



PS17 – Informe de Calidad: Historia de usuario “Listar paradas”

Resumen

A continuación, se realizará el análisis de calidad del código que da funcionalidad a la historia de usuario *Listar paradas*.

Document ID:	PS17/00/2017-QR001-US243482-ListarParadas
Departamento:	Calidad y auditoría, dentro del proyecto integrado.
Tipo:	ANÁLISIS
Privacidad:	CONFIDENCIAL
Estado:	ENTREGABLE
Versión:	1.0.1
Fecha:	02/11/2017
Autores:	Cerezo Fernández, Elsa
Revisores:	Solar Iglesias, Fernando; Oslé García, Luis; Sainz-Maza Ruiz, Javier; Martínez Vila, Javier

HISTORIAL DE CAMBIOS

Versión	Fecha	Cambio	Responsable
1.0.0	01/11/2017	Creación del documento y redacción.	Cerezo Fernández, Elsa
1.0.1	02/11/2017	Continuar la redacción de los apartados.	Cerezo Fernández, Elsa

1. Introducción

El documento contiene el análisis de calidad realizado a la implementación de la historia de usuario *Listar paradas*. En este escrito se recogen los análisis inicial y final de la implementación, los fallos encontrados y las medidas correctoras que se han aplicado.

2. Análisis inicial

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

Además, el código implementado para dar funcionalidad a la historia de usuario *Listar paradas*, contaba con treinta y seis *issues*, dos vulnerabilidades y, por ello, cuatro horas de deuda técnica.

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

3. Errores encontrados y estrategia de mejora

Los principales errores encontrados son dos vulnerabilidades, un *code smell* crítico y nueve mayores.

Primeramente, se han corregido las dos vulnerabilidades que, aunque se etiquetan como fallos de carácter menor, al eliminarlas conseguimos una puntuación de A en *Security*, mejorando así dicha medida.

- > Las vulnerabilidades anteriores se arreglaron sustituyendo la llamada *e.printStackTrace()* por su implementación con *Log*.

El siguiente error reparado se trata de la carencia del caso *default* en el *switch* que aparece en la clase *MainActivity*. Se ha tomado la decisión de arreglar este error dado que estaba etiquetado como crítico.

- > El error comentado se enmendó añadiendo el campo *default*.

Por otra parte, se ha decidido dedicar tiempo a arreglar el *issue* que se va a explicar a continuación, dado que el esfuerzo que había que dedicarle era de mayor valor que el resto de errores de carácter mayor. El error se trata de la falta de un constructor en la clase *ParserJson*.

- > El error se reparó añadiendo un constructor a dicha clase. Sin embargo, como en este caso no nos interesa tener un constructor en la clase lo creamos con visibilidad privada.

Los demás errores de carácter mayor también han sido reparados y se explicará a continuación cómo han sido resueltos:

- > Faltaba de añadir la sentencia *@Override* a un método en el interior de la clase *ListParadasPresenter*.
- > En las clases *MainActivity* y *ParadasFragment*, había código comentado que podía ser eliminado.
- > En las clases *MainActivity* y *ParadasFragment*, se podían encontrar atributos que no eran usados, por lo tanto, han sido eliminados.

- > En la clase *ParadasFragment*, se había utilizado la concatenación de un *String* vacío y un *int*, para mostrar por pantalla el *int* como *String*, se debía utilizar el método *toString()* de la clase *Integer*

4. Análisis final

La situación final del código tras aplicar las correcciones indicadas en el anterior apartado, muestra una puntuación A, A, A en las medidas *Reliability*, *Security*, *Maintainability*, respectivamente.

Además, se han eliminado las vulnerabilidades, los *issues* críticos y mayores. Se ha conseguido disminuir el número de *code smells* de treinta y seis, que existían en un inicio, a veintiocho. Además, cabe resaltar la reducción de la deuda técnica de cuatro a tres horas.

5. Sumario

En este entregable, se ha expuesto el análisis de calidad de la historia *Listar paradas* detallando los cambios que se han realizado en la implementación para mejorar el código.

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