

REVISIÓN CÓDIGO

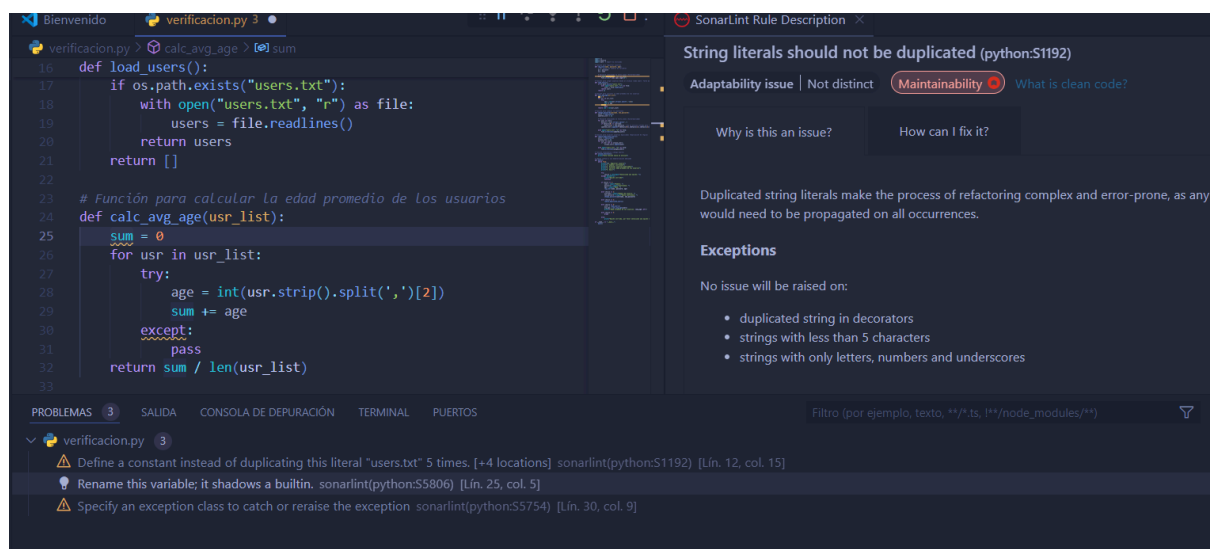
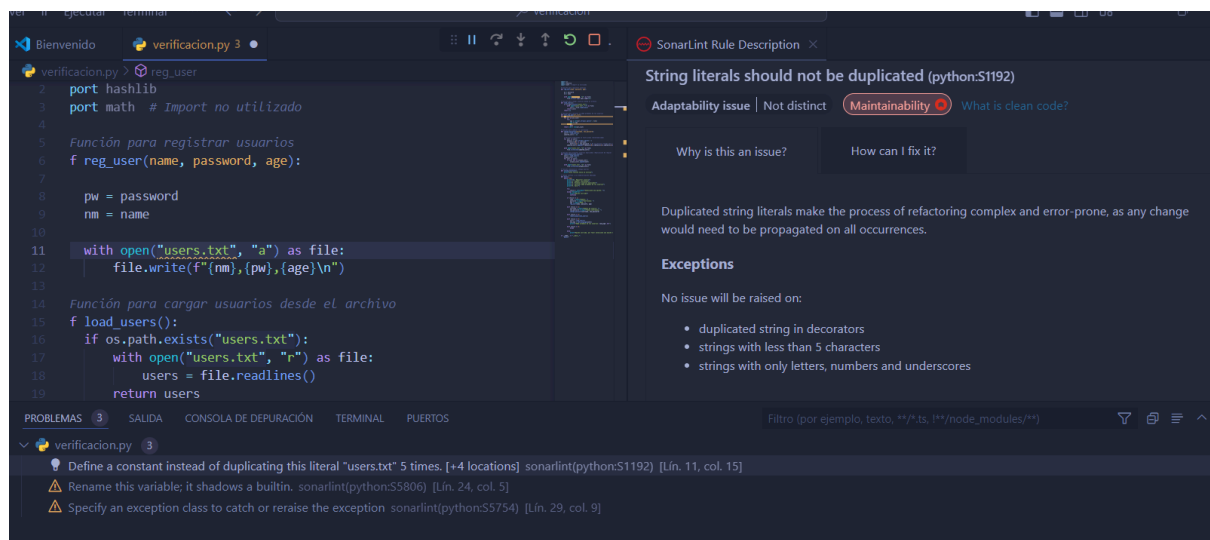
REVISIÓN CÓDIGO

