

LINT REPORT D03

Repository: <https://github.com/alepingar/Acme-Sf-C2.043>

- Miembros:

Alvmarmun1@alum.us.es Alvaro Martín Muñoz

alepingar@alum.us.es Alexander Picón Garrote

Índice	de	Contenidos	1
Resumen			3
Tabla	de	Revisión	4
Introducción			5
Contenido			6
Requisitos		obligatorios	6
Requisitos		funcionales	6
Requisitos	no	funcionales	7
Requisitos		suplementarios	8
Requisitos		funcionales	8
Requisitos	no	funcionales	10
Requisitos	de	gestión	11
Conclusiones			12
Bibliografía			13

Tabla de revisión

Fecha	Version	Descripción	Sprint
25/05/2024	1.0	<ul style="list-style-type: none">• Contenido del documento	4

Introducción

En estas dos fases de los entregables, el enfoque se centró en la implementación de diferentes características, lo que creó múltiples "malos olores" en el proceso. El análisis subsiguiente en este documento se refiere únicamente a la evaluación de lint del trabajo colectivo del grupo. Se realizó mediante el análisis de todos los archivos modificados durante la duración de este entregable, tanto para los requisitos obligatorios como los suplementarios. El análisis de lint para los estudiantes individuales se puede consultar en sus respectivas carpetas.

La estructura del documento se describe de la siguiente manera: una sección introductoria que contextualiza el análisis de lint, seguida de una enumeración exhaustiva de los "malos olores" detectados por Sonar Lint dentro del trabajo grupal. Finalmente, una conclusión sucinta encapsulará el informe, resumiendo los hallazgos clave.

Contenido

Malos olores

Los siguientes malos olores aparecieron varias veces en los archivos individuales analizados para este informe. Por simplicidad, se han consolidado en una enumeración única.

1. Sobrescribir el método "equals" en esta clase. Seguimos las recomendaciones del Framework Acme dadas en las clases teóricas, por lo que este mal olor puede considerarse inocuo.
2. Usar la sintaxis concisa de la clase de caracteres '\d'. Este mal olor es inocuo, ya que realiza las mismas funcionalidades que la opción alternativa propuesta y no dificulta la mantenibilidad del código. El cambio es simplemente más conciso.
3. Renombrar este nombre de paquete para que coincida con la expresión regular '^([a-z_]+)([a-z_0-9_])\$'. Siguiendo las recomendaciones de estructura dadas por el ejemplo del proyecto Acme Jobs, este mal olor puede considerarse inocuo.
4. Reemplazar este assert con una verificación adecuada. Siguiendo las recomendaciones del Framework Acme dadas en las clases teóricas, este mal olor puede considerarse inocuo.
5. Definir una constante en lugar de duplicar este literal "X" veces. Este mal olor es inocuo por naturaleza, ya que la duplicación de literales no afecta directamente la implementación de las características. Podría plantear un problema de refactorización potencial, pero dado que las recomendaciones y ejemplos dados por la asignatura también duplican literales, podemos ignorarlo con seguridad.
6. Completar la tarea asociada a este comentario TODO. Este mal olor es inocuo, aunque es importante hacer un seguimiento de los comentarios TODO faltantes en el código para evitar confusiones en el desarrollo de las características.
7. Capturar Exception en lugar de Throwable. Exception extiende Throwable, por lo tanto, es equivalente. A pesar de esto, nuevamente, Acme Jobs sigue esta estructura, por lo que podemos considerarlo inocuo.
8. Declarar "variable" en una línea separada. Este mal olor es inherentemente inocuo. La única razón por la que esto es considerado un mal olor es porque se considera "difícil de leer", lo cual es discutible.
9. Refactorizar esta repetición que puede llevar a un desbordamiento de pila para entradas grandes. Aunque esto puede considerarse un mal olor, es necesario implementar la repetición en este patrón regex.

Conclusiones

En conclusión, este documento del informe de lint nos permite examinar los obstáculos enfrentados al analizar e implementar diversos requisitos, así como revisar nuestro código para evitar problemas futuros de mantenibilidad.

Bibliografía

En blanco intencionalmente.