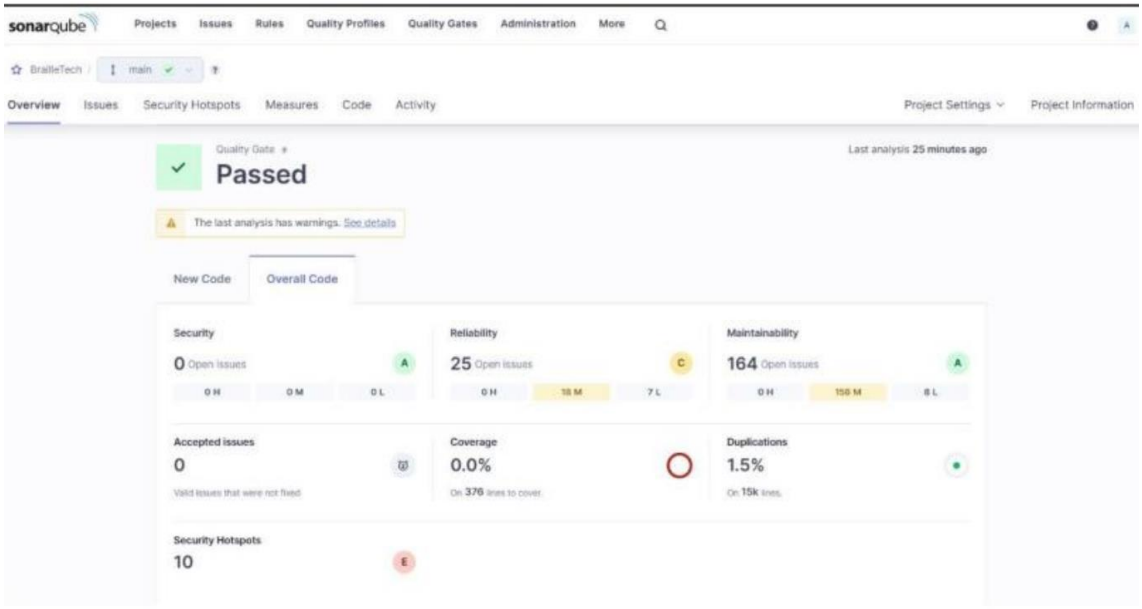
ESCUELA POLITÉCNICA NACIONAL

ISWD652 CALIDAD DE SOFTWARE

NOMBRE ESTUDIANTE:	Max Carrión Milton Pástor Eddy Arias Jonathan Poaquiza Fredviner Bailón
FECHA:	31-07-2024
TEMA:	Elaborar un reporte de los resultados de SonarQube, incluir capturas de todos los resultados que indica la herramienta.

APRENDIZAJE OBTENIDO

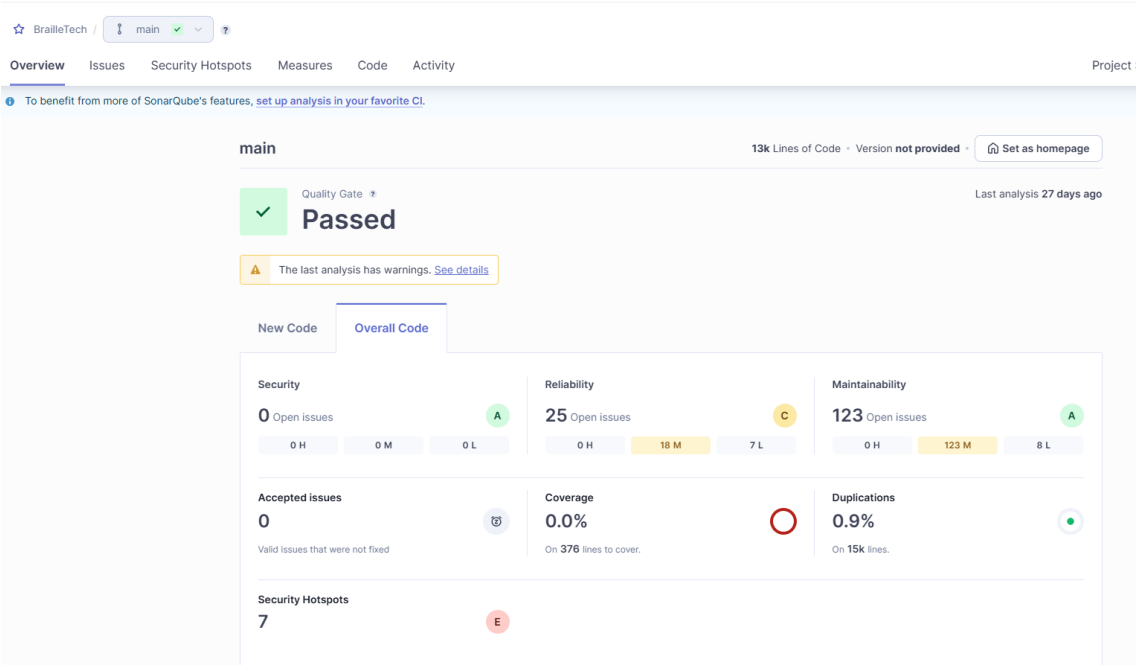
A través de la realización de este proyecto, hemos obtenido un entendimiento más profundo sobre la importancia del análisis y la mejora continua en la calidad del código de software. Al comparar el estado del código antes y después de aplicar mejoras utilizando SonarQube, pudimos identificar y corregir varios problemas que impactaban la mantenibilidad, seguridad y duplicación de código.



