

ADS /GTI**SPRINT 1 – MISSÃO 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.

OBJETIVO

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.

MISSÃO 1

VALE 10% DA NOTA AC-1 (0,25)

Etapa 1 - Planejamento do Projeto (GITHUB - método Kanban):

1. No GitHub acessar conta será utilizada de forma oficial para interação com o professor;
2. Abrir repositório e em seguida, criar projeto com nome “PROJETO-QA-ADS”;
3. Adicionar listas com os seguintes nomes:
 - **BACKLOG**
 - **EM ANDAMENTO**
 - **EM VALIDAÇÃO**
 - **FINALIZADO**
4. Em **BACKLOG** criar o item “MISSÃO-1”, “MISSÃO-2”, “MISSÃO-3”, “MISSÃO-4” e “MISSÃO-5”
5. Colocar “MISSÃO-2” na lista **EM EXECUÇÃO**;
6. Iniciar o processo de execução;

ETAPA 2 – EXECUÇÃO DE TESTE UNITÁRIO 1 :

O **Pytest** é um framework de testes para Python, amplamente utilizado para escrever testes automatizados de software. Oferece funcionalidades como fixtures, parametrização e relatórios.

7. Acesse o **Google Colab** (ou VSCode) e crie uma conta de uso, ou acesse com sua conta Google;
8. Acesse o repositório GitHub do Professor para pegar o código, através do endereço abaixo:

<https://github.com/PROFSANTARELLI/TESTES-QA-2025>

9. Suba os dois arquivos (**test_calculadora.py** e **calculadora.py**) para o Google Colab ou para o VSCode;
10. Código e os Testes: vamos executar uma função simples com código Python e criar testes para ela. A função será uma calculadora básica com soma, subtração, multiplicação e divisão;
11. Instale o Pytest: **pip install pytest**
12. Execute o comando: **pytest test_calculadora.py**
13. Realize os testes automatizados e colete os resultados, colocando-os no Atividade do cartão missão 2;
14. Teste 1: código com erro: após execução do Teste 1, colete os dados da primeira rodada, volte ao quadro Kanban do GitHub, abra e edite o cartão Missão 1 e copie o resultado no quadro e comite;
15. Desafio 1: faça a correção devida no Script de teste para executá-lo novamente, desta vez, sem apresentação de erros.
16. Teste 2: código corrigido: após execução do Teste 2, colete os dados da segunda rodada, volte ao quadro Kanban do GitHub, abra e edite o cartão Missão 2 e copie o resultado e comite;

ETAPA 3 – EXECUÇÃO DE TESTE UNITÁRIO 2 :

17. Faça o mesmo na operação dos arquivos:
soma_cumulativa.py e test_soma_cumulativa.py.
18. Corrija os erros encontrados

ETAPA 4 - FINALIZAÇÃO:

19. Após execução e cópia dos Testes no cartão, vocês irão pular duas linhas após a o último resultado e inserir nome completo e RA dos alunos presentes;
20. Grave (comittar) as informações e feche o cartão;
21. Certifique-se que todos os documentos e arquivos subiram para o repositório.

FIM – SUCESSO A TODOS!!!