

**ENGENHEIRO DE QUALIDADE DE SOFTWARE**

Kelvin Bobsin da Silveira

Análise de Qualidade

Cachoeirinha

2024

# RESUMO

O projeto de conclusão de curso em Engenharia de Qualidade de Software da EBAC- Escola Britânica de Artes Criativas e Tecnologia aborda a otimização dos processos de desenvolvimento de software com foco na qualidade. Utilizando abordagens de engenharia de software, gerenciamento de projetos e técnicas de teste, o projeto visa aprimorar a eficiência e a confiabilidade das aplicações desenvolvidas.

A pesquisa examina a importância da definição de requisitos claros e do uso de metodologias ágeis, como o Scrum, para garantir uma comunicação eficaz entre as equipes de desenvolvimento. Além disso, o estudo se concentra na implementação de estratégias de automação de testes e na adoção de métricas de qualidade para avaliar o desempenho do software.

O objetivo final é demonstrar como a aplicação de práticas de engenharia de qualidade de software pode resultar em produtos mais robustos, reduzindo custos e tempo de desenvolvimento. O projeto destaca a importância crescente da qualidade de software no mercado atual, onde a satisfação do cliente e a competitividade são fundamentais para o sucesso das empresas de tecnologia.

# SUMÁRIO

[1. RESUMO 2](#_Toc99483086)

[2. SUMÁRIO 3](#_Toc99483087)

[3. INTRODUÇÃO 4](#_Toc99483088)

[4. O PROJETO 5](#_Toc99483089)

[4.1 Estratégia de teste 6](#_Toc99483090)

[4.2 Critérios de aceitação 6](#_Toc99483091)

[4.3 Casos de testes 6](#_Toc99483092)

[4.4 Repositório no Github 7](#_Toc99483093)

[4.5 Testes automatizados 7](#_Toc99483094)

[4.6 Integração contínua 8](#_Toc99483095)

[4.7 Testes de performance 8](#_Toc99483096)

[5. CONCLUSÃO 9](#_Toc99483097)

[6. REFERÊNCIAS BIBLIOGRÁFICAS 9](#_Toc99483098)

# INTRODUÇÃO

A qualidade de software é uma disciplina essencial no desenvolvimento de sistemas e aplicativos que atendam às expectativas dos usuários finais em termos de desempenho, confiabilidade e funcionalidade. Neste trabalho de conclusão de curso, exploraremos o cenário desafiador da qualidade de software, com foco na validação do e-commerce EBAC Shop.

O objetivo deste trabalho é aplicar o conhecimento adquirido ao longo do curso de Engenharia de Qualidade de Software para elaborar uma estratégia de testes adequada. Isso implica considerar histórias de usuário refinadas, e seguir todas as etapas do ciclo de vida de um Engenheiro de Qualidade de Software, desde o planejamento até a entrega.

Nas etapas deste trabalho, abordamos uma abordagem abrangente para garantir a qualidade de software na plataforma de e-commerce EBAC Shop. Começamos com a criação de uma Estratégia de Teste, que define objetivos, papéis, fases de testes, padrões, tipos de testes, técnicas, ambientes e ferramentas.

Em seguida, estabelecemos Critérios de desenvolvemos novas histórias de usuário para demais funcionalidades. Posteriormente, elaboramos Casos de Testes, aplicando técnicas como partição de equivalência e cenários de teste negativos.

Os Testes Automatizados foram dividos em três partes: Automação de UI, Automação de API e Automação Mobile. Para a automação de UI utilizamos o framework Cypress, escolhido devido à sua robustez e eficiência na automação de testes de interface de usuário. Na Automação de API, empregamos a ferramenta Supertest para criação de testes de API, com validação de contratos utilizando o Cypress Para Automação Mobile, implementamos testes utilizando o WebDriverIO com Appium, cobrindo a funcionalidade de Catálogo de Produtos em plataformas Android e BrowserStack como device farm.

Executamos os testes automatizados usando o Github Actions, garantindo a qualidade em cada etapa do desenvolvimento.

