

# Reporte de Auditoría de Calidad: Revisión Manual vs. Herramientas Automáticas

Este reporte compara los hallazgos de la revisión manual (basada en principios de Clean Code y SonarLint) con los resultados de las herramientas de análisis estático configuradas en el pipeline de CI/CD (Checkstyle y PMD).

Categoría de Calidad	Hallazgo Manual (SonarLint)	Hallazgo Checkstyle	Hallazgo PMD	Conclusión y Acción Recomendada	Issue Creado
1. Cohesión y Responsabilidad (SRP)	La clase <code>Inventory</code> mezcla lógica de negocio con impresiones en consola ( <code>System.out</code> ).	No detectado.	No detectado.	<b>Crítico.</b> La revisión manual fue esencial para detectar esta violación al principio SOLID. Requiere refactorización.	#9
2. Duplicación de Código (DRY)	Existen dos métodos casi idénticos: <code>getTotalValue()</code> y <code>totalInventoryValue()</code> .	No detectado.	No detectado.	<b>Crítico.</b> La revisión manual identificó código duplicado que las herramientas pasaron por alto.	#10
3. Estilo y Formato del Código	El formato general es inconsistente.	<b>113 violaciones.</b> La mayoría son de <code>Indentation</code> , lo que demuestra una falta total de formato consistente.	No detectado.	<b>Mayor.</b> Aunque no afecta la lógica, dificulta gravemente la legibilidad. Se recomienda aplicar un formateador automático en todo el proyecto.	N/A (Problema global)

Categoría de Calidad	Hallazgo Manual (SonarLint)	Hallazgo Checkstyle	Hallazgo PMD	Conclusión y Acción Recomendada	Issue Creado
4. Calidad de la Documentación (Javadoc)	La documentación es incompleta, faltan descripciones y etiquetas <code>@return</code> .	<b>13 violaciones.</b> Detecta problemas de estilo como Javadocs de una sola línea ( <code>SingleLineJavadoc</code> ) y resúmenes faltantes.	No detectado.	<b>Mayor.</b> Ambas revisiones (manual y automática) confirman que la documentación es deficiente y no cumple el objetivo del 90%.	#17, #18

Resumen de la Auditoría

- **Revisión Manual (con SonarLint):** Fue fundamental para detectar los problemas más críticos de **diseño de software y estructura** (violaciones a los principios SRP y DRY), los cuales las otras herramientas no encontraron.
- **Checkstyle:** Demostró ser extremadamente útil para detectar problemas de **formato y estilo** a gran escala, como la indentación inconsistente, que son difíciles de revisar manualmente en todo un proyecto.
- **PMD:** Con la configuración actual del proyecto, no encontró problemas. Esto indica que el código no contiene los "code smells" o defectos más comunes que este busca, pero no garantiza su calidad general.

Conclusión Final

La calidad del software solo puede medirse de manera efectiva combinando la **rigurosidad de las herramientas automáticas** (que encuentran problemas de estilo y patrones conocidos) con el **criterio y la experiencia humana de un revisor** (que encuentra problemas de diseño y lógica más profundos).