



Ingeniería en Sistemas Computacionales

Calidad del Software

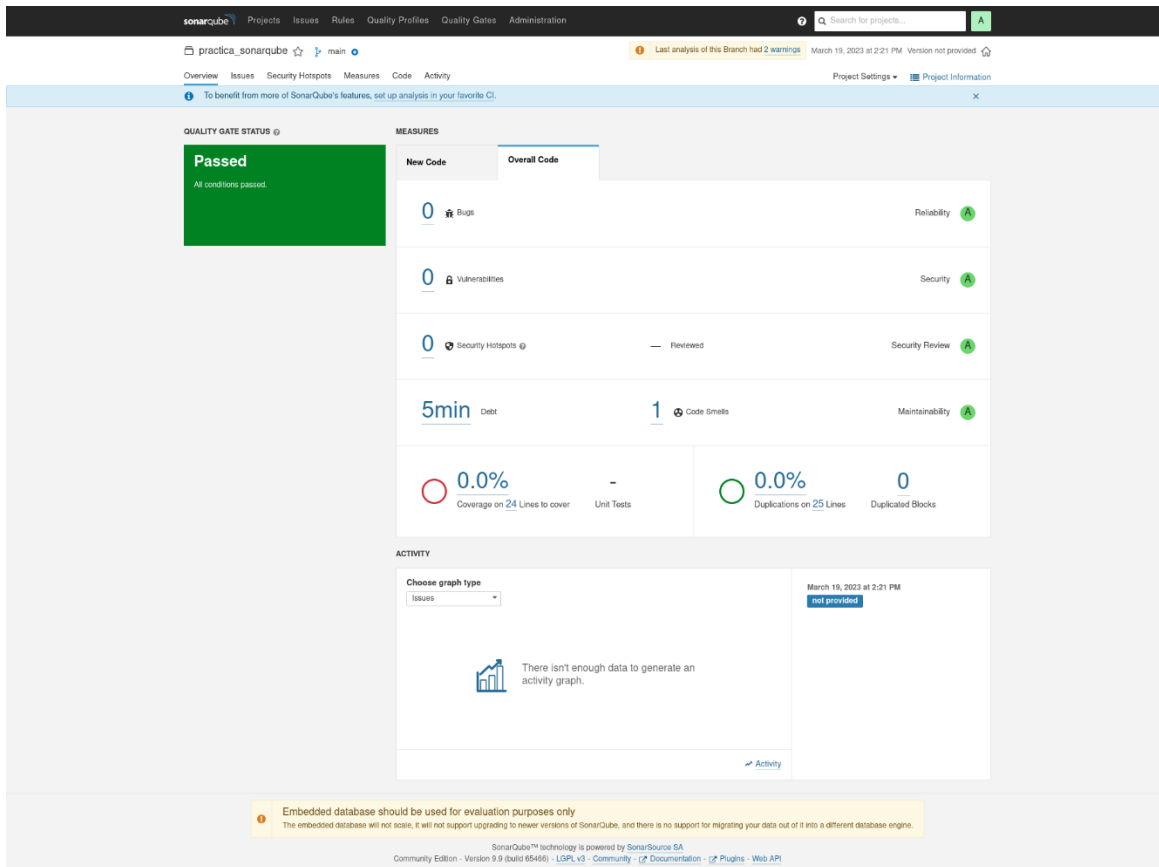
Pruebas estáticas de código – SonarQube

Marco Ricardo Cordero Hernández

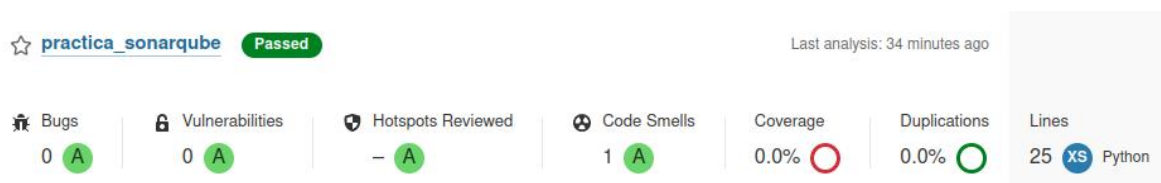
Tlaquepaque, Jal., 21 de marzo de 2023

Para la actividad actual se propone el análisis de código por medio de pruebas estáticas en código.

