

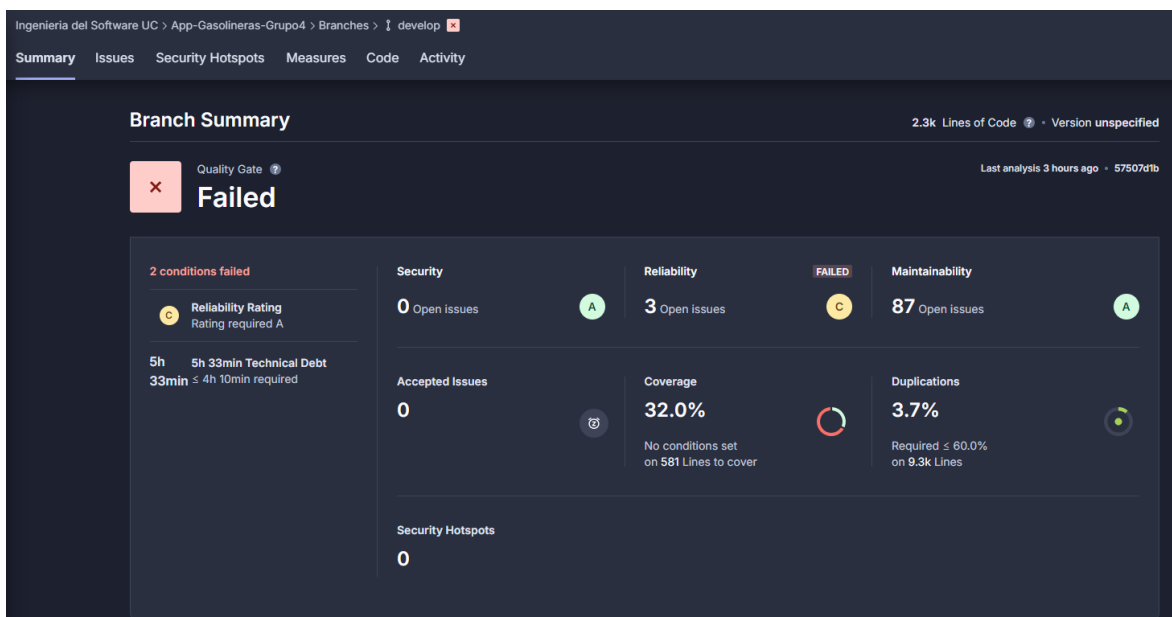
Informe de Calidad

Primer Sprint, Grupo 4

Fecha de realización: 24 de octubre de 2024

Hash del commit: 09e564c8f798f6baab179b59949f89ab184a2ade

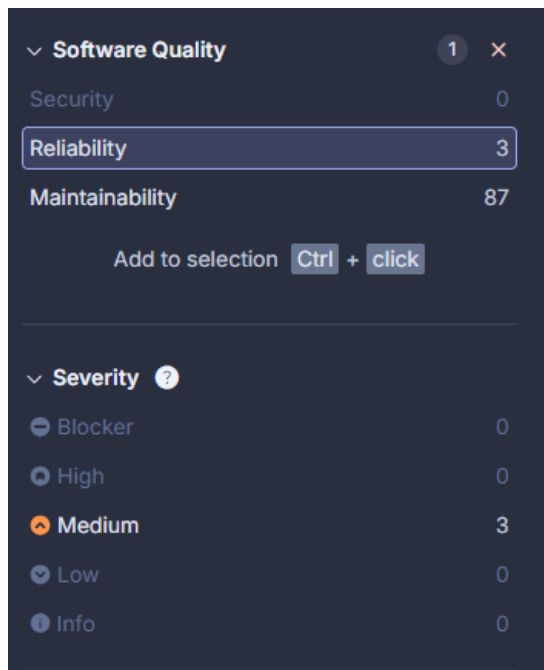
Captura del Resumen del Análisis:



El quality gate falló por tres razones principales: la primera, es porque el Reliability no obtuvo la puntuación requerida (A). La segunda razón, fue porque la deuda técnica superó las cuatro horas con diez minutos. Y la tercera razón, es porque al realizar la prueba con SonarQube dentro de la aplicación, solamente el 32% de líneas de código están siendo probadas. Esto indica que hay un 68% de líneas de código que no han sido probadas y esto permite que existan partes del algoritmo que pueden contener bugs no detectados al no haber sido analizados.

