

Relatório do Trabalho de Testes Automatizados

Disciplina: Arquitetura e Projeto de Software

Professor: Sidney Loyola de Sá

Aluno: Daniel Monteiro de Carvalho 202212193

Trabalho de Testes Automatizados em Python

https://github.com/DanielMonteiroC/testes_automatizados_em_python

Introdução

Este trabalho tem como objetivo identificar, corrigir e validar erros em métodos implementados em três classes fornecidas: StringUtils, UserManager e FibonacciGenerator. Após as correções, foram desenvolvidos testes automatizados utilizando a biblioteca unittest para garantir o funcionamento correto das implementações.

As atividades realizadas incluem:

1. Análise dos métodos para identificação de bugs.
2. Correção dos métodos conforme esperado.
3. Criação de testes automatizados para validar as implementações corrigidas.

Atividades Realizadas

1. Classe: StringUtils

- Correções:
 - Método reverse_string: Corrigido para retornar a string invertida.
 - Método is_palindrome: Validado, já estava correto.
- Testes Automatzizados:
 - Foram criados testes para validar as funcionalidades reverse_string e is_palindrome, cobrindo casos como strings vazias, palavras simples e palíndromos.

2. Classe: UserManager

- Correções:
 - Método `remove_user`: Ajustado para verificar se o usuário existe antes de tentar removê-lo, evitando erros.
- Testes

Automatizados:

 - Testes desenvolvidos para verificar:
 - Adicionar usuários únicos.
 - Prevenir duplicação de usuários.
 - Remoção de usuários existentes.
 - Tratamento de tentativa de remoção de usuários inexistentes.

3. Classe: FibonacciGenerator

- Correções:
 - Método `generate_sequence`: Ajustado para somar corretamente os dois últimos números da sequência.
 - Método `get_nth_number`: Corrigido para calcular corretamente o enésimo número de Fibonacci, removendo operações redundantes.
- Testes

Automatizados:

 - Testes cobrindo:
 - Geração de sequências completas para diferentes entradas.
 - Verificação do enésimo número para entradas válidas e inválidas.

Conclusão

O trabalho realizado permitiu não apenas corrigir os erros presentes nas implementações fornecidas, mas também desenvolver habilidades relacionadas à criação de testes automatizados, utilizando boas práticas de programação e ferramentas de teste. Todas as correções foram validadas por meio de testes abrangentes, garantindo a funcionalidade e confiabilidade dos métodos.