INFORME DE LINT GRUPAL

Grupo 1-C1.020 | https://github.com/PDJ6975/Acme-ANS-D04-25.5.0

Nombre	Correo Corporativo
Antonio Rodríguez Calderón	antrodcal@alum.us.es
Adrián Ramírez Gil	adrramgil@alum.us.es
Jianwu Hu	jiahu3@alum.us.es
Pablo Castrillón Mora	pabcasmor1@alum.us.es
Pablo Olivencia Moreno	pabolimor@alum.us.es

26 DE MAYO DE 2025

DISEÑO Y PRUEBAS II

Tabla de Contenido

1. Introducción

Este informe presenta los resultados del análisis estático de código (Lint) aplicado al proyecto Acme-ANS-25.5.0. El propósito de dicho análisis es detectar posibles inconvenientes en el código fuente, como malos olores de código, errores potenciales o patrones de diseño que podrían optimizarse, con el objetivo de elevar la calidad del software y facilitar su mantenimiento a largo plazo.

2. Metodología

El análisis se llevó a cabo utilizando el plugin integrado de Eclipse para la detección de problemas estáticos. Esta herramienta examina diversos aspectos del código, como declaraciones innecesarias, métodos no utilizados, estructuras de control redundantes y el cumplimiento de convencoiens de codificación.

Para cada caso identificado como ‘code smell’, se evaluará si su presencia es inofensiva y no afecta al comportamiento del sistema, o si, por el contrario, requiere corrección antes de ejecutar un nuevo análisis.

3. Resumen global

Durante el análisis con SonarLint se han detectaron un total de 6 tipos distintos de code smells distribuidos en los distintos archivos de los requisitos grupales. Estos code smells corresponden principalmente a recomendaciones de buenas prácticas generales, pero en su mayoría no representan riesgos funcionales y han sido evaluados como inocuos dentro del contexto académico en el que se desarrolla este software.

4. Listado de malos olores

Regla	Descripción	Afecta a	Tipo	Justificación
Java:S6813	Field dependency injection should be avoided	Controladores y servicios con @Autowired	Inocuo	La inyección de dependencias se realiza tal y como se especifica en la asignatura
Java:S1192	Define a constant instead of duplicating this literal "xxxxxxxx" n times.	Servicios donde se repiten atributos de entidad como model, details, iataCode, etc.	Inocuo	Los literales se usan directamente por simplicidad y claridad en la lógica, como se permite en la asignatura.
Java:S1186	Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation.	Servicios donde el método bind() no es necesario para la funcionalidad específica	Inocuo	El método bind() es parte de una estructura común, pero en varios servicios no se necesita implementar, y dejarlo vacío es suficiente.
Java:S106	Replace this use of System.out by a logger.	AdministratorDashboardShowService	Inocuo	Se usa System.out únicamente con fines de depuración en desarrollo y en

				el contexto de prácticas, sin necesidad de introducir un sistema de logging completo.
Java:S4274	Replace this assert with a proper check.	AdminMaintenanceRecordShowService	Inocuo	No es crítico para la ejecución del sistema y ayuda a detectar errores durante el desarrollo.
Java:S2160	Override the equals method in this class.	Todas las entidades	Inocuo	La creación de las clases se realiza tal y como se especifica en la asignatura.

5. Tabla de revisiones

Versión	Fecha	Descripción de los cambios
1.0	26/05/2025	Primera versión del documento