



**ENGENHEIRO DE QUALIDADE DE SOFTWARE**

**MICHAEL DOUGLAS COSTA MAIA LIMA**

Análise de Qualidade

Jacobina - BA

2024

## **1. RESUMO**

Durante minha transição de carreira em março de 2023, decidi seguir a área de Tecnologia e encontrei o curso "Engenheiro de Qualidade de Software" da EBAC. Nele, adquiri conhecimentos em estratégias de teste, desenvolvimento de critérios de aceitação e casos de teste eficazes. Também explorei o uso do GitHub para versionar testes e a importância da automação no ciclo de desenvolvimento de software.

Aprendi a realizar testes de performance para avaliar a eficiência e escalabilidade dos sistemas, além de integrar testes de forma contínua e automatizada para garantir qualidade. Neste trabalho, apresento os principais insights e habilidades adquiridas ao longo do curso.

## 2. SUMÁRIO

<b>1. RESUMO .....</b>	<b>2</b>
<b>2. SUMÁRIO .....</b>	<b>3</b>
<b>3. INTRODUÇÃO .....</b>	<b>4</b>
<b>4. O PROJETO .....</b>	<b>5</b>
4.1 Estratégia de teste .....	5
4.2 Critérios de aceitação .....	6
4.3 Casos de testes .....	7
4.4 Repositório no Github .....	7
4.5 Testes automatizados .....	7
4.6 Integração contínua .....	8
4.7 Testes de performance .....	8
<b>5. CONCLUSÃO .....</b>	<b>10</b>
<b>6. REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>11</b>

### **3. INTRODUÇÃO**

O profissional de Teste de Software desempenha um papel crucial na monitoração do desenvolvimento, garantindo que os resultados atendam às expectativas. Suas responsabilidades incluem criar planos de testes, rastrear bugs e estabelecer padrões de qualidade. O planejamento é vital, funcionando como um "mapa" que orienta o projeto, permitindo monitorar atividades e gerenciar riscos.

O planejamento abrange três atividades principais: definição de cronograma de atividades, alocação de recursos e estabelecimento de marcos de projeto. A monitoração do progresso é essencial para garantir a conformidade com o plano.

O artigo se concentra no plano de teste, que define escopo, objetivos, requisitos e estratégias para os testes de software. Os testes visam identificar falhas e aumentar a confiabilidade do sistema, embora não garantam a ausência total de erros. A condução disciplinada dos testes é fundamental para maximizar a detecção de problemas.

Além disso, realizar testes envolve responder a questões sobre atributos de qualidade, responsabilidades, recursos e monitoramento. O artigo apresentará três casos práticos que aprofundam os tópicos discutidos, oferecendo uma visão abrangente do aprendizado durante o curso.

## 4. O PROJETO

Para este trabalho de conclusão de curso **Profissão: Engenheiro de Qualidade de software**, você deve utilizar o conhecimento adquirido ao longo do curso para elaborar uma estratégia de testes adequada para validar o e-commerce EBAC Shop (<http://lojaebac.ebaonline.art.br/>). Você deve considerar as histórias de usuário já refinadas como se você estivesse participando de um time ágil. As funcionalidades devem seguir todo o fluxo de trabalho de um *Quality Engineer* (QE), desde o planejamento até a entrega. Siga as etapas dos sub-tópicos para se orientar no trabalho.

### ATENÇÃO:

- Conforme a sua estratégia, você pode executar os testes no endereço disponibilizado ou utilizando as imagens disponíveis no Docker Hub:
  - Banco de Dados: [ernestosbarbosa/lojaebacdb](https://hub.docker.com/r/ernestosbarbosa/lojaebacdb)
  - Loja EBAC: [ernestosbarbosa/lojaebac](https://hub.docker.com/r/ernestosbarbosa/lojaebac)

- Comandos para subir os containers:

```
docker network create --attachable ebac-network
```

```
docker run -d --name wp_db -p 3306:3306 --network ebac-network ernestosbarbosa/lojaebacdb:latest
```

```
docker run -d --name wp -p 80:80 --network ebac-network ernestosbarbosa/lojaebac:latest
```

*Após subir os containers a loja estará em <http://localhost:80>*

### 4.1 Estratégia de teste

- Faça uma estratégia de testes em um mapa mental, seguindo algumas diretrizes como objetivos, papéis e responsabilidades, fases de testes, padrões, tipos de testes, técnicas de testes, ambientes, ferramentas, abordagem (manual ou automatizado), framework ou ferramenta usados, plataformas (web, api, mobile), etc.;
- Referência: Módulo 5
- Após fazer sua estratégia de teste, tire um print e cole aqui:



## 4.2 Critérios de aceitação

- Considere as histórias de usuário:
  - [US-0001] – Adicionar item ao carrinho
  - [US-0002] – Login na plataforma
  - [US-0003] – API de cupons
- Para cada uma delas crie pelo menos 4 critérios de aceitação usando a linguagem Gherkin;

- Crie histórias de usuário para as funcionalidades:
  - Catálogo de Produtos
  - Painel Minha Conta
  - Meus Pedidos
  - Endereços
  - Detalhes da Conta
- Referência: Módulo 8

### 4.3 Casos de testes

- Crie pelo menos 4 casos de testes para cada história de usuário, sempre que possível, usando as técnicas de testes (partição de equivalência, valor limite, tabela de decisão etc.).
- Considere sempre o caminho feliz (fluxo principal) e o caminho alternativo e negativo (fluxo alternativo). Exemplo de cenário negativo: “Ao preencher com usuário e senha inválidos deve exibir uma mensagem de alerta...”
- Identifique quais os casos de teste serão automatizados, sendo ao menos 1 caminho feliz e 1 caminho alternativo.
- Referência: Módulos 4 e 5

### 4.4 Repositório no Github

- Crie um repositório no github com o nome TCC-EBAC-QE;
- Deixe o repositório publico até a análise dos tutores;
- Neste repositório você deve subir este arquivo e todos os código fontes das automações que criar.
- Referência: Módulo 10
- Link do repositório: <https://github.com/qamichaelmaia/TCC-EBAC-QE>

### 4.5 Testes automatizados

#### 4.5.1 Automação de UI

- Crie um projeto de automação WEB com o framework e a linguagem que preferir
- Justifique a sua escolha através de um comparativo entre ao menos 3 opções de ferramentas e linguagem.
- Crie uma pasta chamada UI para os testes WEB dos casos de teste que forem automatizados
- Utilize ao menos um *Testing Pattern* (à sua escolha) na implementação dos testes.

