SonarQube

Proyecto 1 DAW - Entornos de desarrollo

Ainoa y Javier
22/05/2025

ÍNDICE

Proceso de escaneo

Para realizar el análisis del código con SonarQube, ha sido necesario instalar previamente SonarQube y SonarScanner. Una vez instaladas ambas herramientas, se ha descargado el repositorio del proyecto deseado desde GitHub.

En el repositorio, ha sido necesario modificar el archivo .gitignore para eliminar la exclusión de la carpeta bin, ya que por defecto está ignorada. Esta carpeta contiene los archivos .class, que SonarScanner necesita analizar en proyectos Java para evaluar correctamente el código compilado.

Adicionalmente, se ha eliminado del proyecto una carpeta que contenía el conector de MySQL, ya que también incluía archivos .class que no deben ser analizados junto al resto del proyecto.

Con el proyecto preparado, se ha iniciado el servidor de SonarQube y se ha accedido a la carpeta del proyecto desde el terminal. Desde allí, se ha ejecutado el siguiente comando para lanzar el análisis:

```
sonar-scanner.bat -D"sonar.projectKey=rentacar" -D"sonar.sources=."
-D"sonar.host.url=http://localhost:9000"
-D"sonar.token=sqp_f0a43822455a8ad872aba1fa8a32f1395424aee6"
-D"sonar.java.binaries=bin"
```

Una vez completado el análisis, se han revisado los resultados abriendo la interfaz web de SonarQube en el navegador, donde se muestran los posibles errores, vulnerabilidades, código duplicado y otros aspectos relevantes de la calidad del código.

Sprint 3

Tras el primer escaneo en el 3er sprint, en el que comenzamos a tener algo de código, encontramos 170 issues, la mayoría relacionados con nombres de variables. Estos issues decidimos ignorarlos ya que no sugieren cambios muy significativos en el código.

También aparecen issues relacionados con la seguridad que al final lo que indican es que no debemos tener la contraseña de nuestra bdd en texto plano pero también decidimos no

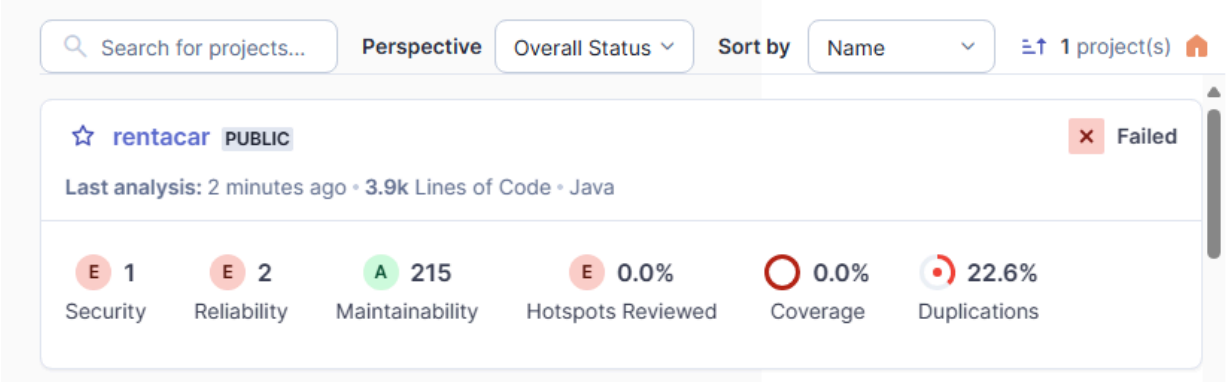
modificarlo ya que no es un problema real en el entorno en el que trabajamos y es un poco engorroso.

Por último los errores que decidimos solucionar están principalmente relacionados con la complejidad de los métodos. Segmentamos algunos métodos más largos en métodos más pequeños y vamos combinándolos para que cada método no sea tan largo como sugiere Sonarqube.

Sprint 4

Igual que en el sprint anterior la mayoría de métodos relacionados con los nombres de las variables y conexión con la base de datos no los gestionamos, pero trabajamos en los que están relacionados con la longitud de las funciones y cosas similares.

Sprint 5



En este último sprint no añadimos mucha cantidad de código como en sprint anteriores por lo que no se modifica mucho la cantidad de issues que reconoce sonarQube.