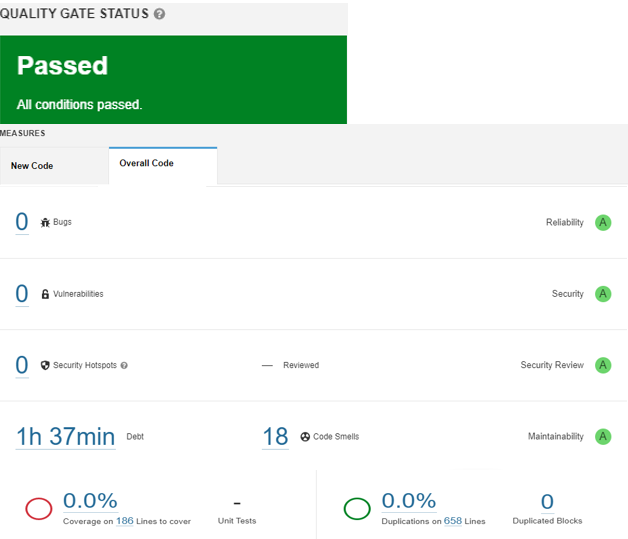
**Reporte SonarQube**

|  |  |
| --- | --- |
| Emitido por: | Yolima Alejandra Guadir Paguay |
| Analista de: | SofkaU |
| Repositorio probado | <https://github.com/Andrescc143/C1-2023-FA-Springboot> |
| Fecha: | 03/03/2023 |

Mediante el presente informe se da a conocer los resultados de las métricas aplicadas para medir la calidad del código al ser probado por la herramienta de análisis estático SonarQube.



|  |  |
| --- | --- |
| **Bugs** | La herramienta de análisis estático de código no encontró ningún resultado que coincida con los criterios predeterminados que están asociados a detectar bugs. |
| **Vulnerabilidades** | La herramienta de análisis estático de código no encontró ningún resultado que coincida con los criterios predeterminados que están asociados a vulnerabilidades. |
| **Seguridad** | La herramienta de análisis estático de código no encontró ningún resultado que coincida con los criterios predeterminados que están asociados a la seguridad. |
| **Cobertura** | La herramienta indica 168 líneas de código no fueron ejecutadas, es decir no existe garantía que el código funcione correctamente. |
| **Duplicación en líneas** | La herramienta indica que hay secciones de código en la aplicación que son idénticas a otras secciones de código en la misma aplicación y que en total, suman 658 líneas de código duplicado. |
| **Duplicación en Blocks** | La herramienta indica que no hay secciones de código en la aplicación que sean idénticas o muy similares a otras secciones de código en la misma aplicación. |
| **Code small** | La herramienta de análisis identificó que el 18% de código tiene problemas de diseño o implementación que pueden afectar la calidad del software, algunos códigos small identificados son los siguientes. |
|  | |

Resumen de informe con respecto a métricas implementadas por la herramienta sonarQube

|  |  |
| --- | --- |
| **Métrica** | **Estado** |
| Reliability | Cumple |
| Security | Cumple |
| Security review | Cumple |
| Maintainibility | No cumple |

El código no cumple con la métrica de mantenibilidad por lo tanto para mejorar la mantenibilidad del sistema se recomienda:

1. Identificar las secciones de código duplicado y tomar medidas para refactorizar o reutilizar el código en lugar de duplicarlo.
2. Aplicar principios de diseño.
3. Aplicar principios de diseño de software.