Lint Report

Grupo E7-07

Repositorio:

https://github.com/adrleogom/Acme-One

Miembros:

Abraham Cobelo Galindo; email: abrcobgal@alum.us.es
Álvaro Escalante Castro; email: alvesccas@alum.us.es
Paula Ferreira Jiménez; email: pauferjim@alum.us.es
Carmen Galván López; email: cargallop2@alum.us.es
Adriana León Gómez; email: adrleogom@alum.us.es
Beatriz Llamas Sainz-Pardo; email: beallasai@alum.us.es

25-04-2022

Tabla de Contenidos

Resumen Ejecutivo

Este informe recoge la información de cómo se ha llevado a cabo la resolución de los problemas encontrados con el SonarLint a partir de la realización de las tareas del Deliverable 3 (D03) por parte de todo el equipo.

Para ello, habrá un apartado en el que se detalle el listado de las malos olores encontrados gracias al análisis del SonarLint con la explicación de cómo se han solucionado o, si son malos olores inofensivos, justificar el porqué no se ha solucionado (si aplica).

Además, también recoge las conclusiones obtenidas a partir de la resolución de los mismos.

Control de Versiones

Versión	**Fecha**	**Descripción**
1.0	25-03-22	Creación del documento
1.1	23-04-22	Introducción y explicación de algunos bad smells encontrados
1.2	25-04-22	Finalización del documento

Introducción

En este documento se recoge la información obtenida al realizar un análisis del código de nuestro proyecto con SonarLint.

SonarLint es una extensión IDE gratuita y open-source que identifica y te ayuda a solucionar los problemas de calidad y seguridad mientras escribes el código. Es un corrector automático que te proporciona comentarios en tiempo real y una guía de corrección para que puedas entregar un código limpio (desde el principio).

En el documento se recogen los “Bad Smells” que nos hayan salido (si aplica) en el SonarLint Report de Eclipse (al haberle dado a Click derecho en el proyecto > SonarLint > Analyze) y se explica.

Al final del documento encontraremos un apartado donde podremos ver las conclusiones obtenidas y otro donde se ve reflejada la bibliografía consultada para la realización del mismo.

En el documento aparece el siguiente contenido:

Bad Smells

En total nuestro equipo ha presenciado 3 tipos de bad smells:

- El primero, ha sido porque teníamos un caracter “?” que no servía para nada en un patrón de código de un atributo de una entidad. El mal olor se ha solucionado eliminando ese caracter.

- El segundo, “Optional value should only be accessed after calling isPresent()”, ha aparecido 3 veces por intentar llamar al método .get() a un .first() sin comprobar primero que ese .first() no es nulo. Se ha solucionado definiendo una variable de tipo Optional que contenga dentro la llamada .first() y posteriormente se ha comprobado con “if,else” que esa llamada no sea nula.

- El tercero, “Maps with keys that are enum values should be replaced with EnumMap”, ha aparecido 2 veces porque utilizábamos un

Map<Status,Integer> cuando hay una clase EnumMap más específica para ello. Se ha solucionado cambiando el tipo Map a EnumMap.

Por último, también nos salen algunos bad smells de bootstrap y javascript que ya estaban contemplados por defecto, por lo que no nos incumben.
En total, nos hemos quedado con 0 bad smells.

Conclusiones

Con la realización del presente documento hemos podido detectar los bad smells que podían perjudicar la eficiencia de nuestro proyecto.

Además, gracias a las explicaciones de SonarLint nos hemos dado cuenta de bad smells potencialmente peligrosos que a simple vista no veríamos y los hemos podido arreglar satisfactoriamente mediante la colaboración de todo el equipo.

Bibliografía

https://www.excentia.es/que-es-sonarlint-y-como-mejora-calidad-codigo#:~:text=SonarLint%20es%20una%20extensión%20IDE,seguridad%20mientras%20escribes%20el%20código.