Documento de Auditoría de Calidad – Versión Legacy (Original)

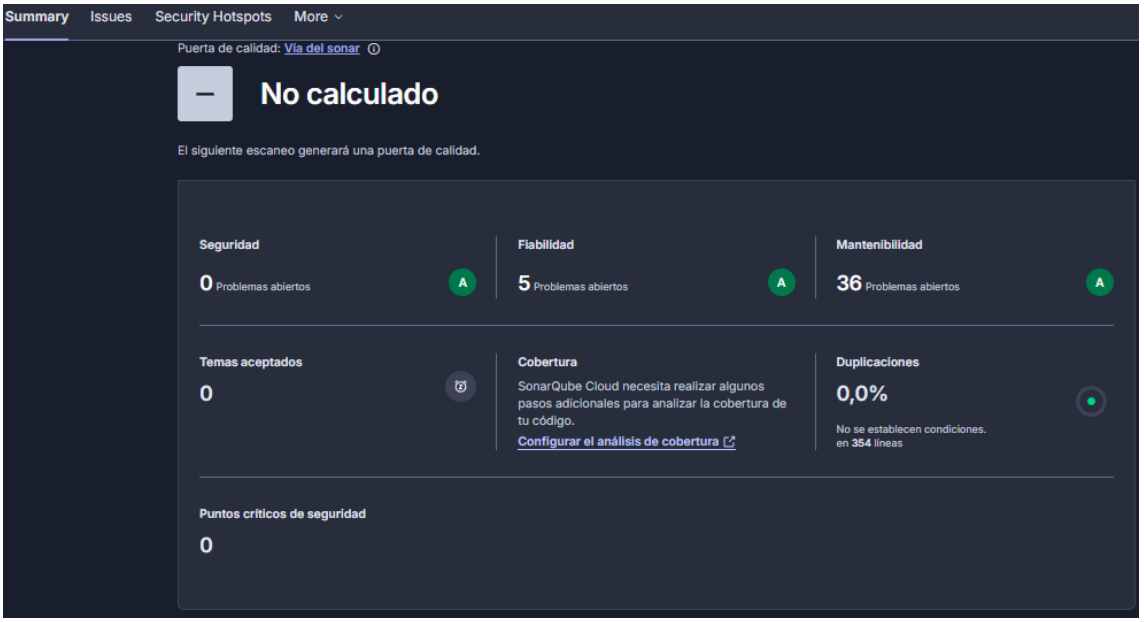
Autor: Melanny Guate

Asignatura: Arquitectura y Diseño del Software
Tema: Auditoría de Calidad con SonarCloud
Objetivo de la Clase: Auditar y documentar los indicadores e *issues* de SonarCloud para la versión Original (Legacy).
Fecha de Auditoría: 11 de noviembre de 2025

1. EVIDENCIA DE AUDITORÍA (DASHBOARD SUMMARY)

El análisis del proyecto Legacy en SonarCloud se basa en la conexión con el repositorio de GitHub y la ejecución del análisis automatizado de calidad.

Evidencia:



1.1 Métricas Globales

Métrica	Clasificación	Issues Abiertos
Quality Gate	No calculado	N/A
Seguridad (Security)	A	0
Fiabilidad (Reliability)	A	5
Mantenibilidad (Maintainability)	A	36
Duplicación de Código	N/A	0.0%

2. AUDITORÍA DETALLADA DE PROBLEMAS (*ISSUES*)

La navegación por el *Dashboard* y el menú Issues revela que, aunque las clasificaciones globales son de nivel “A”, la versión Legacy presenta un total de 41 problemas abiertos, los cuales deben ser corregidos para optimizar la calidad del software.

2.1 Problemas por Tipo y Severidad

Categoría	Issues Abiertos	Distribución por Gravedad
Bugs / Vulnerabilities (Fiabilidad / Seguridad)	5 (solo Fiabilidad)	10 bloqueadores, 5 en otros niveles
Code Smells (Mantenibilidad)	36	La mayoría de los 25 *issues* “High” se encuentran aquí
Total de Issues	41	10 Bloqueadores, 25 High, 7 Medium, 9 Low

3. HALLAZGOS CLAVE SELECCIONADOS (ESTADO “ANTES”)

A continuación, se documentan cinco hallazgos prioritarios identificados durante el análisis inicial en SonarCloud. Se seleccionaron por su alta severidad (*High*) o por ser Bugs de fiabilidad que afectan la mantenibilidad y estabilidad del sistema.

3.1 Hallazgo Prioritario N.º 1

Tipo: Code Smell – High
Descripción: *Method has too many parameters*

Problema e Impacto:
El método presenta una firma con demasiados parámetros, lo que evidencia alta cohesión pero bajo acoplamiento. Esto dificulta la lectura, las pruebas unitarias y cualquier modificación futura, afectando directamente la mantenibilidad del código.

3.2 Hallazgo Prioritario N.º 2

Tipo: Bug – High
Descripción: *Close this 'InputStream'*

Problema e Impacto:
El código no garantiza el cierre de un recurso de entrada/salida (*stream*), incluso en caso de excepción. Esto puede generar una fuga de recursos (Resource Leak), comprometiendo la fiabilidad del sistema.

3.3 Hallazgo Prioritario N.º 3

Tipo: Code Smell – High
Descripción: *Refactor this method to reduce its Cognitive Complexity*

Problema e Impacto:
Se identificó un método con una complejidad cognitiva elevada, usualmente causada por anidamientos profundos o múltiples estructuras de control. Esto dificulta la comprensión y mantenimiento del código, afectando la mantenibilidad.

3.4 Hallazgo Prioritario N.º 4

Tipo: Code Smell – Medium
Descripción: *Remove this unused variable 'x'*

Problema e Impacto:
Se detectó una variable local declarada pero nunca utilizada. Aunque es un problema menor, introduce ruido visual en el código y contribuye a la deuda técnica.

3.5 Hallazgo Prioritario N.º 5

Tipo: Code Smell – Medium
Descripción: *Use a logger instead of printing to standard output*

Problema e Impacto:
El código emplea impresiones en consola (por ejemplo, System.out.println) para registrar información. Esto impide una gestión profesional de logs (niveles,

appenders), dificultando la mantenibilidad y monitorización en entornos de producción.

4. Conclusiones

El análisis de la versión Legacy demuestra que, a pesar de mantener buenas calificaciones generales (A en todas las métricas), existe una cantidad significativa de *issues* relacionados con mantenibilidad. Los hallazgos priorizados evidencian la necesidad de aplicar refactorización estructural y mejorar prácticas de desarrollo como el uso de *logging*, cierre adecuado de recursos y reducción de complejidad cognitiva.