

Lint Report



Grado en Ingeniería Informática - Ingeniería del Software

Diseño y Pruebas II

Curso 2023 - 2024

Código de Grupo: C1-001		
Autor	Correo	Rol
Manuel Vélez López	manvellop2@alum.us.es	Desarrollador

Repositorio: <https://github.com/DP2-2023-2024-C1-001/Acme-SF-D03>

Índice de Contenidos

- 1. Resumen Ejecutivo..... 3
- 2. Control de Versiones..... 4
- 3. Introducción..... 5
- 4. Contenido..... 6
- 5. Conclusión..... 7
- 6. Bibliografía..... 8

1. Resumen Ejecutivo

Objetivo:

El propósito de este informe de análisis Lint realizado a través de Eclipse es ofrecer una evaluación exhaustiva del código fuente del proyecto, identificando áreas críticas para mejorar la calidad del código y minimizar posibles problemas técnicos.

Roles y Responsabilidades:

Desarrollador: Responsable de escribir código, realizar pruebas informales y mantener altos estándares de calidad en el desarrollo del proyecto.

Todos los Roles: Tienen la responsabilidad compartida de escribir informes sobre el trabajo realizado y colaborar en la mejora continua del proceso de desarrollo.

Plan de Acción:

Configuración del Proyecto: Los desarrolladores establecen la configuración del proyecto en Eclipse, asegurando que se sigan las mejores prácticas de codificación y se apliquen las reglas de linting adecuadas.

Análisis del Código: Se ejecuta el análisis Lint en el código fuente del proyecto utilizando las herramientas integradas en Eclipse.

Identificación y Corrección de Problemas: Se identifican posibles violaciones de estándares de codificación, errores o malas prácticas en el código, y se toman medidas correctivas para abordarlos.

Commit y Push: Una vez que se han corregido los problemas, se realiza un commit y push al repositorio remoto para mantener un registro de los cambios realizados.

Revisión del Código: Se crea una solicitud de revisión de código (pull request) y se asigna un revisor para validar los cambios realizados.

Aprobación y Merge: Después de que el revisor aprueba los cambios, se realiza el merge del código corregido con la rama principal del repositorio.

2.Control de Versiones

Fecha	Versión	Descripción
26/04/2024	V1.0	Creación del documento entrega D03.

3.Introducción

Este informe de análisis estático se enfoca en una evaluación exhaustiva de los resultados obtenidos a través de SonarLint en nuestro proyecto. Como parte esencial de este proceso, hemos examinado detalladamente cada uno de los "puntos de mejora" reportados con el objetivo de identificar áreas que requieran atención y tomar decisiones fundamentadas para su corrección o justificación.

En resumen, este informe de Lint se centra en la evaluación minuciosa de los problemas de código identificados, proporcionando recomendaciones y soluciones para su optimización. A través de una estructura clara y sistemática, buscamos facilitar la toma de decisiones informadas y promover el progreso del proyecto en cuestión.

El documento comienza con un detallado listado de los "puntos de mejora" detectados por SonarLint en nuestro proyecto, describiendo cada uno de ellos y explicando su naturaleza. Luego, se presentan argumentos sólidos que respaldan la conclusión de que estos problemas son benignos en el contexto específico del proyecto. Esta estructura permite una comprensión completa de los desafíos identificados por SonarLint y las decisiones adoptadas para abordarlos o justificar su no intervención.

4. Contenido

Listado de Bad Smells:

- **Define a constant instead of duplicating this literal "difficultLevel" 3 times:** Este bad smell nos indica que el string difficultLevel lo hemos repetido varias veces lo cual no sería lo más efectivo, por lo que tendríamos que definir una variable con este valor. Este bad smell se repite en las clases:
DeveloperTrainingSessionCreateService,
DeveloperTrainingModuleCreateService,
DeveloperTrainingModulePublishService,
DeveloperTrainingSessionPublishService
- **Replace this assert with a proper check:** Este indicador de "mal olor" sugiere que la verificación que llevamos a cabo en el método unbind puede no ser lo suficientemente exhaustiva en cuanto al objeto recibido por parámetros. Sin embargo, esta situación no representa un problema real, dado que todas las validaciones necesarias para cumplir con cada requisito se ejecutan de manera adecuada en el método validate. Este bad smell se repite en las clases: **DeveloperTrainingModulePublishService,**
DeveloperTrainingModuleShowService,
DeveloperTrainingModuleCreateService,
DeveloperTrainingModuleUpdateService,
DeveloperTrainingModuleDeleteService,
DeveloperTrainingModulePublishService

5. Conclusión

En resumen, este informe destaca la utilidad de SonarLint para identificar y resolver problemas de calidad en nuestro proyecto. Aunque encontramos algunos "puntos de mejora", determinamos que no requerían cambios inmediatos. Nuestro enfoque ha sido reflexivo, estableciendo un plan de acción para futuras correcciones si fuesen necesarias. Esta práctica nos permite mejorar continuamente la calidad del código y fortalecer nuestro proceso de desarrollo, beneficiando a todos los involucrados.

6. Bibliografía

Intencionadamente en blanco.