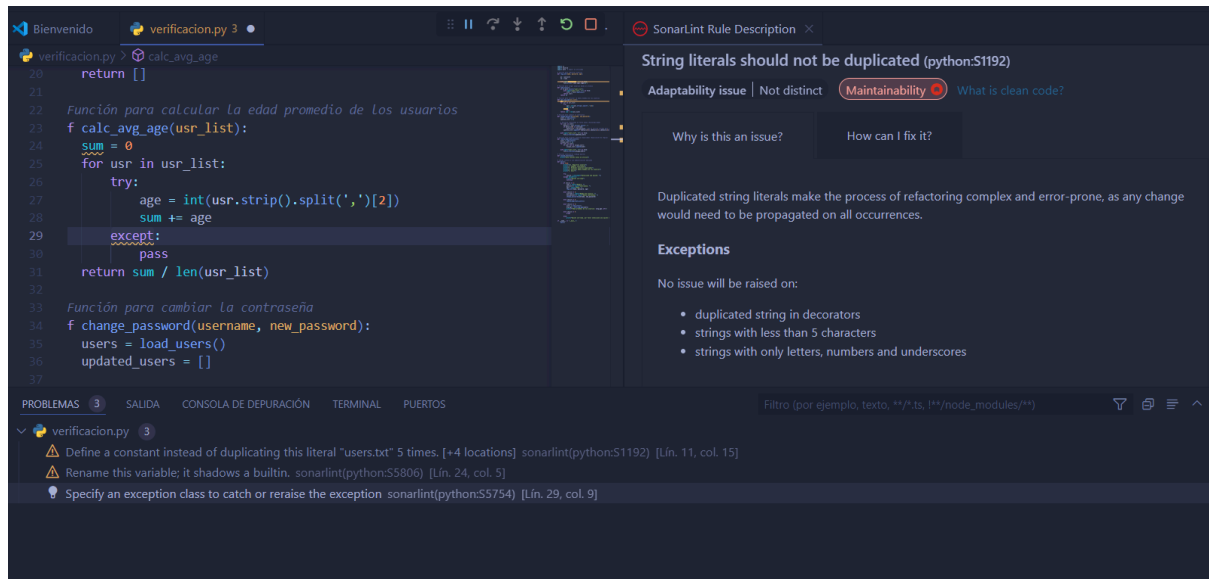
1. Alerta de SonarLint

2. 3 Correcciones

1. Violación S1192: Definir una constante en lugar de duplicar la cadena "users.txt" 2
2. Violación S5806: Renombrar la variable que oculta una incorporada 3
3. Violación S5754: Especificar una clase de excepción a capturar o relanzar la excepción 4

1.Alerta de SonarLint





2. 3 Correcciones

1. Violación S1192: Definir una constante en lugar de duplicar la cadena "users.txt"

Código Antes:

```
python
Copiar código
def read_users():
    with open("users.txt", "r") as file:
        # lógica para leer usuarios
        pass

def write_users(data):
    with open("users.txt", "a") as file:
        # lógica para escribir usuarios
        pass

# otras funciones que usan "users.txt"
```

Código Después:

```
python
```

Copiar código

```
USERS_FILE = "users.txt" # Definimos una constante
```

```
def read_users():  
    with open(USERS_FILE, "r") as file:  
        # lógica para leer usuarios  
        pass
```

```
def write_users(data):  
    with open(USERS_FILE, "a") as file:  
        # lógica para escribir usuarios  
        pass
```

```
# otras funciones que usan USERS_FILE
```

2. Violación S5806: Renombrar la variable que oculta una incorporada

Descripción: se ha nombrado una variable `list`, que es un nombre de función incorporada en Python. Se debe renombrar la variable para evitar confusiones.

Corrección: Cambiar el nombre de la variable `list` a otro que no sea un nombre incorporado.

Código Antes:

python

Copiar código

```
def process_users(user_list):  
    for user in user_list:  
        # lógica para procesar usuarios  
        pass
```

Código Después:

python

Copiar código

```
def process_users(users): # Cambiar el nombre de la variable  
    for user in users:  
        # lógica para procesar usuarios  
        pass
```

3. Violación S5754: Especificar una clase de excepción a capturar o relanzar la excepción

Descripción: Se debe especificar qué tipo de excepción quieres capturar. Capturar `Exception` o no especificar ninguna no es una buena práctica.

Corrección: Especifica la clase de excepción que quieres manejar, o si no puedes hacerlo, al menos relanza la excepción para no perder información.

Código Antes:

python

Copiar código

```
try:
    # código que puede fallar
    pass
except Exception: # No se especifica el tipo de excepción
    print("Ocurrió un error")
```

Código Después:

python

Copiar código

```
try:
    # código que puede fallar
    pass
except FileNotFoundError as e: # Capturamos un tipo específico de
    excepción
    print(f"Error: {e}. El archivo no fue encontrado.")
except Exception as e: # Capturamos cualquier otra excepción
    print(f"Ocurrió un error: {e}")
    raise # Relanzamos la excepción para no perder información
```