Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática



Grado en Ingeniería Informática – Ingeniería del Software  
Diseño y Pruebas II

Curso 2023 – 2024

**Informe Lint**

**Grupo:** C1.014

**Repositorio**: <https://github.com/Enrgaraba/Acme-SF-D03>

**Fecha**: 24/04/2024

|  |  |
| --- | --- |
| **Integrante del grupo** | **Correo** |
| Chico Castellano, Álvaro | alvchicas@alum.us.es |
| del Castillo Piñero, Daniel | dandelpin@alum.us.es |
| García Abadía, Enrique | enrgaraba@alum.us.es |
| Linares Barrera, Jaime | jailinbar@alum.us.es |
| Pérez Fernández, Ibai | ibaperfer@alum.us.es |

**Historial de versiones**

|  |  |  |
| --- | --- | --- |
| **Versión** | **Fecha** | **Descripción de los cambios** |
| V1.0 | 24/04/2024 | Creación del informe |
|  |  |  |

**Índice de contenido**

[**1.** **Resumen ejecutivo** 2](#_Toc165057078)

[**2.** **Introducción** 2](#_Toc165057079)

[**3.** **Contenido** 2](#_Toc165057080)

[**3.1.** **Listado de *bad smells* y la justificación de por qué son inofensivos** 2](#_Toc165057081)

[**4.** **Conclusiones** 4](#_Toc165057082)

[**Bibliografía** 4](#_Toc165057083)

# **Resumen ejecutivo**

Este informe Lint presenta una evaluación exhaustiva de los posibles problemas identificados por Sonar Lint en el proyecto. Este informe incluye una lista detallada de los *bad smells* informados por Sonar Lint y una justificación clara de por qué se consideran inofensivos. Si Sonar Lint está reportando un *bad smell* inofensivo, proporcionando así una justificación clara en el informe.

# **Introducción**

El presente informe de Lint ofrece una evaluación detallada de los posibles problemas identificados por Sonar Lint en el proyecto actual. Sonar Lint es una herramienta ampliamente utilizada para detectar problemas de calidad de código, como lo que comúnmente se denominan *bad smells* o *code smells*. En este informe, se proporcionará una lista exhaustiva de los *bad smells* reportados por Sonar Lint, junto con una justificación clara de por qué se consideran inofensivos en el contexto del proyecto.

Es importante destacar que, según las directrices del informe, los *bad smells* identificados en el marco de Acme pueden ser ignorados de manera segura. Sin embargo, en casos en los que se considere que Sonar Lint está reportando un *bad smell* inofensivo, hay que proporcionar una justificación clara en el informe. Esta práctica garantiza una comprensión precisa de la calidad del código y facilita la corrección de cualquier problema identificado de manera eficiente.

# **Contenido**

# **Listado de *bad smells* y la justificación de por qué son inofensivos**

Replace this assert with a proper check:

**Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media**

Es un *bad smell* inofensivo ya que en este caso lo único que queremos comprobar es que no sea nulo el objeto que recibimos por parámetro. Aunque es cierto que podría cambiarse por un mensaje de error que sea más conciso, pero en principio con esa validación es más que suficiente para las tareas que nos incumben en las funciones en la que aparece.

Define a constant instead of duplicating this literal "XXXXX" Y times:

**Texto

Descripción generada automáticamente con confianza baja**

Este es el *bad smell* más extendido por el código del proyecto. En la mayoría de los casos esto suele ser un *bad smell* que sí que hay que solucionar, por ejemplo, cuando es una cadena para indicar el código de un error. En cambio, en este caso vemos fundamental dejar el nombre tal y como es ya que es como aparece tanto en las entidades y como se recoge de la pantalla que le aparece al cliente y creemos que es importante mantenerlo para no caer en confusiones a la hora de trabajar con el código y poder generar problemas innecesarios.

# **Conclusiones**

En conclusión, este informe Lint proporciona una evaluación detallada de los posibles problemas identificados por Sonar Lint en el proyecto, centrándose en los llamados "bad smells" o "code smells". Se presenta una lista exhaustiva de estos "bad smells", junto con una justificación clara de por qué se consideran inofensivos en el contexto del proyecto. En nuestro caso los "bad smells" mas extendidos a lo largo del código son los referidos a propiedades duplicadas, que debidamente hemos justificado su existencia y su inofensividad.

Se destaca la importancia de seguir las directrices del informe, especialmente en lo que respecta a los "bad smells" identificados en el marco de Acme, que pueden ser ignorados de manera segura. Sin embargo, en aquellos casos en los que se considere que Sonar Lint está reportando un "bad smell" inofensivo, se enfatiza la necesidad de proporcionar una justificación clara en el informe. Esta práctica garantiza una comprensión precisa de la calidad del código y facilita la corrección de cualquier problema identificado de manera eficiente.

Además, se subraya la importancia de mantener un enfoque práctico y centrado en la eficiencia al abordar los problemas identificados por Sonar Lint. Se reconoce que no todos los "bad smells" requieren una intervención inmediata y que algunas soluciones pueden considerarse innecesarias o incluso contraproducentes en determinados contextos de desarrollo.

En resumen, este informe proporciona una guía valiosa para la gestión de los problemas de calidad de código identificados por Sonar Lint, asegurando una evaluación precisa y una respuesta adecuada en el contexto del proyecto.

# **Bibliografía**

Intencionalmente en blanco.