Retomando el código de algunas entregas pasadas, el cual contaba con la sencilla función de calcular el área de un triángulo, se ha realizado dicho análisis por medio de la herramienta “SonarQube”. Los resultados son los siguientes:

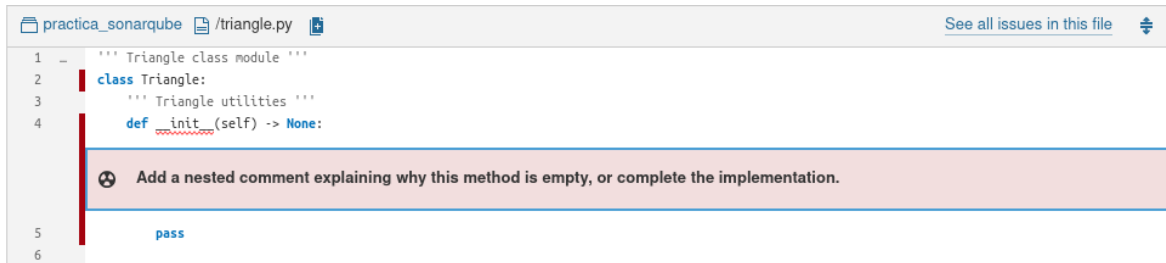


El código cuenta con un grado bastante aceptable de calidad, incluso proporcionando una calificación general catalogada como buena.



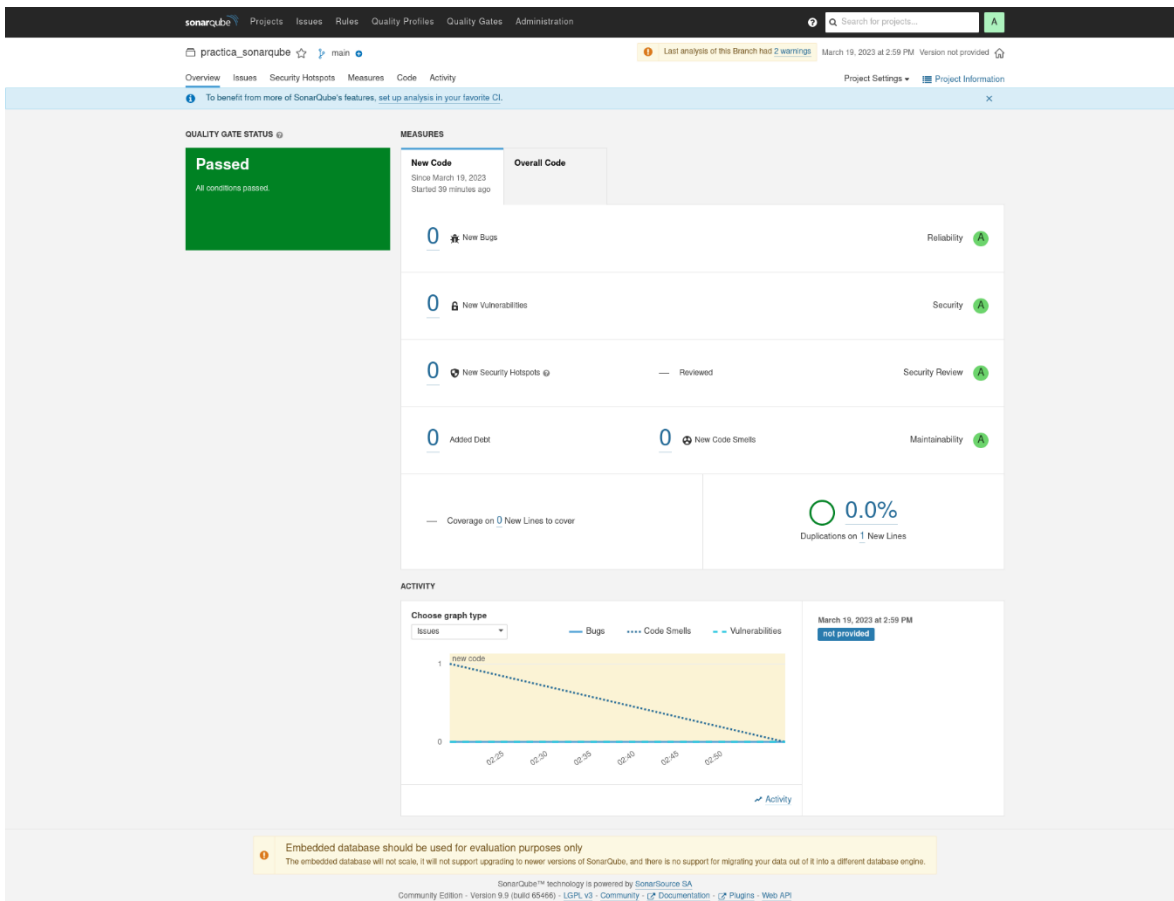
Si bien se ha logrado esto por el uso de PyLint en actividades anteriores, la herramienta no es perfecta (aunque sí tiene un mejor análisis sintáctico que el autor de este reporte), de hecho, gracias a la baja percepción de la calidad en la escritura del código original y algunos detalles que deja pasar el linter utilizado, es que se ha dejado un “code smell” en el proyecto, el cual

