

# Universidad Autónoma de Baja California

## Facultad de Ciencias Químicas e Ingeniería

Plan de Ingeniero en Software y Tecnologías Emergentes



### **Materia**

Verificación y Validación del Software (361)

### **Meta 2.1.3**

Diseña una prueba estática para verificación

### **Docente**

Claudia Gabriel Tona Castro

### **Participante(es)**

[Luis Eduardo Galindo Amaya](#) (1274895)

# Meta 2.1.3

## Diseña una prueba estática para verificación

### Introducción

Las pruebas estáticas son un componente esencial del proceso de aseguramiento de la calidad en el desarrollo de software. Las pruebas estáticas se realizan sin necesidad de ejecutar el código. Por lo general se centran en la revisión del código fuente, el diseño y la documentación relacionada con el software.

### Desarrollo

para esta practica decidí usar sonarlint (SonarQube), para esto instalé el plugin de visual estudio code en la sección de plugins:

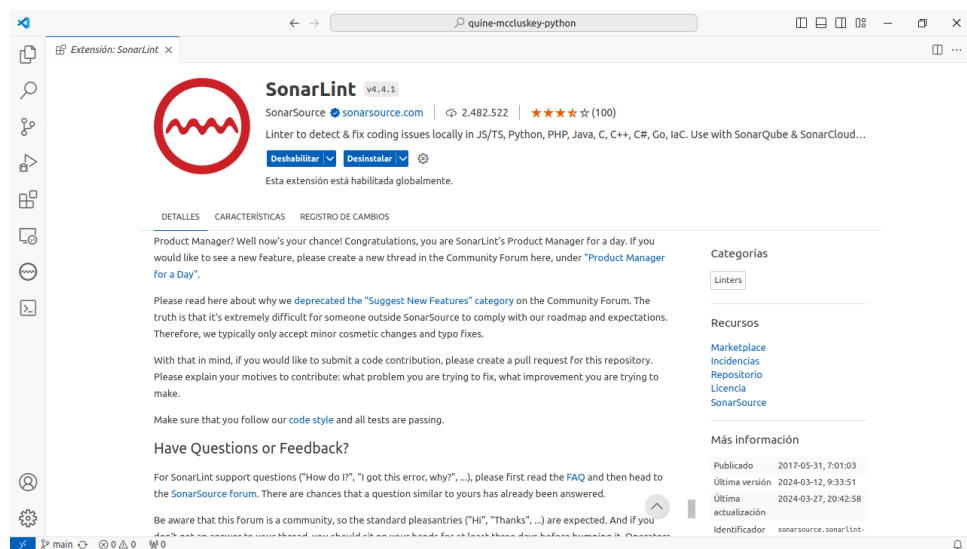


Figura 1: Extenciones de VSCode

Después elegí uno de mis proyectos anteriores, decidí usar un programa para reducir expresiones lógicas al mínimo utilizando el algoritmo de Algoritmo Quine–McCluskey<sup>1</sup>