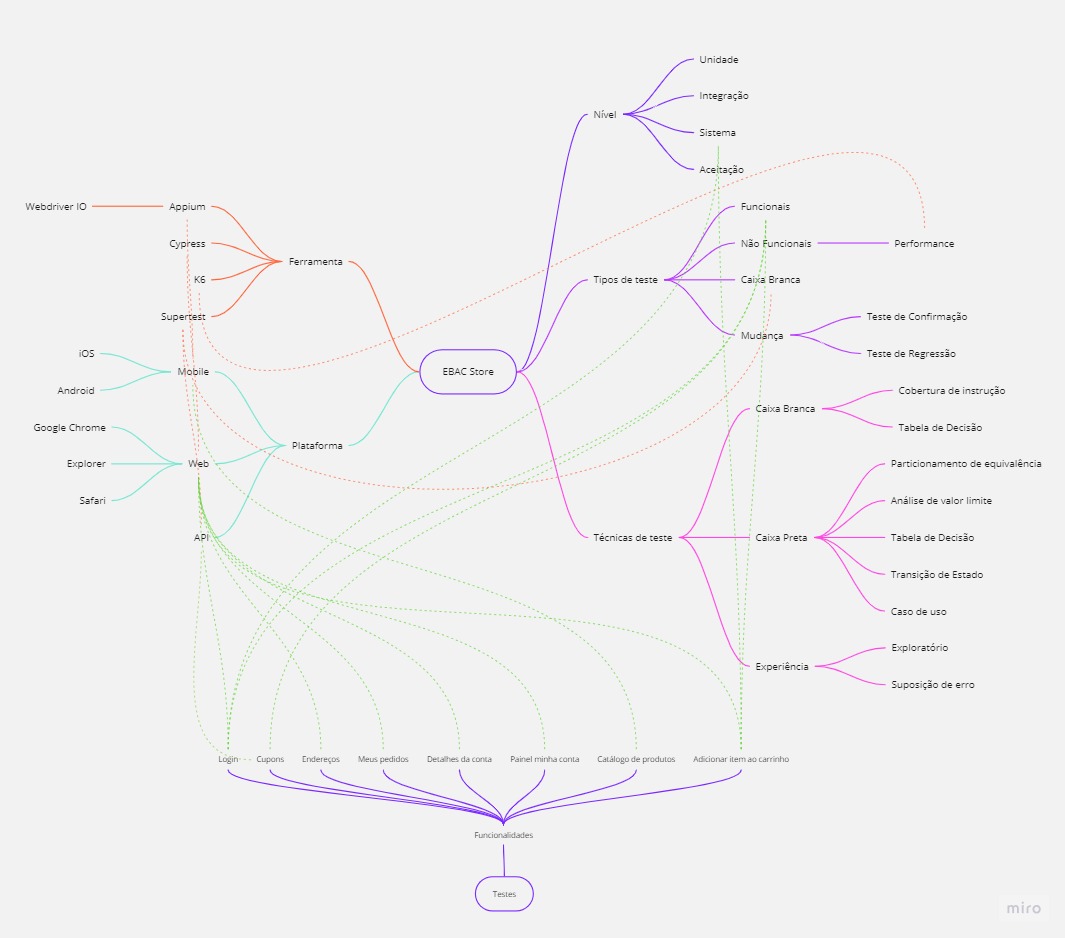
Finalmente, realizamos Testes de Performance com a ferramenta K6, simulando cenários de uso real para verificar o desempenho da API.

# 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.ebaconline.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.

## Estratégia de teste

● Faça uma estratégia de testes em um mapa mental, seguindo algumas diretrizes como objetivos, papeis 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.;



## 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

|  |  |  |  |
| --- | --- | --- | --- |
| **[US-0001] – Adicionar item ao carrinho** | | | |
| **Situação** | Em Andamento | | |
| **Projeto** | EBAC-SHOP | | |
| **Versão Afetada** | 1.0 | | |
| **Tipo:** | História | **Prioridade:** | Média |
| **Solicitante:** | Fábio Araújo | **Responsável:** |  |
| **Resolução:** |  | **Pontuação:** | 13 |
| **Tempo Estimado:** |  | | |
| **Anexos** |  | | |

