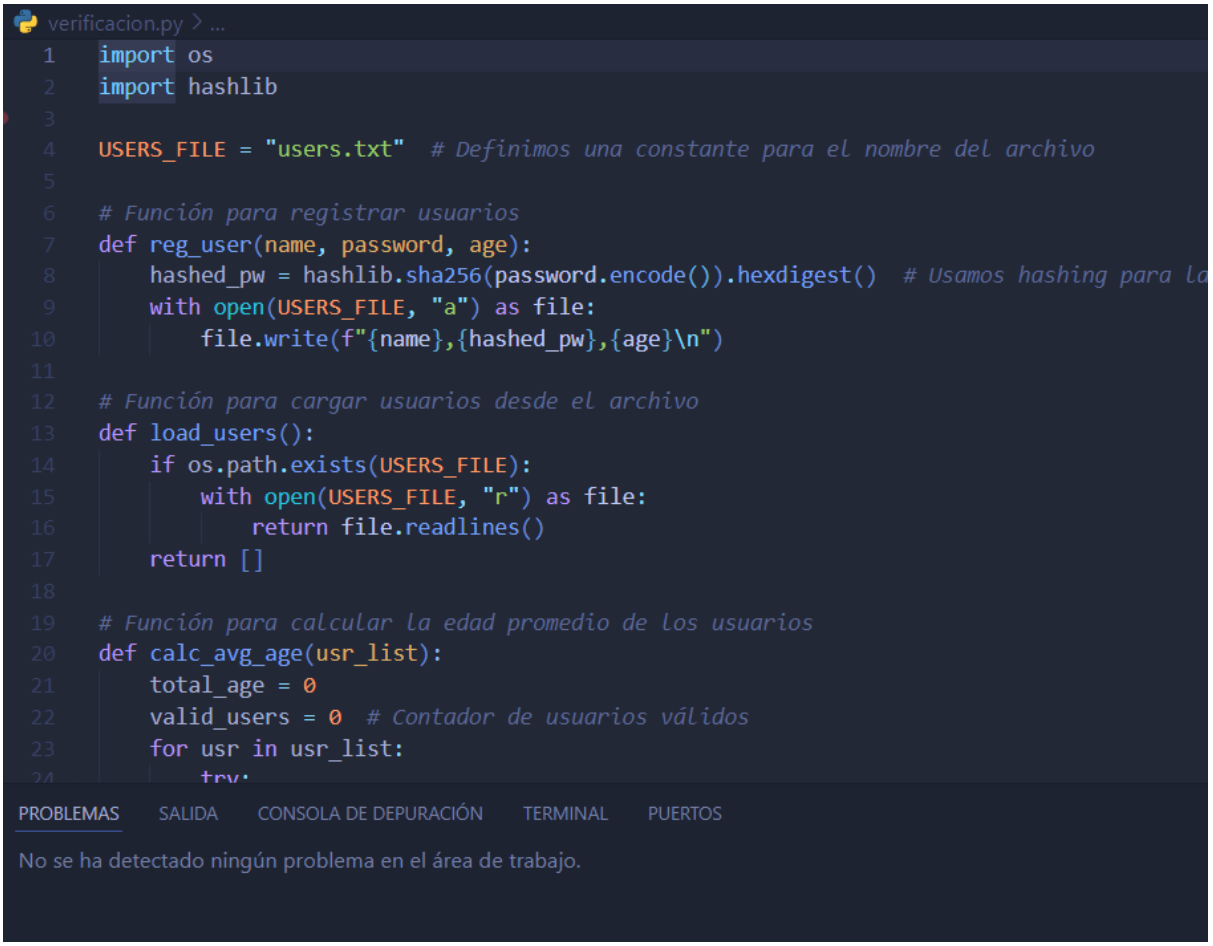
Reporte SonarLint de los archivos corregidos



1.Código corregido

```
import os
import hashlib

USERS_FILE = "users.txt"  # Definimos una constante para el nombre del archivo

# Función para registrar usuarios
def reg_user(name, password, age):
    hashed_pw = hashlib.sha256(password.encode()).hexdigest()  # Usamos hashing para la contraseña
    with open(USERS_FILE, "a") as file:
        file.write(f"{name},{hashed_pw},{age}\n")

# Función para cargar usuarios desde el archivo
def load_users():
    if os.path.exists(USERS_FILE):
        with open(USERS_FILE, "r") as file:
            return file.readlines()
    return []
```

```
# Función para calcular la edad promedio de los usuarios
def calc_avg_age(usr_list):
    total_age = 0
    valid_users = 0  # Contador de usuarios válidos
    for usr in usr_list:
        try:
            age = int(usr.strip().split(',')[2])
            total_age += age
            valid_users += 1  # Aumentamos el contador de usuarios válidos
        except (IndexError, ValueError):  # Capturamos errores específicos
            pass
    return total_age / valid_users if valid_users > 0 else 0  # Evitar división por cero

# Función para cambiar la contraseña
def change_password(username, new_password):
    users = load_users()
    updated_users = []
    hashed_pw = hashlib.sha256(new_password.encode()).hexdigest()  # Usamos hashing
para la nueva contraseña

    for user in users:
        details = user.strip().split(',')
        if details[0] == username:
            details[1] = hashed_pw  # Cambiamos la contraseña a la versión hasheada
        updated_users.append(f"{details[0]},{details[1]},{details[2]}\n")

    with open(USERS_FILE, "w") as file:
        file.writelines(updated_users)

# Función para eliminar usuarios duplicados
def remove_duplicate_users():
    users = load_users()
    unique_users = list(set(users))  # Usamos un conjunto para eliminar duplicados
    with open(USERS_FILE, "w") as file:
        file.writelines(unique_users)

def main():
    while True:
        print("\n1. Registrar usuario")
        print("2. Cambiar contraseña")
        print("3. Eliminar usuarios duplicados")
        print("4. Calcular edad promedio de los usuarios")
        print("5. Salir")

        try:
            choice = int(input("Seleccione una opción: "))
        except ValueError:
```

```
            print("Opción inválida")
            continue

        if choice == 1:
            name = input("Nombre: ")
            password = input("Contraseña: ")
            age = input("Edad: ")
            reg_user(name, password, age)

        elif choice == 2:
            username = input("Nombre de usuario: ")
            new_password = input("Nueva contraseña: ")
            change_password(username, new_password)

        elif choice == 3:
            remove_duplicate_users()

        elif choice == 4:
            users = load_users()
            avg_age = calc_avg_age(users)
            print(f"Edad promedio de los usuarios: {avg_age:.2f}")

        elif choice == 5:
            break

        else:
            print("Opción inválida, por favor seleccione una opción válida.")

if __name__ == "__main__":
    main()
```