



engenheiro de qualidade de software

Josimari Zaghetto Fabri

Análise de Qualidade

Maringá

2024

RESUMO

Este projeto reproduz todo o percurso de aprendizagem do curso de qualidade de software, desde a criação de cenários de teste até o desenvolvimento de aplicações web e mobile, enfatizando a elaboração e execução de um plano de testes automatizados e manuais. Foram utilizadas diversas ferramentas e métodos para garantir a eficiência e qualidade dos projetos. O estudo aborda a automação de testes para interfaces de usuário, APIs e aplicativos móveis, além da integração contínua para promover qualidade e agilidade no desenvolvimento. Também inclui a realização de testes de performance, focando na estabilidade do software. O objetivo é mostrar como as práticas de teste de software são capazes de antever e controlar a qualidade dos produtos digitais, para garantir que clientes e usuários de tecnologias tenham a melhor experiência possível.

SUMÁRIO

1. RESUMO	2
2. SUMÁRIO	3
3. INTRODUÇÃO	4
4. O PROJETO	6
4.1 Estratégia de teste	6
4.2 Critérios de aceitação	6
4.3 Casos de testes	10
4.4 Repositório no Github	15
4.5 Testes automatizados	15
4.6 Integração contínua	15
4.7 Testes de performance	16
5. CONCLUSÃO	17

3. INTRODUÇÃO

No mundo de constante evolução tecnológica, a qualidade de software desempenha um papel crucial no sucesso de projetos e produtos digitais. As expectativas dos usuários em relação à funcionalidade, desempenho, acessibilidade e segurança impulsionam a necessidade da máxima qualidade no desenvolvimento e validação. A Qualidade de Software surge como uma área estratégica, responsável por garantir que as soluções tecnológicas atendam aos requisitos definidos e promovam uma experiência satisfatória tanto para os clientes quanto para o usuário final.

Neste contexto, este projeto final do curso de Engenharia de Qualidade de Software da EBAC - Escola Britânica de Artes Criativas & Tecnologia tem como objetivo demonstrar a aplicação prática dos conhecimentos adquiridos ao longo do processo de aprendizagem, utilizando ferramentas e metodologias para automação de testes, validação de requisitos e garantia de qualidade. O trabalho abordará o desenvolvimento de cenários, testes automatizados e de performance para um sistema de e-commerce, integrando práticas como testes de UI, API e dispositivos móveis.

A qualidade de software está intrinsecamente ligada ao uso de práticas de engenharia que promovam eficiência, precisão e confiabilidade no ciclo de vida do desenvolvimento. Este trabalho busca integrar essas práticas em um ambiente virtual realista, simulando cenários possíveis no dia-a-dia de desenvolvedores e profissionais da área de tecnologia.

Por estar em transição de carreira, este curso me permitiu uma visão abrangente das ferramentas mais utilizadas na área de Quality Assurance (QA), desde o início, como análise e elaboração de casos de teste até a prática de testes manuais, automatizados e acessibilidade. Também permitiu explorar o fluxo e a função de um Analista de QA no trabalho de desenvolvimento de um software, utilizando metodologias ágeis e DevOps.

Na primeira parte, há um foco na estratégia do fluxo do trabalho de ponta a ponta, partindo da elaboração de um mapa mental, onde se decide os caminhos e

as melhores ferramentas para cada aplicação até a análise e desenvolvimento dos casos de teste. Foram gerados os critérios de aceitação com a linguagem Gherkin e possíveis histórias de usuários para funcionalidades comuns em um e-commerce. Foram definidas três histórias para desenvolver as regras de negócio e os casos de teste: adicionar item no carrinho, login na plataforma e API de cupons.

