

Jennifer Toxcon, Brandon Sicay,

Diana Fernández, Daniel Morales,

Dáriel Villatoro, Emilio Solano.

Seguridad en sistemas de computación

Guatemala, mayo 15 de 2025

Tarea: Seguridad en desarrollo de software

Archivo analizado: main.py

Errores identificados por SonarCloud

The screenshot displays the SonarCloud interface for the file `main.py`. It shows three detected issues:

- Issue 1:** "Specify an exception class to catch or reraise the exception". Category: Intentionality. Sub-category: Maintainability. Severity: Critical. Effort: 5min. Location: L8. Status: Open. Author: DanielMorales1103. Tags: bad-practice, error-handling, ...
- Issue 2:** "Either remove or fill this block of code.". Category: Intentionality. Sub-category: Maintainability. Severity: Major. Effort: 5min. Location: L17. Status: Open. Author: DanielMorales1103. Tag: suspicious.
- Issue 3:** "Fix this call that leads to an attribute access on a value that can be 'None'.". Category: Intentionality. Sub-category: Reliability. Severity: Major. Effort: 10min. Location: L34. Status: Open. Author: DanielMorales1103. Tag: cwe.

At the bottom, it indicates "3 of 3 shown".

Como se puede observar, SonarCloud ha marcado el análisis con un *Quality Gate* fallido. Esto debido a problemas de confiabilidad y mantenibilidad detectados en el código nuevo. De igual forma, se reportan 3 problemas o *issues*, de los cuales dos están clasificados como *Code Smells* y uno como *Bug*. A continuación, se hace énfasis sobre los problemas en cuestión y sus soluciones respectivas.

1. Captura Genérica de Excepciones (Línea 8)

Este primer error es marcado con una severidad crítica, y como tal marca que en la función `read_file` se utiliza un *except* sin especificar el tipo de excepción. Esto es considerado una mala práctica, y para solucionarlo se propone especificar la excepción y hacer uso del bloque *finally* si existe necesidad de cerrar recursos.

Corrección:

```
def read_file(file_path):  
    try:  
        with open(file_path, 'r') as file:  
            return file.read()  
    except FileNotFoundError:  
        print("File not found")  
        return None  
    except IOError:  
        print("Error reading file")  
        return None
```

2. Bloque de Código Vacío (Línea 17)

Al hacer uso de *else: pass*, esto resulta siendo innecesario, y con ello reduce la calidad del código. Este error se acredita la severidad de error mayor, y puede ser corregido eliminando ese bloque previamente mencionado para un código más limpio.

Corrección:

```
def write_file(file_path, data):  
    if data is not None:  
        with open(file_path, 'w') as file:  
            file.write(data)
```

3. Posible acceso a atributo sobre valor *None* (Línea 34)

El único *bug* detectado, ya que los anteriores eran considerados de tipo *Code Smell*. En este caso, la variable *data* en el *main* puede ser de tipo *None*, y luego pasa directamente a *process_data*, lo cual puede causar un error de ejecución al iterar sobre un *None*. Para ello, se sugiere agregar una verificación antes de llamar a *process_data*.

Corrección:

```
def main():  
    path = "example.txt"  
    data = read_file(path)  
    if data is None:  
        print("File not found")  
        return  
    processed = process_data(data)  
    print("Processed:", processed)  
    user_input = get_user_input()  
    write_file(path, user_input)
```

En conclusión, el código presenta correcciones necesarias para cumplir con estándares de confiabilidad y mantenimiento. Las observaciones detectadas por SonarCloud llevan a una mejor detección de problemas, y de esta forma corregir de manera oportuna, para así obtener un código más legible y coherente en términos de buenas prácticas y menos errores.