**Descrição:**

**Como** cliente da EBAC-SHOP

**Quero** adicionar produtos no carrinho

**Para** realizar a compra dos itens

**Critérios de aceitação:**

1. Incluir item ao carrinho com sucesso
2. Finalizar a compra dos itens
3. Não deve permitir adicionar produtos sem estoque.
4. Limite de 10 produtos do mesmo produto no carrinho

**Critérios de aceitação - Adicionar item ao carrinho**

Funcionalidade: Adicionar item ao carrinho

Como cliente da EBAC-SHOP

Quero adicionar produtos no carrinho

Para realizar a compra dos itens

Contexto:

Dado que eu acesse a página de produtos da EBAC-SHOP

Cenário: Incluir item ao carrinho

Quando eu adicionar um produto ao carrinho

Então o produto deve ser adicionado ao carrinho

Cenário: Finalizar a compra dos itens

Quando eu adicionar produtos no carrinho

E finalizar a compra

Então deve aparecer uma mensagem “Compra efetuado com sucesso”

Cenário: Adicionar produtos sem estoque

Quando eu adiciono um produto sem estoque

Então deve exibir uma mensagem de alerta “Desculpe, mas este produto está indisponível”.

Cenário: Limite de produtos

Quando eu adicionar 11 produto ao carrinho

Então deve aparecer mensagem de alerta “Você excedeu o limite de produtos”

|  |  |  |  |
| --- | --- | --- | --- |
| **[US-0002] – Login na Plataforma** | | | |
| **Situação** | Em Andamento | | |
| **Projeto** | EBAC-SHOP | | |
| **Versão Afetada** | 1.0 | | |
| **Tipo:** | História | **Prioridade:** | Média |
| **Solicitante:** | Fábio Araújo | **Responsável:** |  |
| **Resolução:** |  | **Pontuação:** | 8 |
| **Tempo Estimado:** |  | | |
| **Anexos** |  | | |

**Critérios de aceitação:**

1. Autenticação válida
2. Autenticação inválida
3. Usuário inexistente
4. Deve bloquear acesso após 3 tentativas de acesso

**Critérios de aceitação - Login na Plataforma**

Funcionalidade: Login na plataforma

Como cliente da EBAC-SHOP

Quero fazer o login na plataforma

Para visualizar meus pedidos

Contexto:

Dado que eu acesse a página de autenticação do Portal EBAC-SHOP

Cenário: Autenticação válida

Quando eu digitar o usuário '[kelvin@ebac.com.br](mailto:kelvin@ebac.com.br)'

E a senha "senha@123"

Então deve exibir uma mensagem de boas vinda "Olá Kelvin"

Cenário: Usuário inexistente

Quando eu digitar o usuário '[xxxx@ebac.com.br](mailto:xxxx@ebac.com.br)'

E a senha 'senha@123'

Então deve exibir uma mensagem de alerta: 'Usuário inexistente'

Cenário: Usuário com senha inválida

Quando eu digitar o usuário '[kelvin@ebac.com.br](mailto:kelvin@ebac.com.br)'

E a senha 'xxx@123'

Então deve exibir uma mensagem de alerta: 'Usuário ou Senha incorreta'

Cenário: Bloqueio de usuário após excesso de tentativas

Quando eu digitar o usuário ou senha incorretos pela 3ª vez seguida

Então deve bloquear o usuário

Esquema do Cenário: autenticar multiplos usuários

Quando eu digitar <usuario>

E a <senha>

Entao deve exibir uma <mensagem> de alerta

Exemplos:

| Usuario | senha | mensagem |

| "[kelvin@ebac.com.br](mailto:kelvin@ebac.com.br)" | "xxxx123" | "Usuário inválido" |

| "[kelvin@ebac.com.br](mailto:kelvin@ebac.com.br)" | "senha@123" | "Olá Kelvin" |

| "[maria@ebac.com.br](mailto:maria@ebac.com.br)" | "senha@123" | "Olá Maria" |

| "[joao@ebac.com.br](mailto:joao@ebac.com.br)" | "senha@123" | "Olá João" |

|  |  |  |  |
| --- | --- | --- | --- |
| **[US-0001] – Adicionar item ao carrinho** | | | |
| **Situação** | Em Andamento | | |
| **Projeto** | EBAC-SHOP | | |
| **Versão Afetada** | 1.0 | | |
| **Tipo:** | História | **Prioridade:** | Média |
| **Solicitante:** | Fábio Araújo | **Responsável:** |  |
| **Resolução:** |  | **Pontuação:** | 13 |
| **Tempo Estimado:** |  | | |
| **Anexos** |  | | |