Severidad de los issues de cada elemento principal:

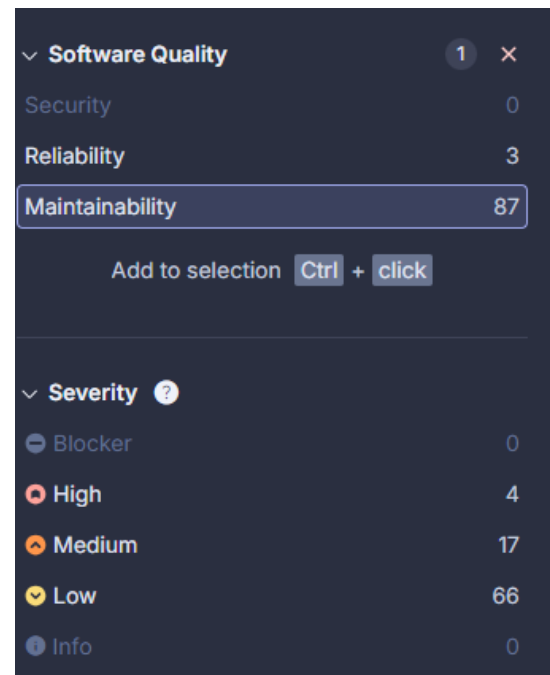
Reliability:



Software Quality	
Security	0
Reliability	3
Maintainability	87
Add to selection Ctrl + click	
Severity	
Blocker	0
High	0
Medium	3
Low	0
Info	0

En esta captura de pantalla, se aprecia que existen tres errores con severidad media de Reliability y es urgente resolverlos.

Maintainability:

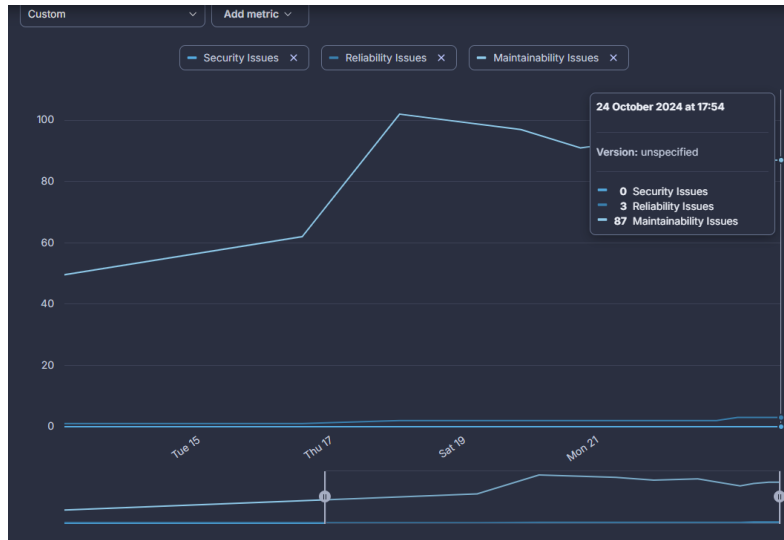


Software Quality	
Security	0
Reliability	3
Maintainability	87
Add to selection Ctrl + click	
Severity	
Blocker	0
High	4
Medium	17
Low	66
Info	0

En esta imagen, se puede observar que hay varios problemas de distinta severidad, existiendo cuatro con severidad alta de mantenibilidad.

Gráfica de número de issues de cada uno de los tres elementos:

Se realiza desde el 14 de octubre, ya que fue el primer análisis de calidad con SonarQube realizado.