Reporte Inicial de SonarQube (Antes de las mejoras)

En la primera imagen del reporte de SonarQube antes de las mejoras, podemos observar lo siguiente:

- **Security:** No se identificaron problemas de seguridad abiertos, lo cual es positivo. Sin embargo, había 10 puntos calientes de seguridad identificados que requerían revisión.
- **Reliability:** Se reportaron 25 problemas abiertos con un tiempo estimado de remediación de 18 minutos, manteniendo una calificación A en fiabilidad.
- **Maintainability:** Este aspecto era el más crítico, con 164 problemas abiertos y una deuda técnica significativa de 155 minutos, lo que resultaba en una calificación A.
- **Coverage:** La cobertura del código era nula, es decir, el 0% de las líneas de código estaban cubiertas por pruebas, lo cual es un indicador negativo.
- **Duplications:** El porcentaje de duplicaciones en el código era de 1.5%, lo que indica la presencia de bloques de código repetidos, que afecta la mantenibilidad.



Reporte Posterior de SonarQube (Después de las mejoras)

En la segunda imagen, después de las mejoras aplicadas, se observan los siguientes cambios:

- **Security:** El número de puntos calientes de seguridad se redujo de 10 a 7, lo que demuestra una mitigación efectiva de ciertos riesgos de seguridad.
- **Reliability:** Se mantuvieron los 25 problemas abiertos con el mismo tiempo estimado de remediación, pero la calificación cambió de A a C. Este cambio podría deberse a la re-evaluación de la gravedad de los problemas.

- **Maintainability:** Hubo una reducción significativa en el número de problemas abiertos de mantenibilidad, pasando de 164 a 123, lo cual es una mejora importante que reduce la deuda técnica.
- **Coverage:** No hubo cambios en la cobertura del código, que se mantuvo en 0%. Esto sugiere que no se añadieron nuevas pruebas o que no se incrementó la cobertura de código existente.
- **Duplications:** El porcentaje de duplicaciones disminuyó de 1.5% a 0.9%, lo que indica un esfuerzo exitoso por reducir el código duplicado y mejorar la calidad general.

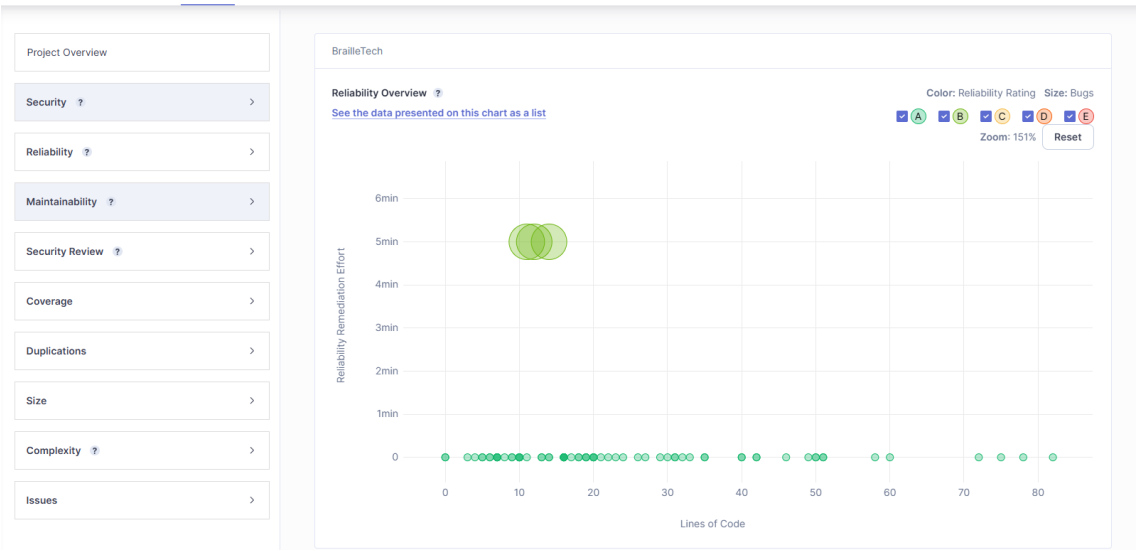


Gráfico de Seguridad

El gráfico de seguridad ilustra las vulnerabilidades en relación con el esfuerzo de remediación. La mayoría de las vulnerabilidades se concentran en un rango de 40 a 100 líneas de código, y requieren un esfuerzo de remediación que varía de bajo a medio. La densidad de los problemas parece haberse reducido tras las mejoras, lo que indica un avance en la mitigación de riesgos de seguridad.

