

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

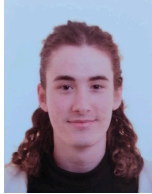
DP2-Reporte de Linting Student 4 D3



Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas II

Curso 2023 – 2024

<u>Group:</u>	C2.X02
<u>Repository:</u>	https://github.com/SoniaRM/Acme-SF-D04-24.5.0
<u>Student #4</u>  UVUS: josmirmar2 Name: José Manuel Miret Martín Email: josmirmar2@alum.us.es	
<u>Date:</u>	Sevilla Julio 2, 2024

Índice de contenido

1. Versiones	2
2. Lista de bad smells	2
3. Conclusión	2
5. Bibliografía	2

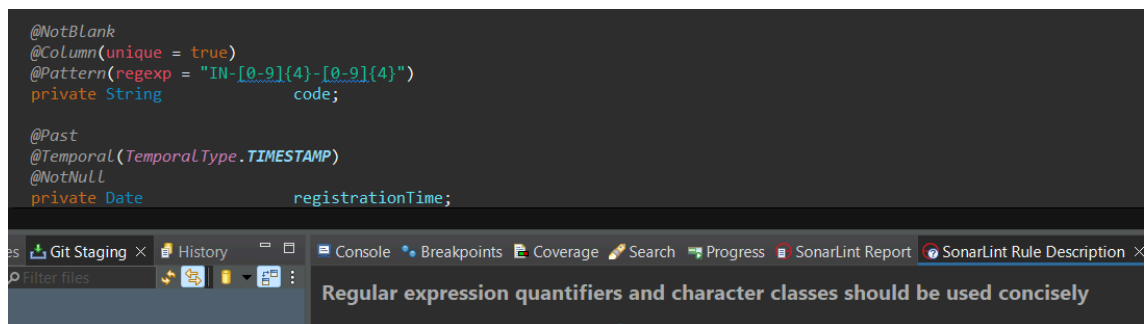
1. Versiones

Versión	Fecha	Autor
1.0	23/04/2024	José Manuel Miret Martín
1.1	02/07/2024	José Manuel Miret Martín

2. Lista de bad smells

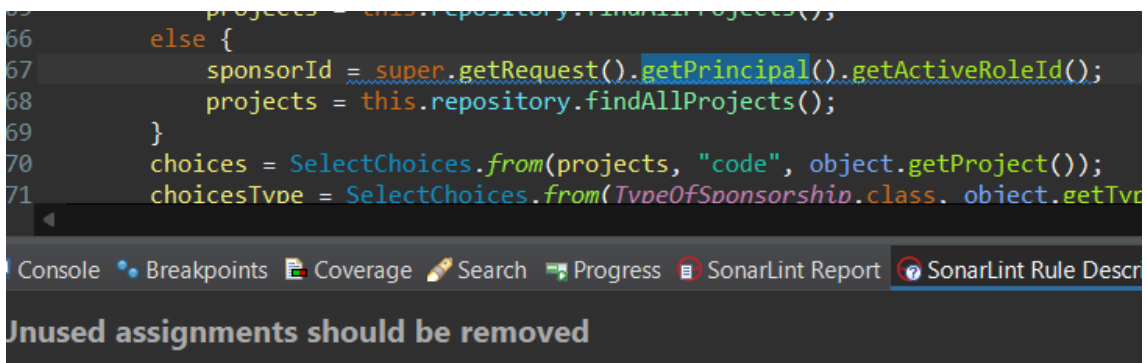
- Poner `\d` en lugar de `[0-9]` en expresiones regulares:

Su descripción es 'Regular expression quantifiers and character classes should be used concisely'. En lugar de utilizar la expresión regular `[0-9]` para representar cualquier dígito del 0 al 9, sería preferible usar `\d` para indicar dígitos numéricos en general. Sin embargo, por motivos de claridad en la expresión regular y porque esta es la que se especificó en los requisitos, he decidido mantenerla tal como está. Esta elección no afecta el funcionamiento del código y, además, contribuye a hacer la expresión más clara y fácil de entender.