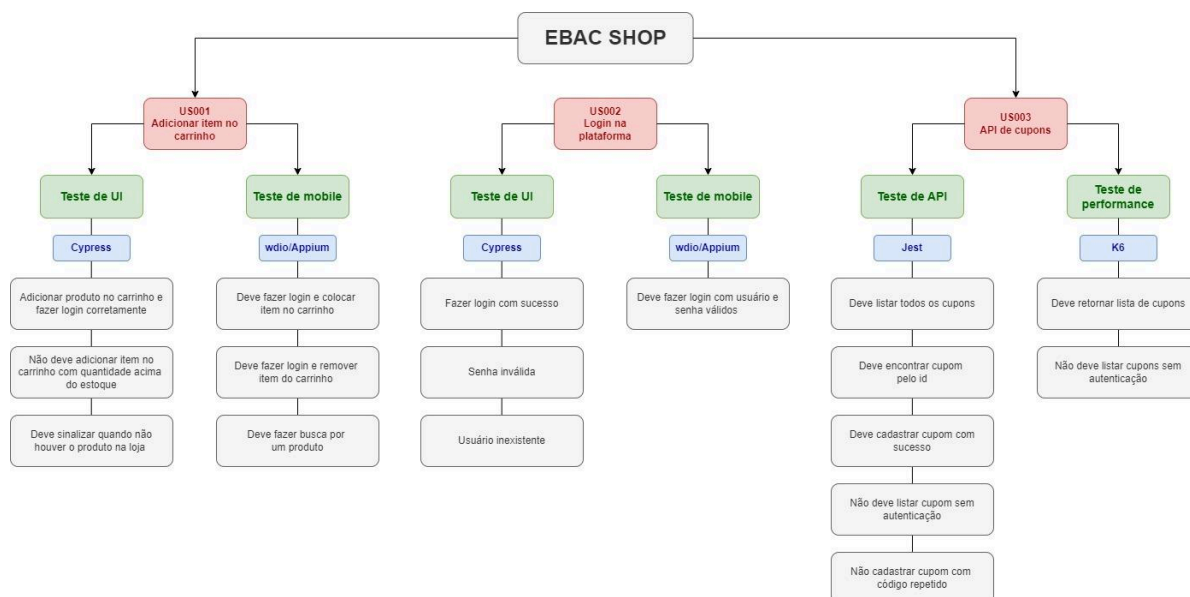
Na segunda parte, implementou-se todo o conhecimento praticado durante o curso, iniciando os testes automatizados com as ferramentas Cypress e a linguagem Javascript; em seguida, os testes automatizados de API com a ferramenta Jest; depois, automação testes de dispositivos móveis, no caso, do sistema Android, com as ferramentas Wdio e Appium. Por fim, foi aplicado um teste de performance com a ferramenta K6, seguindo as novas diretrizes da documentação fornecida pelo desenvolvedor. Também aplicação de integração contínua com o GitHub Actions.

Todos os scripts foram elaborados usando o Visual Studio Code e o Node.js para a execução da linguagem Javascript.

4. O PROJETO

Para este trabalho de conclusão de curso **Profissão: Engenheiro de Qualidade de software**, utilizou-se 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/>). Foram consideradas as histórias de usuário já pré-definidas, como se já tivessem sido refinadas por um time ágil. As funcionalidades seguiram o fluxo de trabalho de um *Quality Engineer* (QE), desde o planejamento até a entrega.

4.1 Estratégia de teste



4.2 Critérios de aceitação

Considere as histórias de usuário:

- o [US-0001] – Adicionar item ao carrinho
- o [US-0002] – Login na plataforma
- o [US-0003] – API de cupons

4.2.1 Gherkin

[US-0001] – Adicionar item ao carrinho

Funcionalidade: Adicionar item ao carrinho

Cenário: Adicionar um produto ao carrinho com sucesso

Dado que o cliente está na página de um produto

Quando clicar no botão "Adicionar ao carrinho"

E o produto deve ser exibido no carrinho

Então a quantidade deve ser atualizada corretamente

Cenário: Impedir que adicione mais de 10 itens do mesmo produto

Dado que o cliente tem 10 unidades de um produto no carrinho

Quando tentar adicionar mais uma unidade do mesmo produto

Então uma mensagem de erro "Não é possível adicionar mais de 10 itens" deve ser exibida

