

Curso:	E.M.I. em Informática
Disciplina:	Engenharia de Software
Professor:	Denis Contini
Assunto:	Trabalho T1_3: Criação e Apresentação de Ambientes de Testes de Software em Python
Data de Entrega e Apresentação:	04/11/24 (Segunda-Feira)
Formato:	Duplas (Apresentação dos testes e resultados para a turma)

Criação e Apresentação de Ambientes de Testes de Software em Python

Objetivo:

A atividade tem como objetivo principal a prática da elaboração de cenários de testes de software onde os alunos deverão criar ambientes de testes de software para programas desenvolvidos anteriormente na disciplina de programação, utilizando a linguagem Python. Em duplas, os alunos irão configurar os testes unitários e funcionais, e, por fim, deverão apresentar e demonstrar a execução dos testes, explicando as técnicas utilizadas e os resultados obtidos.

Instruções da Tarefa

1. Formação das Duplas

Os alunos devem formar duplas e escolher um exercício de programação previamente desenvolvido na disciplina. Cada dupla pode escolher um exercício diferente, garantindo que todos terão projetos diversos para apresentar.

2. Requisitos do Código a Ser Testado

Código existente: O código já deve estar funcional e escrito em Python, com funções ou métodos definidos.

Desafio de testes: A função ou programa deve ter uma lógica clara, com comportamentos que possam ser testados em diferentes cenários, como, por exemplo:

- Algoritmos que manipulam listas, dicionários ou outras estruturas de dados.
- Funções matemáticas ou de manipulação de strings.
- Funções que interagem com arquivos.

3. Configuração do Ambiente de Testes

- a. **Objetivo:** As duplas devem configurar um ambiente de testes para garantir que as funções/métodos do código desenvolvido estejam funcionando conforme o esperado, cobrindo os principais casos de uso.
- b. **Ferramentas de Testes:**
 - Utilizem o módulo **unittest**, **pytest** e/ou **selenium** do Python para criar os testes unitários.
 - Organize o projeto em dois diretórios:
 - **src/**: Código original do exercício.
 - **tests/**: Arquivos contendo os testes.
- c. **Exemplo de Estrutura de Diretórios:**

```
projeto_programacao/  
├── src/                # Código principal do exercício  
│   └── exemplo.py  
└── tests/             # Diretório de testes  
    └── test_exemplo.py
```

4. Criação dos Testes

- a. **Testes Unitários:**
 - **Objetivo:** Criar testes para validar as funções/métodos do programa original. Cada teste deve cobrir um cenário específico, incluindo entradas válidas e inválidas, casos limites, e possíveis exceções.
- b. **Testes Funcionais:**
 - Caso o programa possua uma lógica mais complexa (como manipulação de arquivos ou interações entre funções), as duplas devem adicionar testes funcionais que validem a execução de todo o processo de maneira integrada.

5. Execução dos Testes

1. **Objetivo:** Executar os testes criados e garantir que o programa desenvolvido está funcionando corretamente em todos os cenários propostos.

6. Apresentação e Demonstração dos Testes

- a. **Objetivo:** As duplas devem preparar uma **apresentação prática** na qual irão demonstrar o funcionamento dos testes de software no projeto escolhido.

b. Estrutura da Apresentação:

- **Explicação do Código Original:** Descreva brevemente o programa original, sua funcionalidade e quais aspectos estão sendo testados.
- **Demonstração dos Testes:** Mostre os testes unitários escritos e explique quais cenários estão sendo cobertos (exemplos de entradas, saídas esperadas, etc.).
- **Execução ao vivo:** Execute os testes em tempo real, explicando os resultados obtidos (se todos os testes passaram ou se algum falhou e por quê).
- **Discussão sobre erros:** Caso algum teste falhe, explique a origem do problema e como ele pode ser corrigido no código.

c. Duração da apresentação: Cada dupla terá de **10 a 15 minutos** (aproximadamente) para apresentar e demonstrar os testes.

7. Critérios de Avaliação

A tarefa será avaliada com base nos seguintes critérios:

a. Criação dos Testes:

- Qualidade e variedade dos testes criados (se cobrem diferentes cenários).
- Uso correto de ferramentas de testes unitários (unittest/pytest/selenium).

b. Execução dos Testes:

- Correção na execução dos testes.
- Verificação de que todos os casos foram cobertos e funcionaram corretamente (ou explicação adequada de falhas).

c. Apresentação:

- Clareza e objetividade ao explicar o código e os testes.
- Demonstração prática eficiente dos testes.
- Capacidade de responder a perguntas ou corrigir falhas identificadas durante a execução.

d. Relatório:

- Organização e completude.
- Explicação detalhada dos testes e dos resultados.

8. Entrega

- a. **Código e testes:** Entregar o código original, os testes e o relatório no GitHub da dupla. O repositório deve estar organizado e conter todos os arquivos necessários.
 - b. **Apresentação:** A apresentação será feita em sala/laboratório, no dia 04/11/2024 (segunda-feira), com cada dupla apresentando sua solução e resultados.
-

9. Recursos Adicionais

- [Documentação do unittest \(Python\)](#)
- [Guia do pytest](#)
- [Documentação do Selenium](#)