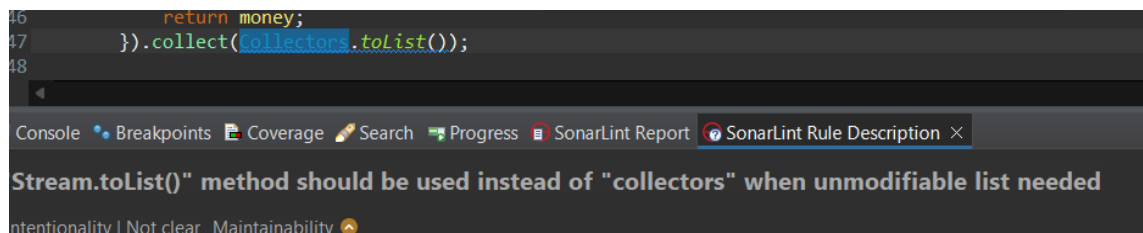
- Eliminar código inservible:

Aparece 'Unused assignments should be removed'. Esto si lo considero un bad smell, ya que si cambia alguna parte del código de tal forma que ese código que no se utiliza da algún error en un futuro, puede dar problemas. Es por ello que si considero eliminar ese código y por lo tanto solucionar ese bad smell.



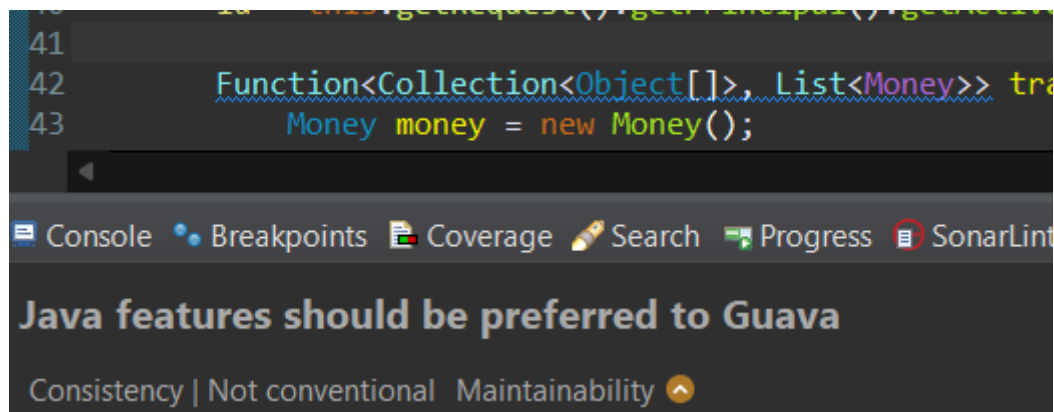
- Utilizar collector para elaborar una lista no mutable:

“Stream.toList()” method should be used instead of “collectors” when unmodifiable list needed’. No creo que se pueda considerar un bad smell, ya que es cambiar el uso de una función que use collectors a otra que usa Stream.toList. Es cierto que si esa lista no es mutable, es mejor el método que me sugiere. Sin embargo, al final la propiedad que obtengo al usar esa función no veo necesario que sea no mutable, ya que no voy a modificar las propiedades que hay dentro más adelante en el código. Es por ello que no creo que pueda considerarse un problema para un futuro y no veo necesario elaborar un cambio.



