

## ADS

### SPRINT 1 – MISSÃO 5 – Parte 1

#### PROJETO: “DEPLOYMENT QUALITY ASSURANCE”

#### ESTUDO DE CASO

Uma certa empresa decidiu estabelecer uma cultura *QUALITY ASSURANCE* em seu modelo de negócio, visando impactar positivamente processos de qualidade em suas áreas de operação e tecnologia.

#### ESCOPO DO PROJETO

O projeto será composto por 2 Sprints (AC-1 e AC-2) que se complementam, onde os alunos deverão construir ações que validem a empresa a cultura orientada a Q.A.

Os alunos desenvolverão projeto 3 em Sprints:

- **SPRINT 1: Vale até 1,25 ponto na AC-1**
- **SPRINT 2: Vale até 1,25 ponto na AC-2**

#### OBJETIVO DO PROJETO

Aprender as nuances e aplicabilidade do *QUALITY ASSURANCE* em uma organização. Construir um projeto de implementação de Gerenciamento de Qualidade Total e realizar atividades que valem nota.

#### SPRINT 1 (1,25 ponto)

Composto por 5 missões que se complementam para a entrega total do projeto:

- Missão 1: Plano do Projeto e Preparação – Vale 20% AC-1
- Missão 2: Testes Funcionais – Vale 20% AC-1
- Missão 3: Testes Não Funcionais – Vale 20% AC-1
- Missão 4: Cenários de Testes: definição e documentação – Vale 20% AC-1
- **Missão 5: Testes de Integração de automação CI/CD (parte 1) – Vale 20% AC-1**

#### MISSÃO 4 – PARTE 2

**VALE 0,25 DA NOTA AC-1**

#### OBJETIVO DA MISSÃO

#### **TESTES UNITÁRIO**

#### TAREFA 1 - PREPARAÇÃO:

1. Baixe o arquivo “**Missão5-parte1-Projeto QA ADS-5.pdf**” disponível no AVA;
2. Abra o GitHub oficial e acesse o repositório usado para o projeto de aula;

3. Anexar e commitar o arquivo “**Missão5-parte1-Projeto QA ADS-5.pdf**”;
4. No GitHub Projects abrir o cartão MISSÃO-5
5. No cartão adicione a informação: TESTES UNITARIOS CI-CD
6. Coloque também a data e os alunos presentes na aula de hoje para conferir presença.

## **TAREFA 2 - EXECUÇÃO**

7. Siga os passos abaixo para perfeita execução do CI-CD.

### Exercício: Testes com Pytest e CI com GitHub Actions

Este exercício tem como objetivo apresentar um exemplo simples de uso do framework Pytest para testes unitários em Python e a configuração de um pipeline de integração contínua (CI) utilizando GitHub Actions.

Exemplo 1: Operações matemáticas simples

1. Crie um arquivo chamado `math\_utils.py` com as seguintes funções:

```
def add(a, b):  
    return a + b
```

```
def subtract(a, b):  
    return a - b
```

2. Crie um arquivo `test\_math\_utils.py` com os testes:

```
from math_utils import add, subtract
```

```
def test_add():  
    assert add(2, 3) == 5
```

```
def test_subtract():  
    assert subtract(5, 2) == 3
```

Exemplo 2: Multiplicação com verificação de tipo

1. Altere `math\_utils.py` para adicionar a função:

```
def multiply(a, b):  
    if not isinstance(a, (int, float)) or not isinstance(b, (int, float)):  
        raise ValueError("Both arguments must be numbers")  
    return a * b
```

2. Adicione o teste correspondente em `test\_math\_utils.py`:

```
def test_multiply():  
    assert multiply(4, 3) == 12
```

```
def test_multiply_invalid_type():
```

```
try:
    multiply("a", 3)
except ValueError:
    assert True
```

### Configuração do GitHub Actions

Crie o arquivo ``.github/workflows/python-app.yml`` com o seguinte conteúdo:

```
name: Python application
```

```
on: [push, pull_request]
```

```
jobs:
```

```
  build:
```

```
    runs-on: ubuntu-latest
```

```
    steps:
```

```
      - uses: actions/checkout@v3
```

```
      - name: Set up Python
```

```
        uses: actions/setup-python@v4
```

```
        with:
```

```
          python-version: '3.10'
```

```
      - name: Install dependencies
```

```
        run: |
```

```
          python -m pip install --upgrade pip
```

```
          pip install pytest
```

```
      - name: Run tests
```

```
        run: |
```

```
          pytest
```

Com isso, ao realizar um push para o repositório, os testes serão executados automaticamente pelo GitHub Actions.

8. Agora a aula está livre para você criar todos os testes unitários e subir para o GitHub
9. Se possível, já automatize seus testes com scripts.
10. Na próxima aula vamos rodar eles no github actions

**SUCESSO A TODOS!!!**