#### 4.5.2 Automação de API

- Crie uma pasta chamada API para os testes de API dos casos de teste que forem automatizados
- Você deve utilizar a ferramenta Supertest para criar seus testes de API
- Não esqueça de validar os contratos! ☺

#### 4.5.3 Automação Mobile

- Considere para os APPs apenas a funcionalidade de Catálogo de Produtos
- Você pode encontrar os APPs em:
  - *Android*: <https://github.com/EBAC-QE/testes-mobile-ebac-shop/tree/main/app/android>
  - *iOS*: <https://github.com/EBAC-QE/testes-mobile-ebac-shop/tree/ios-tests/app/ios>
- Crie uma pasta chamada Mobile para os testes em aplicativos dos casos de teste que forem automatizados
- Utilize ao menos um *Testing Pattern* (à sua escolha) na implementação dos testes.
- Você deve implementar testes para ao menos uma das plataformas Mobile (*Android* ou *iOS*)
- Observações:
  - Considere todas as boas práticas aprendidas até aqui
  - Não esqueça de implementar a geração de relatórios
- Referência: Módulos 11, 12, 14, 16, 17, 22, 23, 24, 29 e 30

#### 4.6 Integração contínua

- Execute os testes automatizados em integração contínua utilizando o Github Actions
- Referência: Módulo 26

#### 4.7 Testes de performance

- Usando o K6, implemente um teste de performance em ao menos 2 casos de testes
- Referência: Módulo 28
- Configurações do teste de performance:
  - Usuários virtuais: 20
  - Tempo de execução: 2 minutos



-RampUp: 20 segundos

-Massa de dados: Usuário / senha:






user1\_ebac / psw!ebac@test

user2\_ebac / psw!ebac@test

user3\_ebac / psw!ebac@test

user4\_ebac / psw!ebac@test

user5\_ebac / psw!ebac@test

<input type="checkbox"/> Nome de usuário	Nome	E-mail	Função
<input type="checkbox"/>  user1_ebac	—	user1_ebac@ebac.com	Assinante
<input type="checkbox"/>  user2_ebac	—	user2_ebac@ebac.com	Assinante
<input type="checkbox"/>  user3_ebac	—	user3_ebac@ebac.com	Assinante
<input type="checkbox"/>  user4_ebac	—	user4_ebac@ebac.com	Assinante
<input type="checkbox"/>  user5_ebac	—	user5_ebac@ebac.com	Assinante

## **5. CONCLUSÃO**

Durante o curso, adquiri conhecimentos e habilidades essenciais para atuar como profissional na área de Teste de Software, focando no desenvolvimento de aplicativos web e mobile. Aprendi a importância do planejamento e da estratégia de testes para garantir a qualidade do software, compreendendo a necessidade de criar um plano de testes que inclua objetivos, papéis, fases, padrões e ferramentas.

Pratiquei a criação de casos de teste usando técnicas como partição de equivalência e valor limite, identificando quais poderiam ser automatizados em diferentes cenários. Explorei abordagens de automação para testes de interface, API e mobile, e compreendi a relevância da integração contínua para a execução automatizada de testes em ambientes colaborativos, aumentando a confiabilidade e agilidade na entrega.

Além disso, trabalhei com testes de performance usando a ferramenta K6 para avaliar a estabilidade e escalabilidade do software em condições intensas. Ao concluir o curso, não só adquiri conhecimentos teóricos, mas também apliquei esses aprendizados na prática, contribuindo para a qualidade dos projetos em que estou envolvido. Este curso foi fundamental para meu crescimento profissional, e estou confiante de que as habilidades adquiridas serão valiosas em minha carreira.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

CYPRESS. Cypress: Fast, easy testing for anything that runs in a browser. Disponível em: <https://docs.cypress.io/guides/overview/why-cypress>. Acesso em: 11 out. 2024.

CYPRESS. Cypress - JavaScript End to End Testing (2022 Series). YouTube, 2022. Disponível em: <https://www.youtube.com/watch?v=69SFwgWHUig&list=PLUDwpEzHYyLvA7QFkC1C0y0pDPqYS56iU>. Acesso em: 23 set. 2024.

EBAC. Curso Engenheiro de Qualidade de Software. Disponível em: <https://ebaonline.com.br/engenheiro-de-qualidade>. Acesso em: 13 out. 2024.

MONITORATEC. O que é qualidade de software e como avaliar seu resultado? Disponível em: <https://www.monitoratec.com.br/blog/qualidade-de-software/>. Acesso em: 08 out. 2024.

POSTMAN. Postman Docs: Introduction to Postman. Disponível em: <https://learning.postman.com/docs/introduction/overview/>. Acesso em: 15 out. 2024.

RICARDO, Vitor. Criando um Workflow AUTOMATIZADO de CI com o Github Actions. YouTube, 2023. Disponível em: <https://youtu.be/F51HlrEeedw?si=WVWdqMXIncCfNAci/>. Acesso em: 05 out. 2024.