Evolución de Métricas Web SonarQube

Lumiere

Índice

Introducción

Este documento analiza la evolución de las métricas de calidad del código en el proyecto **Lumiere Catering**, comparando la primera medición con la métrica final en SonarQube.

1. Resumen de las Métricas Analizadas

Las métricas evaluadas incluyen:
- **Issues detectados** (errores, vulnerabilidades, code smells).
- **Cobertura del código** (porcentaje de líneas cubiertas por pruebas).
- **Duplicación del código** (porcentaje de código repetido).
- **Hotspots de seguridad** (zonas del código con posibles riesgos de seguridad).

2. Primera Métrica de SonarQube

Estado Inicial del Proyecto

- **Issues abiertos:** Alta cantidad de problemas detectados.
- **Code Smells:** Varias advertencias sobre calidad del código.
- **Hotspots de seguridad:** Presencia de elementos potencialmente riesgosos.
- **Cobertura del código:** No se registraron pruebas automatizadas.
- **Duplicación del código:** Código con segmentos repetidos.

Principales Problemas Identificados

- Uso incorrecto de atributos en etiquetas “<meta>”.
- Falta de validaciones en formularios.
- Ausencia de pruebas unitarias y de integración.
- Código redundante en algunos módulos.

3. Métrica Final de SonarQube

Estado Tras Mejoras Implementadas

- **Issues abiertos:** Reducción significativa de errores y advertencias.
- **Code Smells:** Mejoras en la legibilidad y estructura del código.
- **Hotspots de seguridad:** Eliminación de prácticas inseguras.
- **Cobertura del código:** Implementación de pruebas automatizadas.
- **Duplicación del código:** Reducción de código redundante.

Mejoras Implementadas

- Se corrigieron las etiquetas “<meta>” y se eliminaron vulnerabilidades de seguridad.
- Se implementó validación en formularios con JavaScript.
- Se refactorizó el código para mejorar la calidad y mantenibilidad.
- Se añadieron pruebas unitarias para evaluar funcionalidad.

4. Comparación entre Ambas Métricas

Métrica	***Primera Medición***	***Medición Final***	***Cambio***
Issues abiertos	Alto	Bajo	Reducción significativa
Code Smells	Muchos	Pocos	Disminución
Hotspots de Seguridad	Presentes	Eliminados	Resuelto
Cobertura de Código	0%	40.7%	Incremento positivo
Duplicación del Código	Alta	Baja	Optimización

5. Conclusiones

- Se logró una **mejora sustancial** en la calidad del código, reduciendo problemas de seguridad y mejorando la mantenibilidad.
- La cobertura de código aumentó significativamente con la integración de pruebas automatizadas.
- Se aplicaron buenas prácticas para reducir la duplicación y mejorar la eficiencia.
- El proyecto ahora cumple con **estándares de calidad más altos**, garantizando mayor estabilidad y escalabilidad.