**Critérios de aceitação:**

1. Deve mostrar cupons com 10% de desconto
2. Não cadastra cupom com código inexistente
3. Não deve cadastrar cupons com código existente
4. Deve cadastrar cupons com sucesso
5. Deve expor somente cupons válidos

**Critérios de aceitação - API de cupons**

Funcionalidade: API de cupons

Como admin da EBAC-SHOP

Quero criar um serviço de cupom

Para poder listar e cadastrar os cupons

Contexto:

Dado Que acesse a API de cupons

Cenário: Deve mostrar cupons com 10% de desconto

Quando Efetuar uma pesquisa, na loja EBAC-Shop

E pesquisar por cupons 10%

Então Deve exibir uma mensagem de “Cupom EBAC10 primeira compra”

Cenário: Não cadastra cupom com código inexistente

Quando Informar os dados para cadastrar o cupom inexistente

Então Deve exibir a mensagem de “Cupom Inexistente”

Cenário: Não deve cadastrar cupons com código existente

E Informar um código já cadastrado

Então Deve exibir uma mensagem de “Cupom já existe”

Cenário: Deve cadastrar cupons com sucesso

E Informe os dados para cadastrar o cupom

Então A API deve retornar o status 201 cadastrando um novo cupom

Cenário: Deve expor somente cupons válidos

Quando eu selecionar a loja EBAC-SHOP

E Clicar em visualizar cupons

Então Deve exibir uma mensagem “Cupons válidos para hoje da loja EBAC-SHOP”

**Histórias de usuário - Catálogo de Produtos**

**Como** cliente da EBAC-SHOP

**Quero** visualizar a lista de produtos disponíveis no catálogo

**Para** que eu possa ver os itens que posso comprar

**Histórias de usuário - Painel Minha Conta**

**Como** cliente da EBAC-SHOP

**Quero** acessar a página "Minha Conta"

**Para** visualizar minhas informações pessoais

**Histórias de usuário - Meus Pedidos**

**Como** cliente da EBAC-SHOP

**Quero** acessar a página “Meus pedidos”

**Para** conferir minhas compras

**Histórias de usuário - Endereços**

**Como** cliente da EBAC-SHOP

**Quero** acessar a página de endereços

**Para** adicionar outros endereços

**Histórias de usuário - Detalhes da Conta**

**Como** cliente da EBAC-SHOP

**Quero** acessar a página de “Detalhes da conta”

**Para** atualizar meus dados

## 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.)

**Caso de teste: Adicionar item ao carrinho**

**RN01**

Não é permitido inserir mais de 10 itens de um mesmo produto ao carrinho;

|  |  |
| --- | --- |
| 0 até 10 | >10 |
| 0,1, 2,....,9,10 | 11,12, 13,...,98,100 |
| **Dados válidos** | **Dados inválidos** |

**Caso de teste**

|  |  |  |
| --- | --- | --- |
|  | **Entrada** | **Saída** |
| **Teste 1** | Inserir 11 itens de um mesmo produto ao carrinho | Inválido |
| **Teste 2** | Inserir 9 itens de um mesmo produto ao carrinho | Válido |
| **Teste 3** | Inserir 10 de um mesmo produto ao carrinho | Válido |

**RN02**

Os valores não podem ultrapassara R$990,00

|  |  |
| --- | --- |
| 0 até 990,00 | >990,00 |
| 0,10, 20, ...,100 ...,990 | 990,00,1.000,000, ... |
| **Dados válidos** | **Dados inválidos** |

**Caso de Teste**

|  |  |  |
| --- | --- | --- |
|  | **Entrada** | **Saída** |
| **Teste1** | Total do valor de R$1.000 | Inválido |
| **Teste2** | Total do valor de R$ 80 | Válido |
| **Teste3** | Total do valor de R$ 990 | Válido |

