

Universidade de Vassouras – Campus Maricá

Disciplina: Arquitetura e Projeto de Software

Professor: Sidney Loyola

Aluno: Mateus Chagas 202212170

Introdução

Este trabalho revisita o desafio de corrigir métodos com bugs e criar testes automatizados em Python, seguindo os mesmos requisitos da versão anterior, mas com uma abordagem diferente. O objetivo é garantir funcionalidade e confiabilidade com soluções menos complexas e mais diretas.

Os arquivos fornecidos foram os seguintes:

FibonacciGenerator: Cálculo da sequência de Fibonacci.

StringUtils: Manipulação básica de strings.

UserManager: Gerenciamento de uma lista de usuários.

O processo incluiu:

Identificar os bugs.

Corrigir os métodos.

Criar testes automatizados com a biblioteca unittest para validar as implementações.

Atividades Realizadas

1. Arquivo: **FibonacciGenerator.py**

O objetivo dessa classe é gerar a sequência de Fibonacci e retornar o enésimo número da sequência. Problemas detectados:

O cálculo da sequência estava errado.

O loop para encontrar o enésimo número não cobria todas as iterações necessárias.

Em **generate_sequence**, foi utilizado um método iterativo direto para calcular os valores necessários.

Em **get_nth_number**, foi utilizado um cálculo direto para Fibonacci com um único loop.

2. Arquivo: **StringUtils.py**

A classe realiza manipulações básicas em strings, incluindo inversão e verificação de palíndromos. O método de inversão de strings não estava implementado.

Foi utilizada uma abordagem direta com o método `reversed` para garantir simplicidade e clareza.

3. Arquivo: **UserManager.py**

A classe gerencia uma lista de usuários, permitindo adicionar e remover usuários. Problema encontrado:

O método **remove_user** não tratava usuários inexistentes.

Foi implementado uma validação simples com try-except para tratar erros de remoção.

Testes Automatizados

1. Testes para FibonacciGenerator

Validação das funções de geração de sequência e obtenção do enésimo número.

2. Testes para StringUtils

Validação da inversão de strings e identificação de palíndromos.

3. Testes para UserManager

Validação da adição e remoção de usuários, incluindo casos de erro.

Conclusão

A abordagem revisada demonstrou que é possível corrigir os métodos com alternativas diretas e eficientes, mantendo a

simplicidade. Os testes automatizados garantem o funcionamento correto para diversos cenários, incluindo casos de erro.