



PS17 – Informe de calidad asociado a la historia de usuario “Añadir parada a grupo”

Resumen

El presente documento recoge el análisis de calidad asociado a la historia de usuario 242777 – Añadir parada a grupo.

Document ID:	PS17/00/2017-QR006-US242777-AñadirParadaAGrupo
Departamento:	Calidad y Auditoría, dentro del proyecto integrado
Tipo:	Análisis de calidad
Privacidad:	CONFIDENCIAL
Estado:	ENTREGABLE
Versión:	1.0.5
Fecha:	21/11/2017
Autores:	Oslé García, Luis
Revisores:	Martínez Vila, Javier; Cerezo Fernández, Elsa

HISTORIAL DE CAMBIOS

Versión	Fecha	Cambio	Responsable
1.0.0	21/11/2017	Creación del documento.	Oslé García, Luis
1.0.1	21/11/2017	Redacción parcial del análisis de calidad.	Oslé García, Luis
1.0.2	28/11/2017	Completada redacción del documento.	Oslé García, Luis
1.0.3	29/11/2017	Corrección sintáctica del documento.	Martínez Vila, Javier
1.0.4	29/11/2017	Corrección ortográfica del documento.	Martínez Vila, Javier
1.0.5	29/11/2017	Revisión final y cambios menores.	Cerezo Fernández, Elsa

1. Introducción

El presente documento contiene el proceso de análisis de calidad llevado cabo para la historia de usuario #242777 – Añadir parada a grupo.

2. Desarrollo del proceso de análisis de calidad

2.1. Análisis inicial

Para efectuar el análisis de calidad de la historia de usuario previamente citada y realizar las correcciones pertinentes, se procedió como se describe a continuación.

En primer lugar, se subió el código del proyecto al servidor *Sonarcloud*. El análisis inicial del código arrojó los siguientes resultados:

- **Deuda técnica:** seis horas.
- **Medidas e issues:**
 - Cuatro bugs (tres de severidad *major*, uno de severidad *minor*).
 - Cuarenta y dos *code smells* (dos de severidad *major*, quince de severidad *minor*, veinticinco de severidad *info*).
- **Calificaciones:**
 - Bugs: C.
 - Vulnerabilities: A.
 - Code smells: A.

Una vez analizados los resultados obtenidos, se procedió a realizar las correcciones más convenientes para mejorar la calidad del código desarrollado.

El plan de acción seguido fue el de dar prioridad en primer lugar a los dos *bugs* y, en segundo lugar, a los *code smells*. Dentro de estos últimos, se han priorizado los de severidad *major*, posteriormente los de severidad *minor* y, en último lugar, los de severidad *info*.

Dentro del orden de prioridades citado en el párrafo anterior, se ha tenido en cuenta la proporción óptima entre el coste temporal de la resolución de cada *issue*, y el impacto en la calidad del mismo, de manera tal que algunos *code smells* de severidad *info* no han sido tratados debido al tiempo necesario para solventarlos y su bajo impacto en la calidad del código final.

A continuación, se adjunta un listado con los *issues* que, a juicio del responsable del análisis, resultaban más relevantes y, por tanto, han sido tenidos en cuenta de manera prioritaria al ejecutar acciones correctivas.

2.2. Issues y acciones correctivas efectuadas

2.2.1. Bugs

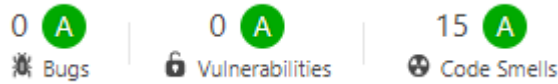
Issue	Severidad	Solución adoptada
En la clase Database.java, podía lanzarse la excepción NullPointerException (línea 305) por estar el cursor fuera de la comprobación.	Major	Se ha cambiado el cierre del cursor para que se produzca dentro de la comprobación.
En la clase ListGruposAdapter, podía lanzarse la excepción NullPointerException (línea 84).	Major	Se ha verificado mediante un <i>if</i> adicional que la variable utilizada no era nula.
En la clase ListGruposAdapter, podía lanzarse la excepción NullPointerException (línea 92).	Major	Se ha verificado mediante un <i>if</i> adicional que la variable utilizada no era nula.
En la clase ListGruposAdapter, podía suceder que el return fuese nulo, y existe anotación @NonNull.	Minor	Se ha eliminado la anotación @NonNull debido a que no era necesaria.

2.2.2. Code Smells

Issue	Severidad	Solución adoptada
En la clase Database.java, se utilizaba <i>instanceof</i> para diferenciar una excepción y su tratamiento de las demás. (línea 414).	Major	La excepción concreta se ha capturado en un catch adicional anterior al del resto de excepciones.
En las clases Database.java, DialogoEligeColorPresenter.java, GruposFragment.java, LineasFragment.java, ListGruposAdapter.java y MainActivity.java existían <i>imports</i> que no se utilizaban.	Minor	Se han eliminado todos los imports que no se utilizaban.
En la clase ListGruposPresenter, se comprueba si la lista está vacía utilizando un <i>size</i> y no un <i>isEmpty</i> (línea 44).	Minor	Se ha sustituido el <i>size</i> por <i>isEmpty</i> .
En la clase ListGruposAdapter, la legibilidad se reduce debido al empleo de switch para condicionales de tan solo dos casos.	Minor	Se han sustituido los switch con dos casos por bloques if.
En las clases Database.java, GrupoParada.java, ListGruposPresenter.java, DialogoEligeColorFragment.java se inicializaban los ArrayList sin aprovechar el operador diamante (ArrayList<Object> a=new ArrayList<Object> () en lugar de ArrayList<Object> a=new ArrayList<> ()).	Info	Se han sustituido las inicializaciones sin operador diamante por otras equivalentes empleándolo.
En las clases ListGruposAdapter.java y DialogoEligeColorFragment.java se realizaban <i>cast</i> innecesarios.	Info	Se han eliminado los casteos innecesarios.

2.3. Análisis final, *issues* pendientes y conclusiones

La corrección de todos los *issues* citados anteriormente arroja los resultados siguientes:



- Cinco horas de deuda técnica.
- Quince *code smells*, de los cuales uno es de severidad *major* y catorce son de severidad *info*.

Puede apreciarse que el número de *issues* ha descendido considerablemente con respecto a la versión inicial. Asimismo, puede remarcarse que, aunque la deuda técnica no ha sufrido un descenso acusado, sí que se han mejorado las calificaciones y reducido (casi eliminado) los errores más importantes y con mayor impacto en la mantenibilidad de la aplicación.

Los *issues* que han quedado sin tratar debido al bajo impacto en la calidad del producto y el alto esfuerzo temporal que suponen son:

Issue	Severidad	Tipo
En la clase <code>ListLineasAdapter</code> , existen dos bloques de código duplicados.	Major	Code smell
La llamada a <code>Thread.sleep(2000)</code> no debería realizarse al programar un test.	Info	Code smell
El uso de <code>progress dialog</code> se considera obsoleto (<i>deprecated</i>).	Info	Code smell

En revisiones posteriores, será necesario revisar dichos *issues*.

3. Sumario

En este documento se ha recogido el proceso seguido, los pasos analíticos y las acciones correctivas llevadas a cabo para optimizar la calidad del código asociado a la historia de usuario “Añadir parada a grupo”.