Cenário: Impedir que o valor total ultrapasse R\$ 990,00

Dado que o valor total no carrinho é R\$ 990,00

Quando o cliente tenta adicionar um produto que ultrapassa o limite

Então uma mensagem de erro "Valor máximo do carrinho excedido" deve ser exibida

Cenário: Aplicar cupom de desconto automaticamente baseado no valor

Dado que o cliente tem um total entre R\$ 200,00 e R\$ 600,00 no carrinho

Quando visualizar o resumo do carrinho

Então o cupom de 10% deve ser aplicado automaticamente

E o valor total deve ser atualizado com o desconto

Cenário: Aplicar cupom de desconto automaticamente baseado no valor

Dado que o cliente tem um total acima de R\$ 600,00 no carrinho

Quando visualizar o resumo do carrinho

Então o cupom de 15% deve ser aplicado automaticamente

E o valor total deve ser atualizado com o desconto

[US-0002] – Login na plataforma

Funcionalidade: Login na plataforma

Cenário: Login válido para usuário ativo

Dado que o cliente está na página de login

Quando informar <usuario> e <senha> corretamente de um usuário ativo

Então deve acessar a conta com sucesso

Cenário: Mostrar mensagem de erro para email e senhas incorretos

Dado que o cliente está na página de login

Quando informar <usuario> e <senha> incorretos

Então uma mensagem de erro "Usuário ou senha incorretos" deve ser exibida

Cenário: Bloquear login após três tentativas sem sucesso

Dado que o cliente falhou 2 vezes ao inserir a senha incorreta

Quando falhar novamente na terceira tentativa

Então o login deve ser bloqueado por 15 minutos

E uma mensagem "Sua conta foi temporariamente bloqueada" deve ser exibida

Cenário: Impedir login para usuários inativos

Dado que o cliente possui uma conta inativa

Quando tentar fazer login

Então uma mensagem "Usuário inativo" deve ser exibida

[US-0003] – API de cupons

Funcionalidade: API de cupons

Cenário: Listar todos os cupons cadastrados

Dado que o admin acessa a listagem de cupons

Quando realizar uma requisição GET para "/coupons"

Então deve retornar um status 200

E uma lista com todos os cupons cadastrados

Cenário: Listar cupom por ID

Dado que o admin acessa a listagem de cupons

Quando realizar uma requisição GET para "/coupons/\${id}" com um ID válido

Então deve retornar um status 200

E os dados do cupom correspondente devem ser exibidos

Cenário: Cadastrar um novo cupom com sucesso

Dado que o admin possui os dados necessários para criar um cupom

Quando realizar uma requisição POST para "/coupons" com os campos obrigatórios

Então deve retornar um status 201

E o cupom criado deve estar disponível na listagem de cupons


```
Cenário: Impedir cadastro de cupom com código repetido
  Dado que já existe um cupom com o código "Ganhe10"
  Quando realizar uma requisição POST para "/coupons" com o mesmo
código
  Então deve retornar um status 400
  E a mensagem "Código do cupom já existente" deve ser exibida
```

4.2.1 História do usuário

Histórias de usuário para as funcionalidades:

- Catálogo de Produtos
- Painel Minha Conta
- Meus Pedidos
- Endereços
- Detalhes da Conta

- **Catálogo de Produtos**

Como cliente da EBAC-SHOP,

Quero visualizar um catálogo de produtos,

Para escolher os itens que desejo comprar.

- **Painel Minha Conta**

Como cliente da EBAC-SHOP,

Quero acessar um painel com informações sobre minha conta,

Para visualizar meus dados pessoais.

- **Meus Pedidos**

Como cliente da EBAC-SHOP,

Quero visualizar a lista do menu “Meus Pedidos”,

Para acompanhar o status e histórico de compras.

- **Endereços**

Como cliente da EBAC-SHOP,
Quero gerenciar meus endereços de entrega,
Para facilitar as compras futuras.