indica una deuda técnica de 5 minutos. Al revisar cuál es la falta en cuestión, se puede percatar que es algo que toma a lo mucho 2 minutos en realidad, por si el estimado no era lo bastante reconfortante.



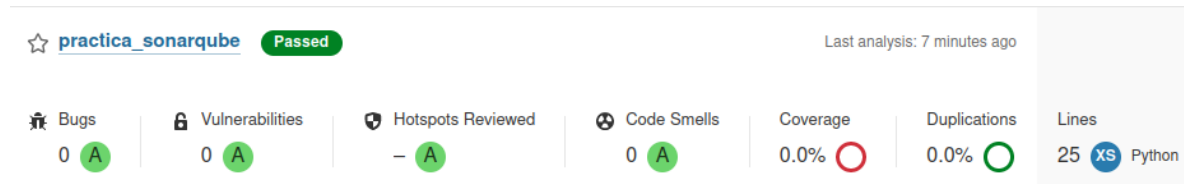
Como se puede apreciar en la imagen, este defecto viene de la ausencia de especificidad en un inicializador vacío en uno de los archivos del proyecto (cabe recordar que este fue escrito en Python). Personalmente hablando, no creo que esta advertencia deba ser pasada por alto, no obstante, este tipo de detalles dependerán mucho de los paradigmas y reglas que cada equipo e inclusive empresa definan para sus desarrollos.

Con las correcciones realizadas, el nuevo reporte es el siguiente:



En esta ocasión, con datos suficientes para realizarlo, una gráfica ha sido generada, la cual muestra la disminución (o más bien supresión) de “code smells” con respecto al análisis anterior. Esto se logró agregando un “docstring” al inicio del inicializador señalado anteriormente, en donde se señala el porqué del método vacío.

Nuevamente, la calidad del código se mantiene, al menos por el momento...



El veredicto de esta actividad es definitivo: aunque puedan parecer una pérdida de tiempo, este tipo de herramientas analistas son mucho más útiles de lo que se pudiera llegar a pensar. Es posible que los detalles que aparentemente son mínimos en primera instancia resulten desastrosos más adelante; Si se mitigan desde etapas tempranas, el desarrollo de proyectos puede llegar a ser ameno y bastante soportable (¡JA! Como si eso pasara en realidad...).