DP2 2024

Acme Software Factory

Repositorio: https://github.com/DP2-2024-C1-029/Acme-Software-Factory.git

Miembro:

* José María Portela Huerta (josporhue@alum.us.es)

Tutor: José María Portela Huerta

GRUPO C1.029

26/04/2024

Versión 1.0

Índice

[Historial de versiones 3](#_Toc165029355)

[Introducción 4](#_Toc165029356)

[Contenido 4](#_Toc165029357)

[Conclusiones 6](#_Toc165029358)

[Bibliografía 7](#_Toc165029359)

# Historial de versiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Entrega |
| 26/04/2024 | V1.0 | Inicio documento D03 | D03 |
|  |  |  |  |
|  |  |  |  |

# Introducción

A continuación, se procede a enumerar los code smells detectado por sonar lint

# Contenido

Enumeramos los comentarios dados tras el reporte de sonarlint:

Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza bajaTabla

Descripción generada automáticamenteImagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Se han detectado notificaciones de "assert object != null;" en múltiples archivos del código heredado. Estas notificaciones se mantienen como parte del formato original del código y no se modificarán.

Se ha decidido conservar el formato original del código heredado, incluyendo las notificaciones, para mantener la integridad del mismo y facilitar su comprensión por parte de aquellos familiarizados con el estilo de codificación utilizado por los profesores.

Debido a la presencia de estas notificaciones en todos los archivos, no se adjuntarán imágenes individuales para cada caso. Se considera que la descripción proporcionada es suficiente para identificar y comprender el problema.

Es importante tener en cuenta que la instrucción "assert object != null;" se utiliza para verificar que un objeto no sea nulo antes de utilizarlo. Si bien esta práctica puede ser útil para detectar errores de programación, se recomienda utilizar métodos de comprobación de nulos más robustos en el desarrollo de nuevo código.



Este método nos dice que agreguemos la anotacion @Override, sin embargo, no se ha dejado explícito en la clase porque no ha sido necesario en el contexto del proyecto.

Un conjunto de letras negras en un fondo blanco

Descripción generada automáticamente con confianza mediaTabla

Descripción generada automáticamente

Esto nos indicas que “hay código duplicado”. Simplemente es como se repiten a la hora de construir los métodos bind y unbind los string los nombres salta la notificación. No se ha hecho una variable, puesto que se quiere seguir la lógica y dejarlo todo como cadenas de texto





Esa recomendación la marca para cambiar la forma en la que creo una expresión, sin embargo la que hay ahora mismo es funcional y es con la que mejor me manejo

También se ha notificado líneas comentadas que estaban duplicadas, y variables que no se usaban. Esas notificaciones han sido corregida, eliminando todo lo que estuviera duplicado y no estuviera en uso.

# Conclusiones

Se ha utilizado la herramienta SonarLint para identificar y eliminar code smells en el proyecto, principalmente variables inutilizadas y código comentado que no se estaba utilizando.

Gracias a la utilización de SonarLint, el proyecto se encuentra ahora en un estado más limpio, entendible y mantenible. Esto contribuye a mejorar la calidad general del código y a facilitar su desarrollo y mantenimiento en el futuro.

# Bibliografía

Diapositivas de la asignatura Diseño y Pruebas 2 – Universidad de Sevilla.