

## 1. Testes Automatizados de Carga

**Objetivo:** Avaliar a experiência com testes de performance e ferramentas de carga, como K6 ou JMeter.

**Tarefas Práticas:**

- **Tarefa 1:** Utilizando K6, e crie um teste de carga básico para uma API pública (pode ser uma API de mock). Configure para simular 500 usuários simultâneos por 5 minutos.
  - **Avaliação:** Qualidade do script, uso correto de métricas e identificação de potenciais gargalos.
- **Tarefa 2:** Gerar um relatório de teste de carga e apresentar uma análise do resultado.

## 2. Testes Automatizados de API

**Objetivo:** Avaliar a capacidade de automatizar testes de API e interpretar os resultados.

**Tarefas Práticas:**

- **Tarefa 1:** Usando ferramentas como Rest Assured, Cypress ou Playwright, crie um teste para validar endpoints de uma API de exemplo. O teste deve incluir verificações de resposta (status codes, headers e corpo).
  - **Avaliação:** Uso correto de ferramentas, clareza dos testes e cobertura de cenários importantes (testes positivos, negativos, etc.).
- **Tarefa 2:** Automatizar testes para múltiplos endpoints da API, validando diferentes métodos HTTP (GET, POST, PUT, DELETE), e realizar validações de status codes, headers e corpo de resposta para cada um dos métodos.
  - **Relatório:** Gerar e apresentar um relatório detalhado com os resultados dos testes.

## 3. Testes End-to-End (E2E)

**Objetivo:** Avaliar a experiência em automação de testes de ponta a ponta utilizando Cucumber, garantindo que os testes E2E sigam boas práticas e estejam bem documentados.

**Tarefas Práticas:**

- **Tarefa 1:** Usando Cucumber em conjunto com ferramentas como Selenium, Cypress ou Playwright, crie um teste simples que abra uma aplicação web, faça login e navegue até uma página específica (ex.: um formulário). O teste deve incluir assertivas para validar que a navegação foi bem-sucedida.
  - **Avaliação:** Estrutura e clareza dos testes, uso de boas práticas (como Page Object Pattern), tempo de execução e estabilidade.
- **Tarefa 2:** Automatizar um fluxo de checkout simples em um e-commerce de exemplo, validando a adição de produtos ao carrinho, preenchimento de formulário com os dados de pagamento e finalização da compra.
  - **Avaliação:** Complexidade do cenário, boa prática de automação e estabilidade do teste.
  - **Relatório:** Gerar e apresentar um relatório detalhado dos testes E2E.

## 4. Testes Automatizados para Aplicativos Móveis

Objetivo: Avaliar a experiência do candidato com automação de testes em aplicativos móveis usando ferramentas como Appium.

### Tarefas Práticas:

- **Tarefa 1:** Usando uma ferramenta de automação de testes para mobile (como Appium ou Espresso), crie um teste para um aplicativo de exemplo que execute o login, navegue por algumas telas e valide um elemento (como uma mensagem ou um botão).
  - **Avaliação:** Configuração do ambiente, uso correto de locators nativos, fluidez dos testes e clareza no código.
- **Tarefa 2:** Automatizar um cenário onde o usuário preenche um formulário e envia os dados em um aplicativo móvel de exemplo.
  - **Avaliação:** Validação de elementos no formulário, clareza e estabilidade dos testes.
  - **Relatório:** Gerar e apresentar um relatório detalhado com os resultados dos testes móveis.

## 5. Desafio de Integração e CI/CD

**Objetivo:** Testar o conhecimento sobre pipelines de CI/CD e a integração de testes automatizados nesses processos.

### Tarefas Práticas:

- **Tarefa 1:** Descrever ou configurar um pipeline de CI/CD (GitLab, GitHub Actions) que execute testes de API, E2E e mobile de forma automatizada após cada commit.
  - **Avaliação:** Clareza da configuração, uso eficiente de ferramentas de CI e como os testes estão integrados ao processo.
  - **Relatório:** Apresentar um relatório detalhado das execuções e dos resultados obtidos no pipeline.

### Ferramentas sugeridas para o teste prático:

- **Testes de Carga:** K6, JMeter
- **Testes de API:** Rest Assured, Cypress, Playwright
- **Testes E2E:** Cucumber com Selenium, Cypress ou Playwright
- **Testes Mobile:** Appium
- **CI/CD:** GitLab CI, GitHub Actions

### Observações

- O projeto deve ser publicado no GitHub ou GitLab.
- Salvar evidências do teste (report).
- É obrigatório ter **readme** com:
  - nome do projeto e descrição breve;
  - arquitetura/estrutura de pastas;
  - versões utilizadas;
  - dependências (como instalar);
  - como executar os testes e gerar o relatório.