

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

LINT REPORT



Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas II

Curso 2024 – 2025

Grupo C2.021 - Student 4		
Autores	Correos	Rol
Paula Sánchez Gómez	pausangom1@alum.us.es	Desarrollador/Tester/Manager

Repositorio: <https://github.com/DP2-C1-021/Acme-ANS-C2>

30 de Junio de 2025

Control de versiones

Versión	Fecha	Descripción
v1.0	30/06/2025	Primera versión del documento

Índice

Introducción	3
Bad smells detectados y corregidos	3
1. Inyección por campo (@Autowired)	3
2. Duplicación de literales ("resolution", "indicator", etc.)	3
3. Complejidad cognitiva excesiva en el método validate(...)	3
4. Métodos vacíos sin comentario	4
5. Repetición innecesaria de expresiones (collection.size() > 0)	4
6. Uso innecesario de Collectors.toList() seguido de .stream()	4
Conclusión	4

Introducción

Como Estudiante 4, he realizado el análisis estático del código fuente del proyecto utilizando **SonarLint**. El objetivo ha sido identificar y corregir todas las **malas prácticas** (*bad smells*) reportadas por la herramienta dentro del desarrollo realizado, excluyendo aquellos elementos pertenecientes al ACME Framework, según las directrices docentes.

A continuación, se detallan los *bad smells* detectados durante el desarrollo y cómo han sido corregidos para garantizar un código limpio, mantenible y conforme a las buenas prácticas de programación en Java y Spring Framework.

Bad smells detectados y corregidos

1. Inyección por campo (@Autowired)

- **Ubicación:** Todos los controladores y servicios personalizados (AssistanceAgentClaimController, TrackingLogCreateService, AdministratorClaimController, AdministratorAssistanceAgentController, etc.)
- **Problema:** Uso de @Autowired en campos privados.
- **Corrección aplicada:** Reemplazado por inyección mediante constructor con campos final, cumpliendo las recomendaciones de Spring y SonarLint.

2. Duplicación de literales ("resolution", "indicator", etc.)

- **Ubicación:** Servicios relacionados con TrackingLog.
- **Problema:** Repetición de literales en múltiples llamadas (bindObject, unbindObject, state(...), etc.)
- **Corrección aplicada:** Extracción de los valores a constantes private static final String, mejorando la mantenibilidad y eliminando la duplicación.

3. Complejidad cognitiva excesiva en el método validate(...)

- **Ubicación:** TrackingLogPublishService.
- **Problema:** El método validate superaba el umbral de complejidad permitido (> 15).
- **Corrección aplicada:** Refactorización en métodos privados auxiliares (por ejemplo: validateClaimsPublished, validateResolutionPercentageLimits, etc.), reduciendo la complejidad y mejorando la claridad.

4. Métodos vacíos sin comentario

- **Ubicación:** En algunos métodos bind o validate de los servicios personalizados de AssistanceAgent y TrackingLog.
- **Problema:** Métodos vacíos como validate(...) sin explicación.
- **Corrección aplicada:** Añadido comentario // Intentionally left blank para justificar la ausencia de lógica.

5. Repetición innecesaria de expresiones (collection.size() > 0)

- **Ubicación:** En AssistanceAgentDashboardShowService.
- **Problema:** Uso de collection.size() > 0 en lugar de !collection.isEmpty().
- **Corrección aplicada:** Sustitución por isEmpty() para mayor claridad y eficiencia.

6. Uso innecesario de Collectors.toList() seguido de .stream()

- **Ubicación:** TrackingLogCreateExceptionalCaseService, entre otros.
- **Problema:** Patrón stream().collect(Collectors.toList()).stream() innecesario, genera una lista temporal inútil.
- **Corrección aplicada:** Eliminado el .collect(Collectors.toList()) cuando no era necesario y se podía trabajar directamente sobre el Stream.

Conclusión

Todos los *bad smells* detectados por SonarLint han sido **corregidos de forma explícita**. El código final cumple con las buenas prácticas de diseño, mantiene alta legibilidad y está preparado para su validación por parte del equipo docente. Se han respetado las excepciones del ACME Framework y no se han ignorado malas prácticas dentro del ámbito del desarrollo asignado.