Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática

SONAR LINT REPORT INDIVIDUAL - D03



**C1.027**

**Repositorio**: <https://github.com/josrojrom1/DP2-G27-Acme>

**Autor:**

Devós Bono, Agustín

agudevbon@alum.us.es

**Fecha:** 03/07/2024

**Tabla de contenidos**

[**Resumen**](#_heading=h.gjdgxs) **2**

[**Tabla de revisión**](#_heading=h.30j0zll) **2**

[**Introducción**](#_heading=h.1fob9te) **2**

[**Contenidos**](#_heading=h.3znysh7) **3**

[**Bibliografía**](#_heading=h.3dy6vkm) **3**

# Resumen

Este documento se trata de un listado de registros de malos olores con la herramienta Sonar Lint. En caso de detectar malos olores, se decidirá su corrección o no en función de si es importante o no el registro en cuestión para el correcto funcionamiento del código y su legibilidad.

# Tabla de revisión

|  |  |  |
| --- | --- | --- |
| **Versión** | **Fecha** | **Descripción** |
| V1 | 03/07/2024 | Creación y desarrollo del documento |

# Introducción

Con motivo de detectar malos olores, y su posible corrección, se han aportado capturas del análisis ejecutado con Sonar Lint y se ha realizado un análisis de cada captura con la decisión al respecto.

# Malos olores detectados tras análisis y valoración

Se ha ejecutado un análisis del proyecto individual completo, obteniendo los siguientes malos olores, los cuales se mostrarán mediante capturas del propio log generado por la herramienta:



**Captura 1**. AnyProject List and Show Service bad smells

En la **Captura 1** observamos unicamente dos malos olores relacionados con el uso de ***asserts*** en los métodos de los servicios.

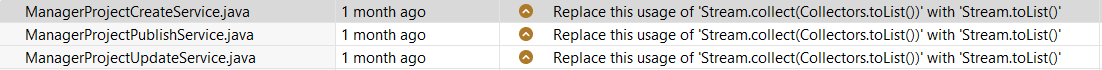
**Valoración**: Los malos olores que nos recomiendan reemplazar los ***asserts*** por ***checks*** no se tendrán en cuenta, pues en el framework que empleamos es el modo en el que se comprueban todos los objetos usados en los respectivos métodos son distintos de ***null*** y no se emplean para nada más. Para evitar repetir valoraciones sobre malos olores de forma repetitiva, esta valoración también se tendrá en cuenta para los servicios de Create, Delete, Update, List y Show de las entidades Project, UserStory y ProjectUserStory, clases en las cuales encontramos este mismo mal olor.



**Captura 2**. Constants suggestion in *authorise, bind* and *unbind* methods from services

En la **Captura 2** observamos un aviso en rojo que nos sugiere definir constantes para los parámetros que se pasan en los métodos *authorise*, *bind* y *unbind* de los servicios ManagerProjectPublishService, ManagerProjectUserStoryCreateService, ManagerUserStoryCreateByProjectService y ManagerUserStoryListByProjectService, por lo que solo se mostrará esta captura para representar todas las apariciones de dicha advertencia para mejorar la legibilidad del documento.

**Valoración:** En el diseño de los métodos que reciben parámetros como masterId, ya sea para extraerlos de la solicitud o para pasarlos a otro método que los requiera (como los atributos de una entidad que se pasan a una vista o se devuelven en un rollback), se espera que dichos parámetros se proporcionen en formato de cadena, es decir, entre comillas dobles. Si creáramos una constante, podríamos introducir una posible vulnerabilidad al tener que controlar perfectamente su uso. Por estas razones, y debido a que se mencionó en clase que no es necesario declarar constantes en estos casos, se han ignorado los code smells relacionados con el uso de constantes.



**Captura 3**. Replace Collectors.toList() with Stream.toList()

En la **Captura 3** observamos un aviso en naranja que nos sugiere reemplazar el uso del método *Collectors.toList()* a Stream.toList().

**Valoración:** Se ha probado a aplicar la sugerencia sin obtener ningún error por lo que se ha aplicado dicho cambio para mejorar la calidad del código.



**Captura 4**. Regular expression naming bad smell

En la **Captura 4** observamos varios avisos idénticos relacionados con el seguimiento de una expresión regular a la hora de nombrar las clases de cualquier rol principal sobre proyectos e historias de usuario. Estos avisos se pueden encontrar repetidas veces a lo largo del proyecto individual.

**Valoración:** Para nombrar las clases se han seguido las recomendaciones de los profesores, donde esta empieza por el Rol, para el que se desarrolla dicha clase, comenzando por mayúsculas. Es por esto por lo que no se tendrán en consideración dichos malos olores.



**Captura 5**. Replace negated “anyMatch()” with “noneMatch()”

En la **Captura 5** observamos un aviso en naranja que nos sugiere reemplazar el uso de la expresión anyMatch() negada con la expresión noneMatch().

**Valoración:** Se ha probado a aplicar la sugerencia sin obtener ningún error por lo que se ha aplicado dicho cambio para mejorar la calidad del código.

Para realizar el análisis se ha tenido en cuenta las respuestas en el foro realizadas acerca de los malos olores y el uso de Sonar Lint. Para ver los detalles puede visitar el siguiente enlace al foro con más información: <https://ev.us.es/webapps/discussionboard/do/message?action=list_messages&course_id=_85092_1&nav=discussion_board&conf_id=_405265_1&forum_id=_234042_1&message_id=_415711_1>

# Bibliografía

-Documento 08 Annexes, de los contenidos de la plataforma virtual de esta asignatura.