

Grupo: C1.015

Repositorio: <https://github.com/jormunrod/Acme-SF-24.1.1>

# Lint Report

Isaac Solís Padilla ([isasolpad@alum.us.es](mailto:isasolpad@alum.us.es))

Sevilla, 27-5-2024

Contenido

1. Resumen ejecutivo ..... 2

2. Tabla de revisión ..... 3

3. Introducción..... 4

4. Contenidos..... 5

5. Conclusiones..... 6

6. Bibliografía ..... 7

## 1. Resumen ejecutivo

Este reporte presenta los resultados de la evaluación de calidad del código utilizando SonarLint para identificar y analizar "bad smells" en el proyecto. Se han listado todos los problemas detectados, excluyendo aquellos encontrados en el Acme Framework según las instrucciones recibidas.

Para cada "bad smell" identificado, se ha proporcionado una justificación detallada explicando por qué se considera inocuo dentro del contexto del proyecto o se ha descrito cómo fue corregido.

El objetivo de este reporte es asegurar que el código no solo cumpla con los estándares de calidad, sino que también mantenga su funcionalidad y eficiencia, minimizando las interrupciones innecesarias y mejorando la mantenibilidad del software.

## 2. Tabla de revisión

<b>Nº Revisión</b>	<b>Fecha</b>	<b>Descripción</b>
1	27-05-2024	Versión inicial
2	27-05-2024	Versión final

### 3. Introducción

El presente Lint Report tiene como objetivo analizar y evaluar la calidad del código del proyecto actual mediante el uso de SonarLint, una herramienta de análisis estático que ayuda a identificar problemas potenciales, conocidos como "bad smells", en el código fuente.

Este reporte detalla los resultados de dicho análisis, proporcionando una lista exhaustiva de los bad smells detectados, junto con justificaciones claras sobre su relevancia y las medidas correctivas adoptadas.

Además, se excluyen deliberadamente los hallazgos relacionados con el Acme Framework, conforme a las directrices proporcionadas. Este análisis permite no solo mejorar la calidad del código, sino también asegurar su alineación con las mejores prácticas de desarrollo y mantenimiento de software

## 4. Contenidos

A continuación, se presenta un resumen detallado de los "bad smells" detectados en el código del proyecto utilizando la herramienta SonarLint. Este reporte incluye tanto las incidencias repetidas como aquellas únicas, detallando la naturaleza del problema y las acciones tomadas en respuesta, ya sea mediante correcciones aplicadas directamente en el código o justificaciones claras por las cuales ciertos hallazgos pueden ser considerados inocuos. La tabla busca proporcionar una visión clara y estructurada para facilitar la revisión y la toma de decisiones futuras respecto a la calidad del código

Archivo	Bad Smell	Justificación/Corrección
<b>AnyTrainingModuleController</b>	Rename this package name to match the regular expression.	No es posible renombrar el paquete, ya que es necesario su formato para el correcto funcionamiento del framework.
<b>AnyTrainingModuleListService</b>	Rename this package name to match the regular expression.	No es posible renombrar el paquete, ya que es necesario su formato para el correcto funcionamiento del framework.
<b>AnyTrainingModuleRepository</b>	Rename this package name to match the regular expression.	No es posible renombrar el paquete, ya que es necesario su formato para el correcto funcionamiento del framework.
<b>AnyTrainingModuleShowService</b>	Rename this package name to match the regular expression.	No es posible renombrar el paquete, ya que es necesario su formato para el correcto funcionamiento del framework.
<b>AnyTrainingModuleShowService</b>	Replace this assert with a proper check.	No es un problema en el contexto del framework de la aplicación.
<b>AnyTrainingModuleListService</b>	Replace this assert with a proper check.	No es un problema en el contexto del framework de la aplicación.

-

## 5. Conclusiones

El análisis realizado con la herramienta SonarLint ha permitido identificar una serie de "bad smells" en el código del proyecto. A pesar de las múltiples incidencias detectadas, todas ellas se han determinado como no problemáticas dentro del contexto específico del framework utilizado en la aplicación. En casos donde no fue posible implementar cambios sugeridos por SonarLint, como la renombración de paquetes, se debió a requerimientos estructurales del framework que son esenciales para su funcionamiento.

Este reporte no solo mejora la comprensión de la calidad actual del código, sino que también guía futuras intervenciones y mantenimientos, asegurando que las decisiones tomadas estén bien informadas y sean adecuadas para el contexto del proyecto. La identificación de estos problemas y la acción correspondiente o justificación de su inocuidad contribuyen significativamente a la integridad y robustez del software desarrollado.

## 6. Bibliografía

Intencionalmente en blanco.