- **Detalhes da Conta**

Como cliente da EBAC-SHOP,
Quero verificar meus dados pessoais,
Para manter minhas informações sempre atualizadas.

4.3 Casos de testes

História do usuário: [US-0001] – Adicionar item ao carrinho

- **RN01** - Não é permitido inserir mais de 10 itens de um mesmo produto ao carrinho;

=< 10 itens	> 10 itens
Dados válidos	Dados inválidos
0, 1, 2, ...,8, 9, 10	11, 12, 13, ...

Caso de testes:

	Entrada	Saída
Teste 1	Inserir até 10 itens iguais no carrinho	Válido
Teste 2	Inserir mais de 10 itens iguais no carrinho	Inválido

- **RN02** - Os valores não podem ultrapassar a R\$ 990,00

<= 990,00	> 990,00 itens
Dados válidos	Dados inválidos

R\$ 0,01, R\$ 500,00, R\$ 989,99

R\$ 990,01, R\$ 1.000,00...

Caso de testes:

	Entrada	Saída
Teste 1	Valor total da compra menor ou igual a R\$ 990,00	Válido
Teste 2	Valor total da compra acima de R\$ 990,00	Inválido

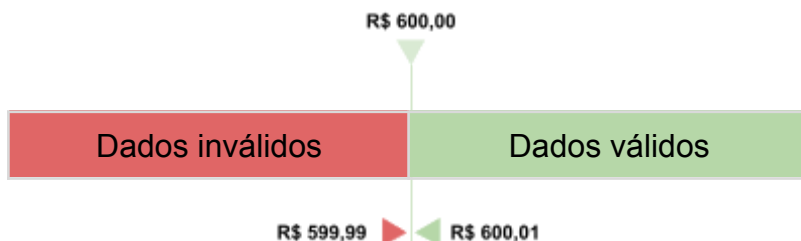
- **RN03** - Valores entre R\$ 200,00 e R\$ 600,00 ganham cupom de 10%



Caso de testes:

	Entrada	Saída
Teste 1	Compra com valor total de R\$ 199,99	Inválido
Teste 2	Compra com valor total de R\$ 200,00	Válido
Teste 3	Compra com valor total de R\$ 200,01	Válido
Teste 4	Compra com valor total de R\$ 599,99	Válido
Teste 5	Compra com valor total de R\$ 600,00	Inválido
Teste 6	Compra com valor total de R\$ 600,01	Inválido

- **RN04** - Valores acima de R\$ 600 ganham cupom de 15%



Caso de testes:

	Entrada	Saída
Teste 1	Compra com valor total de R\$ 599,99	Inválido
Teste 2	Compra com valor total de R\$ 600,00	Válido
Teste 3	Compra com valor total de R\$ 600,01	Válido

História do usuário: [US-0002] – Login na plataforma

- **RN01** - Somente usuários ativos podem fazer login;
- **RN02** - Deve exibir uma mensagem de erro caso o usuário erre o login e senha;

Condições	Regra 1	Regra 2	Regra 3	Regra 4	Regra 5	Regra 6	Regra 7	Regra 8
E-mail válido?	Sim	Sim	Sim	Sim	Não	Não	Não	Não
Senha válida?	Sim	Sim	Não	Não	Sim	Sim	Não	Não
Usuário ativo?	Sim	Não	Sim	Não	Sim	Não	Sim	Não

Usuário inativo?	Não	Sim	Não	Sim	Não	Sim	Não	Sim
Ação								
Acessar loja?	Sim	Não	Não	Não	Não	Não	Não	Não

Casos de testes:

1. Login de usuário ativo com e-mail e senha válidos, **deve** permitir acesso.
2. Login de usuário inativo com e-mail e senha válidos, **não deve** permitir acesso.
3. Login de usuário ativo com e-mail válido e senha inválida, **não deve** permitir acesso.
4. Login de usuário inativo com e-mail válido e senha inválida, **não deve** permitir acesso.
5. Login de usuário ativo com e-mail inválido e senha válida, **não deve** permitir acesso.
6. Login de usuário inativo com e-mail inválido e senha válida, **não deve** permitir acesso.
7. Login de usuário ativo com e-mail inválido e senha inválida, **não deve** permitir acesso.
8. Login de usuário inativo com e-mail inválido e senha inválida, **não deve** permitir acesso.