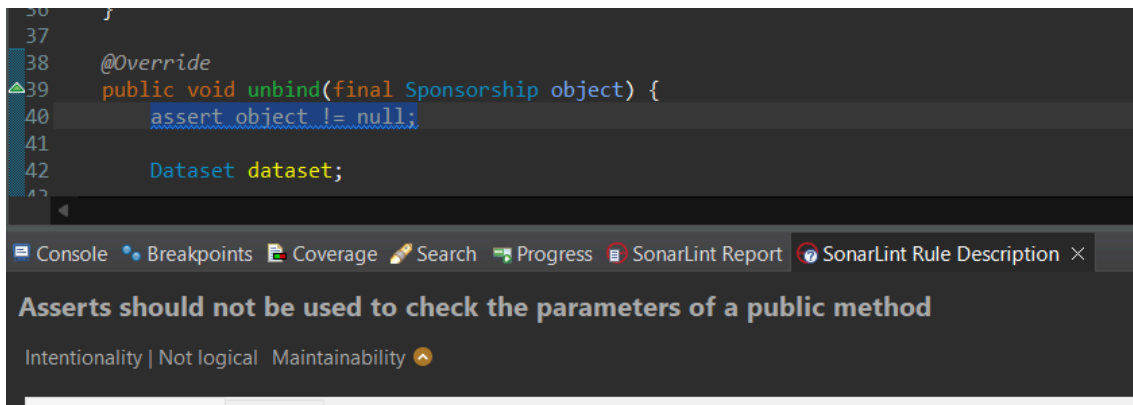
- Utilizar Guava en vez de Java:

La descripción del bad smell es: ‘Java features should be preferred to Guava’. No creo que se pueda considerar un bad smell, ya que estoy utilizando las características proporcionadas por Java en lugar de la biblioteca Guava, sin tener en cuenta lo útil que puede ser. Es por ello que no creo que pueda considerarse un problema para un futuro.



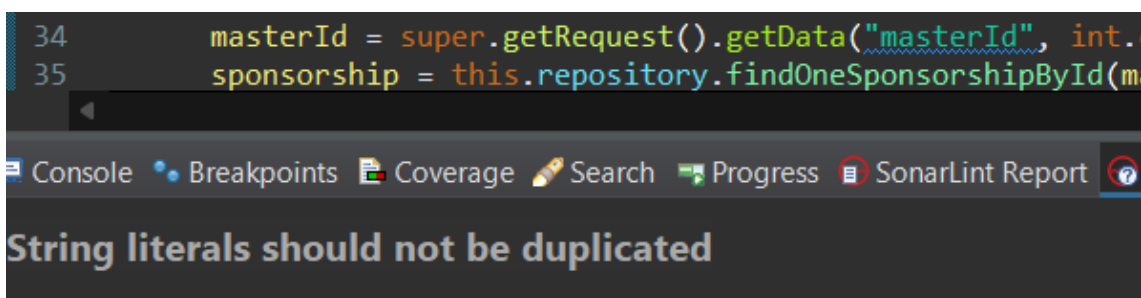
- Evitar usar assert:

‘Replace this assert with a proper check’. Este problema de calidad de código no es crítico porque el código sigue asegurando que el objeto no sea nulo, cumpliendo así con su propósito. Por esta razón, he decidido no abordarlo. Además, esta instrucción es parte de las pautas recomendadas para implementar servicios utilizando el framework que estamos utilizando, por lo que no planeo modificarla.



- Crear constantes genéricas en lugar de repetir la misma variable en varios sitios:

Este code smell tiene la siguiente descripción: ‘String literals should not be duplicated’. Este problema también se repite en varios archivos, no solo con masterId, también con dueDate o quantity. Aunque este problema de calidad de código no es crítico, ya que sigue cumpliendo su propósito sin afectar la comprensión ni el funcionamiento del código, quiero mencionar que estas estructuras fueron recomendadas en la capacitación de servicios que recibimos. Por lo tanto, he seguido esta estructura en mi código y por lo tanto no tengo pensado cambiarlo.



3. Conclusión

En resumen, aunque existan algunos "bad smells" en el código, podemos concluir que este código no es de baja calidad y funciona correctamente. Es fácil de entender y leer, lo que es fundamental para su mantenimiento y evolución.

4. Bibliografía

No hay bibliografía presente para esta entrega. (intentionally blank)