**RN03**

Valores entre R$200 e R$600, ganham cupom de 10%

|  |  |  |
| --- | --- | --- |
| <199 | 200 até 600 | >600 |
| 0, 10, 20, 100, 199 | 200, 300, 400, 500, 600 | 700, 800, ... |
| **Dados inválidos** | **Dados válidos** | **Dados inválidos** |

**Caso de Teste**

|  |  |  |
| --- | --- | --- |
|  | **Entrada** | **Saída** |
|  | Total do valor de R$ 200 | inválido |
|  | Total do valor de R$ 600 | válido |
| **Teste3** | Total do valor de R$ 150 | válido |

## Repositório no Github

Link do repositório:

## Testes automatizados

* + 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.

Considerando o projeto de conclusão de curso e levando em consideração os seguintes critérios:

- Tecnologia presente no projeto

- Testes de API

- Testes WEB

- Testes de Performance

- Testes de Segurança

- Integração CI/CD

- Facilidade de uso de massa de dados

- Facilidade em gerar relatórios e obter evidências.

- Manutenção da ferramenta e uso pela comunidade de QA.

- Facilidade de aprendizagem. A decisão tomada para a framework utilizada no projeto foi o cypress, principalmente considerando o fato de que boa parte do curso foi realizada com o cypress e o conhecimento do framework é maior em comparação com demais ferramentas.



* + 1. 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! J
  + 1. 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

## Integração contínua

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

## 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

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

# CONCLUSÃO

Ao concluir este trabalho de conclusão de curso, foi possível mergulhar profundamente na Engenharia de Qualidade de Software e aplicá-la de maneira prática. Essa experiência foi enriquecedora em diversos aspectos, proporcionando aprendizados valiosos que podem ser aplicados na vida profissional.

Uma das lições fundamentais diz respeito à elaboração de Histórias de Usuário, Critérios de Aceitação e Casos de Testes. Compreendi a importância de criar histórias de usuário detalhadas e bem definidas, juntamente com critérios de aceitação claros. Essa prática tornou a implementação de testes mais direcionada e eficaz, garantindo que os testes abordassem os requisitos essenciais do sistema.

Além disso, elaborar uma estratégia de testes sólida, considerando todos os aspectos, desde objetivos até escolhas de ferramentas, provou ser fundamental para garantir a eficácia e eficiência dos testes de qualidade.

A automação de testes também se destacou como um tópico crucial. A utilização do Cypress para automação de UI, o Supertest para testes de API e o WebDriverIO com Appium para testes mobile foram experiências valiosas. Essas ferramentas, juntamente com a compreensão das melhores práticas de automação, fortaleceram minha capacidade de criar conjuntos de testes eficazes que economizam tempo e aumentam a confiabilidade dos sistemas.

Além disso, a integração contínua e os testes de desempenho mostraram-se fundamentais para manter a qualidade ao longo do ciclo de vida do software. Enfrentei desafios técnicos e organizacionais ao implementar essa abordagem, e essas experiências me prepararam para lidar com desafios semelhantes em futuros projetos.

Os testes de desempenho com K6 proporcionaram uma compreensão mais profunda de como os sistemas se comportam sob carga e como melhorar sua escalabilidade.

Em resumo, este trabalho foi uma jornada de aprendizado enriquecedora, proporcionando não apenas conhecimento técnico, mas também habilidades práticas na área de Engenharia de Qualidade de Software. As lições aprendidas aqui moldarão minha abordagem profissional, permitindo-me contribuir de forma mais eficaz para o desenvolvimento de software de alta qualidade e atender às expectativas dos usuários finais.

# REFERÊNCIAS BIBLIOGRÁFICAS

**WebDriverIO** Disponível em: <https://webdriver.io/docs/api/>. Acesso em: 1/10/2023

**Cypress** Disponível em: <https://docs.cypress.io/guides/overview/why-cypress>. Acesso em: 1/10/2023

**K6** Disponível em: <https://k6.io/docs/>. Acesso em: 1/10/2023

**Supertest** Disponível em: <https://www.npmjs.com/package/supertest>. Acesso em: 1/10/2023