- **RN03** - Se o usuário errar por 3 vezes a senha, deve travar por 15 minutos o login

<= 2 tentativas	> 2 tentativas
Dados inválidos	Dados válidos

Caso de testes:

	Entrada	Saída
Teste 1	Travar login por 15 minutos após 2 tentativas	Inválido
Teste 2	Travar login por 15 minutos com 3 tentativas	Válido

História do usuário: [US-0003] – API de cupons

- **RN01** - Deve listar todos os cupons cadastrado ou listar buscando por ID do cupom

Condições	Regra 1	Regra 2	Regra 3	Regra 4
Requisição GET?	Sim	Não	Sim	Não
ID válido?	Sim	Sim	Não	Não
Ação				
Retornar cupom cadastrado?	Sim	Não	Não	Não

Casos de testes:

1. Realizou uma requisição GET com ID válido, **deve** retornar cupom cadastrado.
2. Realizou uma requisição POST com ID válido, **não deve** retornar cupom cadastrado.
3. Realizou uma requisição GET com ID inválido, **não deve** retornar cupom cadastrado.
4. Realizou uma requisição POST com ID inválido, **não deve** retornar cupom cadastrado.

- **RN02** - Deve cadastrar os cupons com os campos obrigatórios abaixo:

Condições		T1	T2	T3	T4	T5
Código		1	0	1	1	0
Valor		1	1	0	1	0
Tipo de desconto		1	1	1	0	0
Descrição		1	1	1	1	0
Ação						
Cadastro do cupom?	Sim	X				
	Não		X	X	X	X

4.4 Repositório no Github

Link do repositório: <https://github.com/JosimariFabri/TCC-EBAC-QE>

4.5 Testes automatizados

4.5.1 Automação de UI

Para a automação dos testes de UI, foi escolhida a ferramenta Cypress com a linguagem Javascript, com a testing pattern Page Objects. Pela linguagem mais simples e intuitiva e abrangência em vários requisitos na elaboração de testes, essa ferramenta se destaca de outras conhecidas do mercado, como Selenium e o Playwright. Foram desenvolvidos testes das funcionalidades do carrinho e login da aplicação supracitada. Ao final, a geração de relatórios com o Mochawesome.

4.5.2 Automação de API

Para a funcionalidade de API de cupons, foram utilizadas as ferramentas Jest com a biblioteca Supertest, onde é possível fazer as requisições da API e implementar os testes com eficiência. Os contratos foram validados com o Joi.

4.5.3 Automação Mobile

Para dispositivos móveis, foram feitos testes com o Wdio das funcionalidades como carrinho, login e pesquisa de produto, em uma aplicação para Android emulado pelo Appium.

4.5.4 Integração contínua

Na integração contínua, foram feitos com o GitHub Actions para os testes de UI.

4.5.5 Testes de performance

Por fim, foi elaborado um teste de performance de duas funcionalidades da API de cupons, com a ferramenta K6, que permite um script enxuto e funcional, em que é possível testar diversos cenários de carga e desempenho de uma aplicação.

CONCLUSÃO

Este projeto final buscou unir todas as principais ferramentas e boas práticas que um profissional de QA precisa ter. Mesmo para iniciantes, os módulos e exercícios permitiram explorar funções e conhecer as diversas camadas que formam o desenvolvimento de software. Pude desenvolver minhas habilidades de observação, planejamento e organização, além de desenvolver ainda mais meus conhecimentos de controle de qualidade que foram adquiridos com experiências passadas.

Assim sendo, uma vez que foi a porta de entrada para esta área, acredito que estou capacitada para iniciar no mercado de trabalho de tecnologia e apta a aprender cada vez mais sobre esse mundo tão rico e desafiador.