Gráfico de Cobertura

Este gráfico muestra la cobertura del código en función de la complejidad ciclomática. Aunque no se observa una mejora en la cobertura, se visualiza que la cobertura es nula en todos los niveles de complejidad, lo que sugiere la necesidad de implementar más pruebas unitarias.

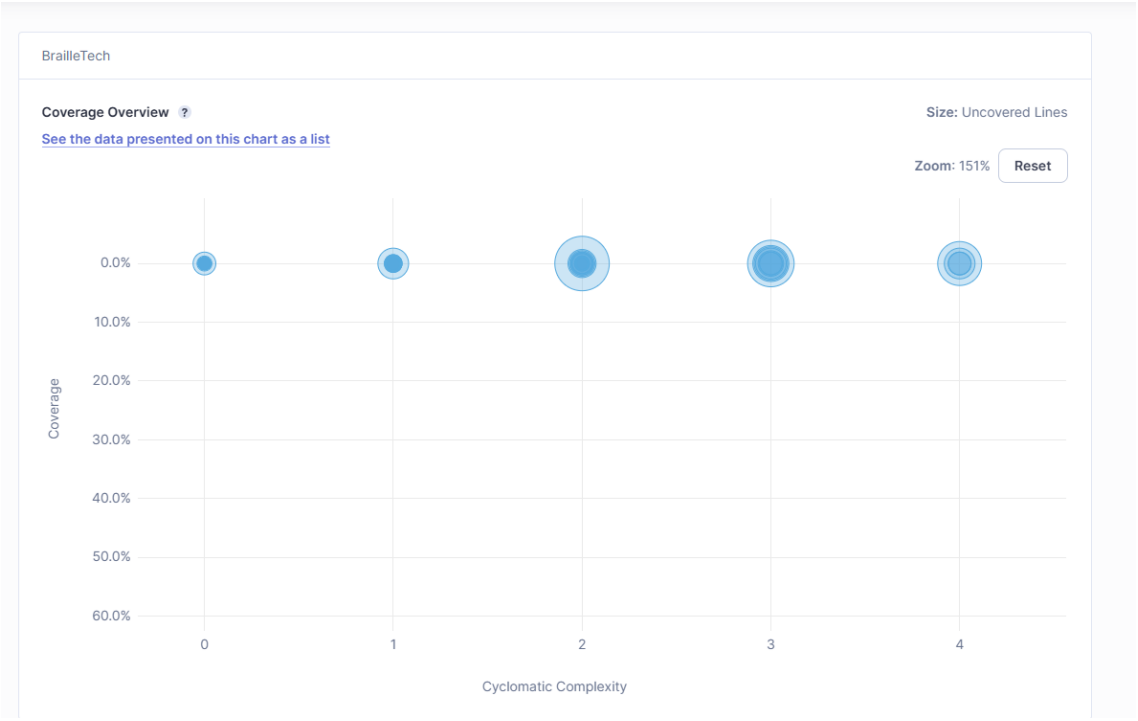
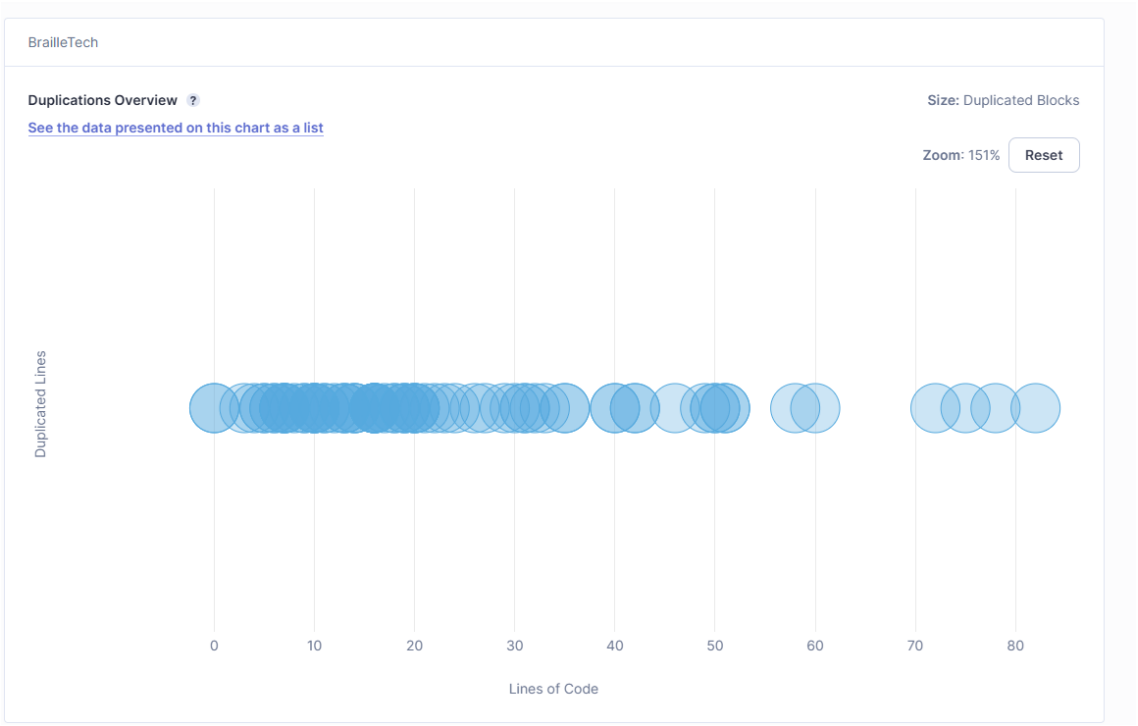