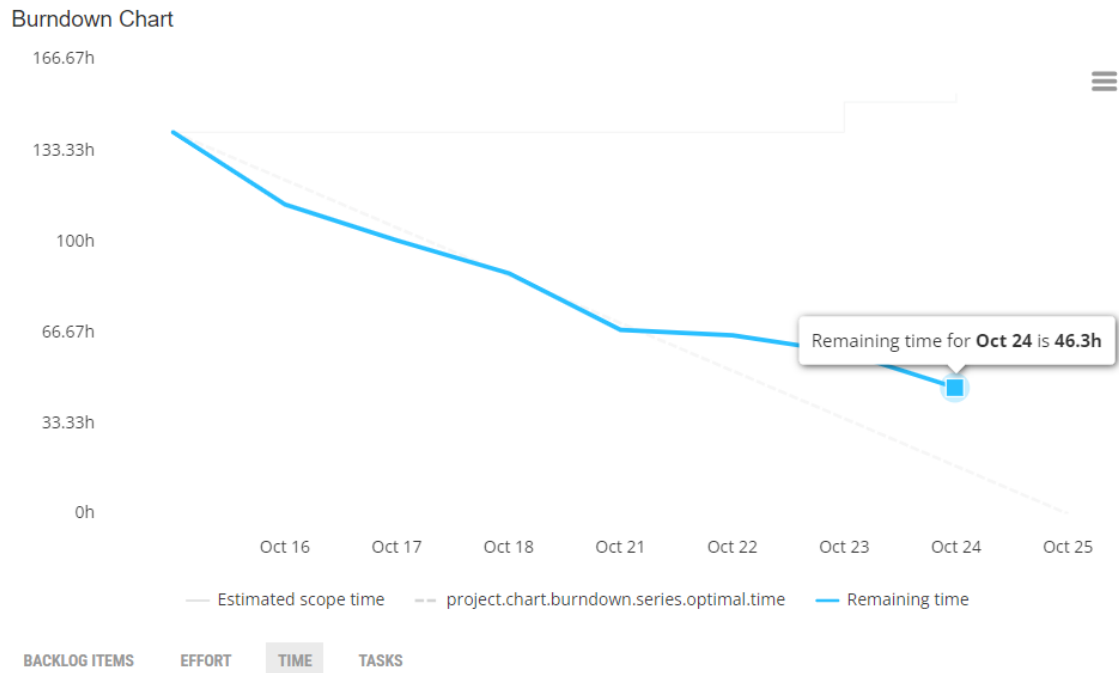
En esta gráfica, la mantenibilidad es el área más crítica, con un incremento de incidencias. La fiabilidad también presenta problemas, aunque en menor cantidad, pero necesitan atención inmediata para evitar más fallos.

Porcentaje de código repetido y complejidad ciclométrica:



En esta gráfica, se puede evidenciar que la tanto la complejidad como la duplicidad de los algoritmos han aumentado desde el 17 de octubre, lo que hace que el código sea más complejo y afecte negativamente la mantenibilidad del mismo.

Burndown Chart:



Plan de Acción

Al analizar todas las gráficas, nos podemos dar cuenta que existen varios problemas por solucionar, pero debido al poco tiempo restante del sprint, la prioridad será resolver las incidencias de Reliability y reducir la deuda técnica para poder cumplir con el Quality Gate.

1. Incidencias a Solventar

Incidencia	Prioridad	Justificación
A "NullPointerException" could be thrown; "tipoCombustible" is nullable here.	Alta	Este tipo de error es crítico porque puede causar fallos en tiempo de ejecución que interrumpen el funcionamiento normal de la aplicación, generando un mal funcionamiento o incluso un bloqueo total. Además, se debe solucionar para cumplir con el Quality Gate.
Reducción de deuda técnica en PaymentHistoryView.java y PaymentHistoryPresenter	Alta	Reducir la deuda técnica es prioritario porque mejora la mantenibilidad del código y evita que problemas menores se acumulen, lo que puede llevar a complicaciones mayores en el futuro.

Add a default case to this switch	Baja	Falta un caso por defecto en un switch es una causa potencial de comportamiento inesperado, especialmente si se manejan nuevos valores en el futuro que no están contemplados.
-----------------------------------	------	--