

Aluno(a): Iza Lopes Ribeiro

Matrícula: 316

Curso: Eng. De Software – P7

## Lista de Exercícios - Teste de API (*Application Programming Interface*) com PostMan

### Instruções:

- Trabalho pode ser realizado individualmente, em duplas ou trios.
- Caso optem por não fazer individualmente é OBRIGATÓRIO que todos os participantes realizem pelo menos 1 commit relevante (do código de teste) para validar sua nota.
- Em caso de commits irrelevantes como um readme, ou arquivos de configuração, a nota do aluno será ZERADA.
- O exercício deve ser entregue na data estipulada durante a aula.
- Toda entrega de exercício solicitada deve ser feita via tarefa no Teams.
- Não deixe de contactar o professor ou monitor caso tenha **QUALQUER** dúvida.

Bons Estudos!

**Exercício 1** Utilizando todo conhecimento obtido nos exercícios anteriores, desenvolva no mínimo 6 cenários diferentes de teste para alguma *API* da sua escolha (Pelo menos 2 casos devem ser **negativos**). Atente-se para as seguintes informações:

1. **Recomenda-se** a utilização da ferramenta PostMan.
2. **Não** é permitido utilizar as *APIs* já testadas nos exercícios anteriores.
3. O exercício deve ser entregue via **GitHub**. Com um **README** adequado que forneça informações relevantes (Como executar o projeto e obter o relatório de testes).

**Exercício 2** Para todos os testes desenvolvidos anteriormente, responda e justifique.

1. Quantas suítes de testes você desenvolveu?

Resposta: Foram desenvolvidas 5 suítes de testes.

2. Os testes desenvolvidos são manuais ou automatizados?

Resposta: Os testes desenvolvidos pelo Postman são testes automatizados.

3. Onde os testes se localizam na pirâmide apresentada?

Resposta: Na pirâmide de testes, os testes fornecidos se localizam principalmente na camada de integração (GET) e na camada de sistema/end-to-end (POST, PUT, PATCH, DELETE).

4. Os testes desenvolvidos são funcionais ou não-funcionais?

Resposta: Os testes desenvolvidos são funcionais, pois verificam se as funcionalidades específicas do sistema (como criação, leitura, atualização e exclusão de objetos) estão funcionando corretamente.

5. Alguns dos testes desenvolvidos são testes Fim-a-Fim (*End-To-End*)?

Resposta: Sim, os testes de POST, PUT, PATCH e DELETE são considerados testes E2E, pois testam o sistema do início ao fim, verificando o comportamento completo das operações.

6. O que se deve fazer para que os testes desenvolvidos funcionem em modo regressão?

Resposta: Automatizar os testes, integrá-los ao pipeline de CI/CD e garantir que sejam executados regularmente após cada alteração no código para detectar regressões.