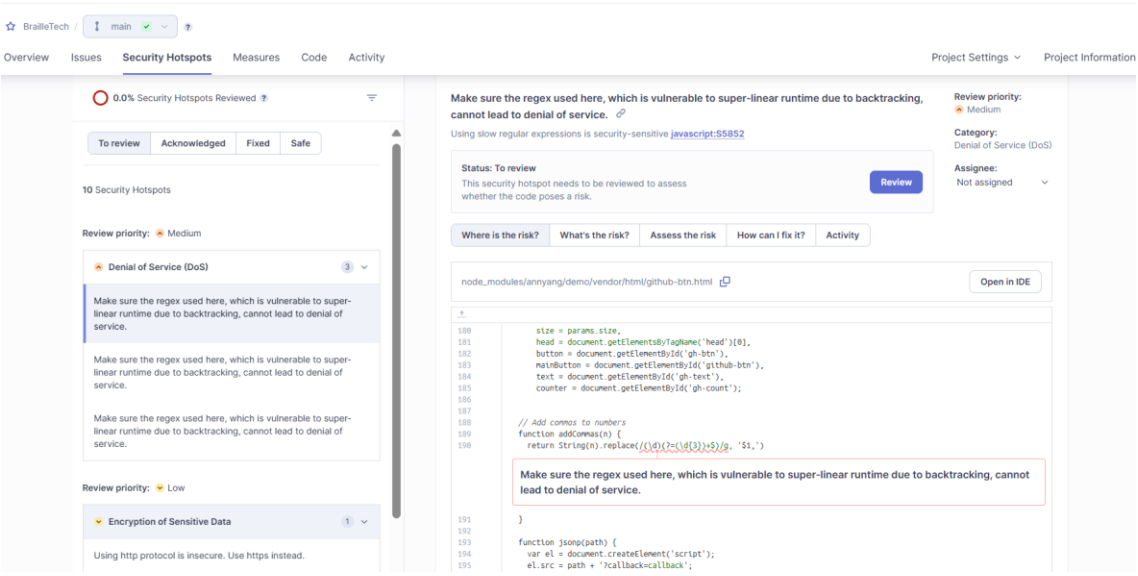
Gráfico de Duplicaciones

En este gráfico se observa una reducción de la cantidad de líneas de código duplicadas después de las mejoras. Los bloques de código duplicados se concentran principalmente en módulos de código más extensos, y la reducción en duplicaciones indica una mejora en la estructura del código.



Puntos Calientes de Seguridad

En la última imagen se detallan los puntos calientes de seguridad identificados. Se observa una mejora en la gestión de estos puntos, con una reducción en su cantidad. Sin embargo, aún quedan elementos pendientes que requieren atención, particularmente en áreas críticas como la protección contra ataques DoS (Denial of Service).



Conclusión

El análisis comparativo entre el estado del código antes y después de las mejoras muestra un progreso significativo en la reducción de problemas de mantenibilidad y duplicación de código. A pesar de que aún hay áreas que requieren atención, como la cobertura del código y ciertos aspectos de la fiabilidad, las mejoras realizadas han tenido un impacto positivo en la calidad general del código, reduciendo la deuda técnica y mejorando la seguridad del proyecto. Es recomendable seguir trabajando en la cobertura de código y en la resolución de los problemas de fiabilidad para continuar mejorando la calidad del software.