

Universidade de Vassouras – Campus Maricá

Professora: Sidney Loyola

Engenharia de Software – 6º período

Arquitetura e Projeto de Software

Turma: B

Estudante: Hygor Rasec

**Atividade:** Relatório sobre modificações realizadas

**Arquivo:** FibonacciGenerator.py

1. Problemas Identificados

- **Método generate\_sequence:**
  - i. O cálculo do próximo número da sequência estava errado.  
(next\_number = sequence[-1] + sequence[-1] # BUG)
  - ii. Este cálculo deveria somar o último número com o penúltimo
- **Método get\_nth\_number:**
  - i. O loop estava iterando uma vez a mais do que o necessário, gerando resultados errados. (for \_ in range(n - 1): # BUG)

2. Modificações Realizadas

- **Método generate\_sequence:**
  - i. Corrigido o cálculo para:
    - 1. next\_number = sequence[-1] + sequence[-2]
- **Método get\_nth\_number:**
  - i. Ajustado o loop para iterar o número correto de vezes (baseado em índice 1):
    - 1. for \_ in range(n - 2):

**Arquivo:** StringUtils.py

1. Problemas Identificados

- **Método reverse\_string:**
  - i. Não estava revertendo a string como esperado. Ele simplesmente retornava a string original.

2. Modificações Realizadas

- **Método reverse\_string:**
  - i. Ajustado para reverter corretamente a string:
    - 1. return s[::-1]

**Arquivo:** UserManager.py

1. Problemas Identificados
  - **Método remove\_user:**
    - i. Tentava remover um usuário sem verificar se ele existia, o que poderia gerar exceções.
2. Modificações Realizadas
  - **Método remove\_user:**
    - i. Adicionada a verificação de existência antes de remover:
      1. if user in self.users:
      2. self.users.remove(user)
      3. else:
      4. return "User not found!"

## Arquivos de Teste

1. **test\_fibonacci\_generator.py:**
  - a. Testa os métodos generate\_sequence e get\_nth\_number.
  - b. Verifica se as saídas correspondem aos valores esperados para diferentes entradas.
2. **test\_string\_utils.py:**
  - a. Testa os métodos reverse\_string e is\_palindrome.
  - b. Garante que strings são revertidas corretamente e palíndromos são identificados.
3. **test\_user\_manager.py:**
  - a. Testa os métodos add\_user e remove\_user.
  - b. Valida a adição e remoção de usuários, incluindo casos de erro (usuário duplicado ou inexistente).

## Resultados Finais

1. **Funcionalidade:**
  - Todos os bugs identificados foram corrigidos.
  - Métodos estão funcionando conforme o esperado.
2. **Testes Automatizados:**
  - Todos os 6 testes foram executados com sucesso:
    - i. 6 passed in 0.03s

## Conclusão

As modificações realizadas corrigiram os erros encontrados e garantiram o funcionamento correto das classes e métodos. Os testes automatizados agora validam com eficácia cada funcionalidade, assegurando a estabilidade do código.