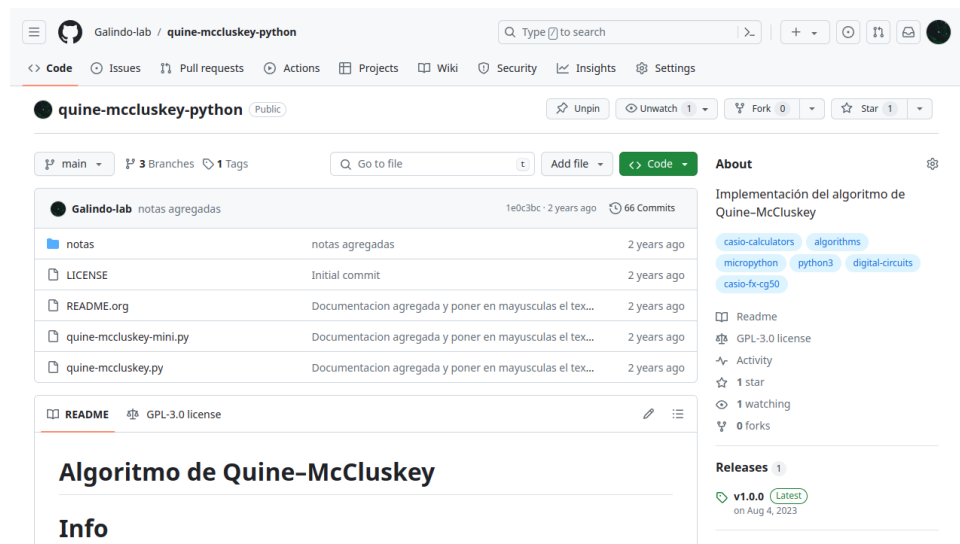


Figura 2: Repositorio del proyecto

A continuación abrí el proyecto en VSCode e inmediatamente Sonarlint empezó a mostrar posibles problemas con el código:

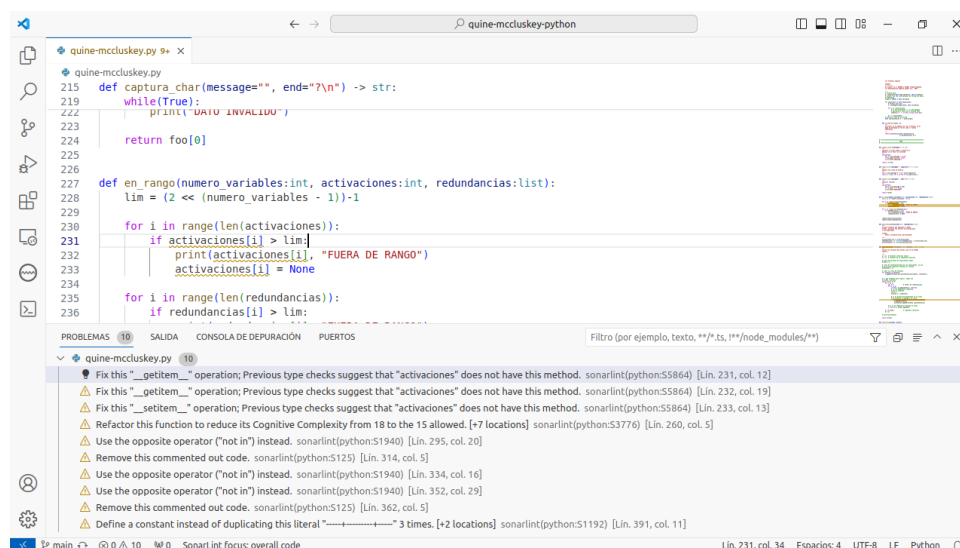


Figura 3: Cosas detectadas por SonarLint

1 <https://github.com/Galindo-lab/quine-mccluskey-python>

Si hacemos clic en las notas que hace sonar lint podemos obtener una descripción detallada sobre el error que esta marcando:

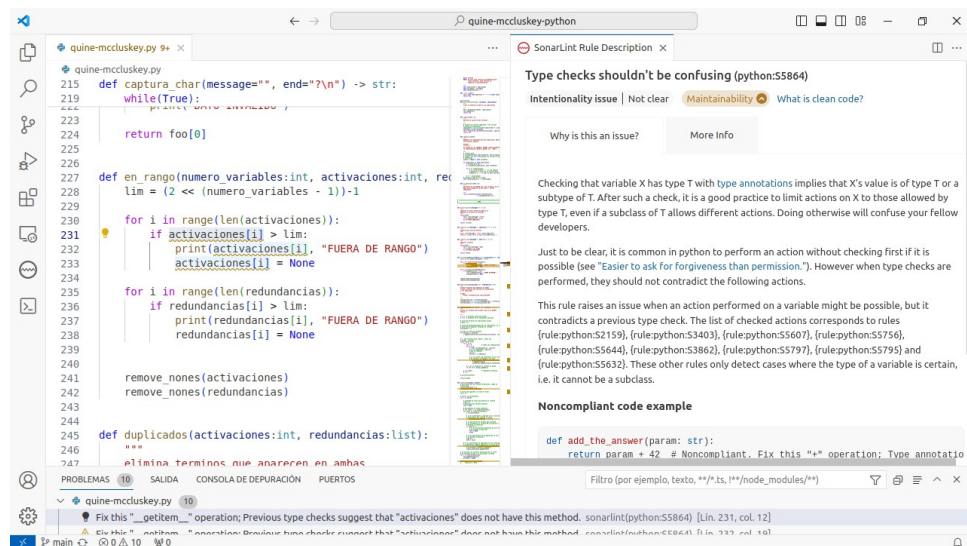


Figura 4: Explicación detallada del problema identificado

Sonarlint ofrece algunas opciones para refactorizar el código, pero son por lo general problemas sencillos como cambiar operadores o invertir ifs:

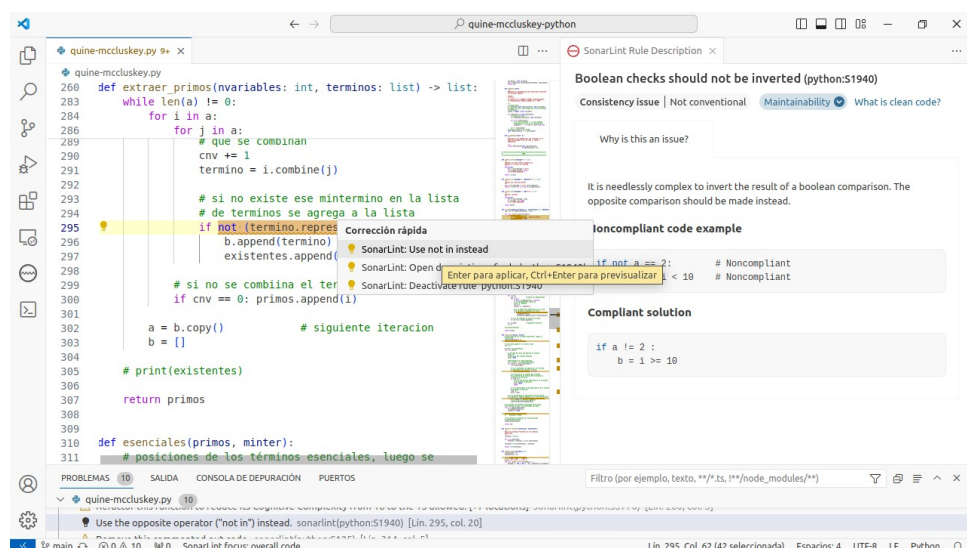


Figura 5: Refactorizaciones de sonarlint

Algunos de los errores que aparecen en sonarlint no son muy graves, la mayoría son simplificaciones de expresiones lógicas o renombrado de variables.

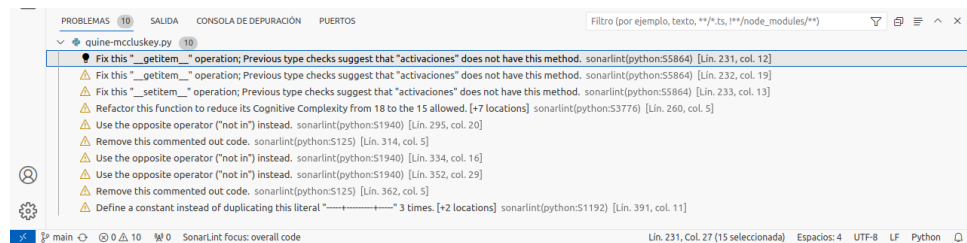


Figura 6: Lista de errores en todo el código

## Conslusion

A lo largo de esta meta aprendí como hacer pruebas estáticas de manera sencilla y en que situaciones pueden ser útiles, las herramientas modernas para análisis estático son tan avanzadas que pueden integrarse fácilmente en cualquier desarrollo, permitiendo que todos los programadores puedan utilizar buenas técnicas de programación durante el desarrollo.