

**ENGENHEIRO DE QUALIDADE DE SOFTWARE**

Lucas Di Franco Linden

Análise de Qualidade

Taquara

2023

# RESUMO

Neste trabalho será abordado uma metodologia completa de Engenharia de Qualidade de Software, visando um processo de testes para o caso de um e-commerce, a EBAC-SHOP. O material apresentado através deste trabalho inclui a aplicação de conceitos de metodologias ágeis, mapas mentais, escrita de histórias de usuários, critérios de aceitação e casos de testes, além da aplicação de cenários de testes manuais e automatizados.

Dentro do escopo de testes automatizados, o trabalho apresenta automações de *User Interface* (UI), *Application Programming Interface* (API), automação para dispositivos móveis e para *performance*. Dentre as ferramentas utilizadas estão os *frameworks* de automação Cypress, SuperTest, Webdriver I/O e K6. Assim, este estudo oferece um completo material de referência de testes em diversas camadas de uma aplicação, bem como sobre processos de qualidade de desenvolvimento de software.

# 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

É indiscutível a importância dos processos de qualidade no desenvolvimento de software para a entrega de um produto estável e confiável. Nesse contexto, o presente trabalho apresenta uma espécie de *framework* de modelo de testes de software, aplicado a várias camadas de uma aplicação, desde a concepção dos cenários de teste até a execução dos mesmos de forma manual e automatizada, utilizando-se de tecnologias atuais e com destaque no mercado.

Desta forma, será apresentado ao leitor exemplos de histórias de usuários, critérios de aceitação, cenários de teste e também código aplicado a programação de testes automatizados, utilizando-se os *frameworks* Cypress, SuperTest, Webdriver I/O e K6, todos na linguagem JavaScript, uma linguagem moderna e em uso crescente no mundo do software. Toda a automação estará dentro de um contexto de integração contínua, através da ferramenta GitHub Actions.

Este trabalho tem como intuito expandir o entendimento sobre o tema Qualidade de Software através de exemplos práticos que serão apresentados em cada capítulo.

# O PROJETO

Ementa base do presente trabalho:

*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/*](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.;
* Referência: Módulo 5
* Após fazer sua estratégia de teste, tire um print e cole aqui:

[Imagem: Mapa mental – Estratégia de teste]

## Critérios de aceitação

Os critérios de aceitação podem ser acessados através dos links abaixo:

* [US-0001] – Adicionar item ao carrinho:

[https://github.com/LLinden/TCC-EBAC/blob/main/Crit%C3%A9rio%20de%20aceita%C3%A7%C3%A3o%20USs/%5BUS-0001%5D%20%E2%80%93%20Adicionar%20item%20ao%20carrinho.feature](https://github.com/LLinden/TCC-EBAC/blob/main/Critério de aceitação USs/%5BUS-0001%5D – Adicionar item ao carrinho.feature)

* [US-0002] – Login na plataforma [Contém cenários que serão automatizados]

[https://github.com/LLinden/TCC-EBAC/blob/main/Crit%C3%A9rio%20de%20aceita%C3%A7%C3%A3o%20USs/%5BUS-0002%5D%20%E2%80%93%20Login%20na%20plataforma.feature](https://github.com/LLinden/TCC-EBAC/blob/main/Critério de aceitação USs/%5BUS-0002%5D – Login na plataforma.feature)

* [US-0003] – API de cupons

[https://github.com/LLinden/TCC-EBAC/blob/main/Crit%C3%A9rio%20de %20aceita%C3%A7%C3%A3o%20USs/%5BUS-0003%5D %20%E2%80%93%20API%20de%20cupons.feature](https://github.com/LLinden/TCC-EBAC/blob/main/Critério de  aceitação USs/%5BUS-0003%5D  – API de cupons.feature)

As histórias de usuários de cada funcionalidade encontram-se nos links que seguem:

* Catálogo de Produtos

[https://github.com/LLinden/TCC-EBAC/blob/main/USs/US004%20-%20Cat%C3%A1logo%20de%20Produtos.odt](https://github.com/LLinden/TCC-EBAC/blob/main/USs/US004 - Catálogo de Produtos.odt)

* Painel Minha Conta

[https://github.com/LLinden/TCC-EBAC/blob/main/USs/US005%20-%20Painel%20Minha%20Conta.odt](https://github.com/LLinden/TCC-EBAC/blob/main/USs/US005 - Painel Minha Conta.odt)

* Meus Pedidos

[https://github.com/LLinden/TCC-EBAC/blob/main/USs/US006%20-%20Meus%20Pedidos.odt](https://github.com/LLinden/TCC-EBAC/blob/main/USs/US006 - Meus Pedidos.odt)

* Endereços

[https://github.com/LLinden/TCC-EBAC/blob/main/USs/US007%20-%20Endere%C3%A7os.odt](https://github.com/LLinden/TCC-EBAC/blob/main/USs/US007 - Endereços.odt)

* Detalhes da Conta

[https://github.com/LLinden/TCC-EBAC/blob/main/USs/US008%20-%20Detalhes%20da%20Conta.odt](https://github.com/LLinden/TCC-EBAC/blob/main/USs/US008 - Detalhes da Conta.odt)

## Casos de testes

As histórias de usuários de cada funcionalidade encontram-se nos links que seguem:

* Catálogo de Produtos

[https://github.com/LLinden/TCC-EBAC/blob/main/CTs/CTs%20US004%20-%20Cat%C3%A1logo%20de%20Produtos.feature](https://github.com/LLinden/TCC-EBAC/blob/main/CTs/CTs US004 - Catálogo de Produtos.feature)

* Painel Minha Conta

[https://github.com/LLinden/TCC-EBAC/blob/main/CTs/CTs%20US005%20-%20Painel%20Minha%20Conta.feature](https://github.com/LLinden/TCC-EBAC/blob/main/CTs/CTs US005 - Painel Minha Conta.feature)

* Meus Pedidos

[https://github.com/LLinden/TCC-EBAC/blob/main/CTs/CTs%20US006%20-%20Meus%20Pedidos.feature](https://github.com/LLinden/TCC-EBAC/blob/main/CTs/CTs US006 - Meus Pedidos.feature)

* Endereços

[https://github.com/LLinden/TCC-EBAC/blob/main/CTs/CTs%20US007%20-%20Endere%C3%A7os.feature](https://github.com/LLinden/TCC-EBAC/blob/main/CTs/CTs US007 - Endereços.feature)

* Detalhes da Conta

[https://github.com/LLinden/TCC-EBAC/blob/main/CTs/CTs%20US008%20-%20Detalhes%20da%20Conta.feature](https://github.com/LLinden/TCC-EBAC/blob/main/CTs/CTs US008 - Detalhes da Conta.feature)

## Repositório no Github

* Link do repositório: <https://github.com/LLinden/TCC-EBAC>

## Testes automatizados

* + 1. Automação de UI

A Automação de UI foi realizada com Cypress utilizando-se o *Testing Pattern* Page Objects Model.

Justificativa para escolha do framework Cypress para os testes de UI:

1. Características tecnológicas da aplicação: o Cypress foi desenvolvido especificamente para testes de plataformas web, contendo inclusive a possibilidade de realização de testes de componentes do Frontend, caso a equipe deseje incrementar os testes de UI no futuro.

2. Composição dos times: considerando as características do portal EBAC-SHOP supõe-se que haja um foco maior no Frontend do que Backend. Assim, seria mais fácil e seguro para os QAs a obtenção de apoio para o desenvolvimento da automação em caso de dúvidas junto aos desenvolvedores de Frontend, que possuem conhecimento em JavaScript (linguagem do Cypress);

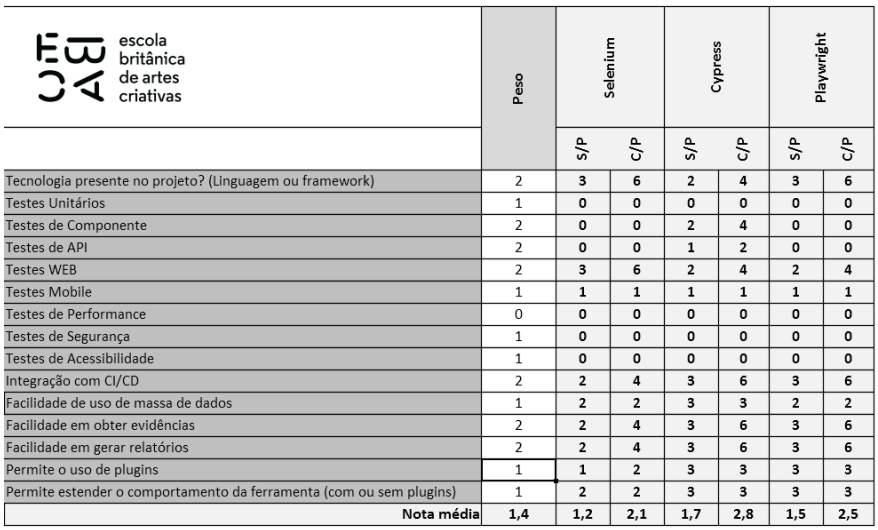
3. Disponibilidade de material de treinamento e suporte: Cypress é uma ferramenta que vem crescendo no mercado, o que significa que há uma crescente paralela de cursos, artigos e diversos materiais de apoio disponíveis online.

4. Extensões: Cypress já possui por padrão uma ferramenta de análise

(Analytics) que facilita a visualização da execução dos testes para

stakeholders e possui fácil integração com ferramentas de geração de relatórios. Além disso, pode integrar-se com o Percy, facilitando testes de regressão visual que podem, inclusive, envolver o UX Designer da equipe.

Comparativo entre outros framewoks:



Repositório dos testes de UI:

https://github.com/LLinden/TCC-EBAC/tree/main/UI

* + 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!
  + 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

Em suma, este trabalho apresentou uma Análise de Qualidade completa para o exemplo de um sistema de e-commerce, abordando metodologias ágeis, criação e execução de cenários de teste, e apresentando modelos de automação para as camadas de UI e API, além de automação para um sistema móvel e para performance. Os resultados que puderam ser obtidos através do processo aplicado, reforçam a importância de uma Análise de Qualidade no contexto de desenvolvimento de software e deixam explícitos os benefícios dessas práticas para a obtenção de um produto final com maior confiabilidade e acabamento.

Em um aspecto mais pessoal, este trabalho permitiu a aplicação de conhecimentos práticos e teóricos desenvolvidos ao longo da minha formação, e reforçou o conteúdo aprendido durante o desenvolvimento das atividades propostas.

# REFERÊNCIAS BIBLIOGRÁFICAS

Seguir regras ABNT