errores menores generados con esta historia de usuario ya integrada en el proyecto quedarían solucionados.

4. Análisis final.

Después de tratar de corregir los errores citados en los anteriores apartados. Nos encontramos que la situación del código continúa siendo de A en los tres apartados. Se ha reducido el número de code smells a veinticuatro, disminuyendo la deuda técnica generada por el nuevo código, hasta un total de cuatro horas.

5. Conclusión.

En este entregable, se ha expuesto el análisis de calidad de la historia de usuario *“US-248546-Añadir comentario a parada”* detallando los cambios que se han ido realizando en la implementación para mejorar la calidad del código.

Se puede concluir que con el análisis realizado se ha conseguido mejorar razonablemente la calidad del código.