



## QUALIDADE DE SOFTWARE

Helen Cristiny Gomes Carvalho

Análise de Qualidade

Brasília

2025

## **1. RESUMO**

Este projeto aborda as práticas fundamentais de qualidade de software, explorando conceitos como estratégias de teste, critérios de aceitação, automação de testes e integração contínua. O trabalho foi desenvolvido como parte do curso de Profissão: Engenheiro de Qualidade, simulando um ambiente ágil de desenvolvimento. Foram realizadas atividades de planejamento, escrita de casos de teste e automação utilizando o Cypress para testes de UI e API. Além disso, implementou-se integração contínua com Jenkins e executaram-se testes de performance para validar a robustez do sistema. Este projeto consolidou os fundamentos da área de QA e servirá como base para o desenvolvimento profissional na área de tecnologia.

## SUMÁRIO

<b>1. RESUMO</b>	<b>2</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	5
4.2.1 História de usuário 1: [US-0001] – Adicionar item ao carrinho	5
4.2.2 História de usuário 2: [US-0002] – Login na plataforma	6
4.2.3 História de usuário 2: [US-0003] – API de cupons	6
4.3 Casos de testes	6
4.3.1 História de usuário 1:	6
4.3.2 História de usuário 2:	7
1.1.1 História de usuário 3: API de Cupom	7
1.2 Repositório no Github	7
1.3 Testes automatizados	7
1.4 Integração contínua	8
1.5 Testes de performance	8
<b>2. CONCLUSÃO</b>	<b>9</b>
<b>3. REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>9</b>

## 2. INTRODUÇÃO

O presente trabalho foi desenvolvido com o objetivo de aplicar os conceitos aprendidos ao longo do curso de Qualidade de Software, focando em práticas de mercado para garantir a entrega de um produto robusto e confiável. O projeto simula um cenário real de um time ágil, abordando desde a análise de

histórias de usuário até a automação de testes e sua integração em um pipeline de CI/CD. O sistema em questão contempla funcionalidades como login, carrinho de compras e aplicação de cupons de desconto, sendo testado tanto em nível de interface quanto de API. Espera-se que este projeto não apenas demonstre o domínio técnico sobre ferramentas e metodologias, mas também reforce a importância da qualidade no ciclo de vida do desenvolvimento de software.

### 3. O PROJETO

Para o Projeto mestre da primeira parte do curso Profissão: Engenheiro de Qualidade, você deve considerar as histórias de usuário já refinadas e como se você estivesse participando de um time ágil. As funcionalidades devem seguir todo o fluxo de trabalho de um QA, desde o planejamento até a entrega. Siga as etapas dos sub-tópicos para te orientar no trabalho. Todas as boas práticas, tanto de documentação, escrita e desenvolvimento, serão consideradas na nota. Portanto caprichem, pois além de trabalho servir como nota para o curso, vai servir como Portfólio em seu github.

#### 3.1 Estratégia de teste



### 3.2 Critérios de aceitação

#### 3.2.1 História de usuário 1: [US-0001] – Adicionar item ao carrinho

Critérios de aceitação: Adicionar item ao carrinho com sucesso

**Cenário 01:** Adicionar um item ao carrinho

**Dado** que o cliente está na página de um produto

**Quando** clicar no botão "Adicionar ao carrinho"

**Então** o item deve ser exibido no carrinho

**E** a quantidade total de itens no carrinho deve ser atualizada

Critério de aceitação 2: Restrições de quantidade e valor

**Esquema do Cenário 2:** Restringir a quantidade de itens e validar valores máximos

**Dado** que o cliente deseja adicionar um produto ao carrinho

**Quando** tentar adicionar <quantidade> itens

**Então** deve exibir a mensagem "<mensagem>"

**E** a operação deve ser bloqueada, se aplicável

Exemplos:

quantidade	mensagem	
11	Não é permitido mais de 10 itens	
5	Itens adicionados ao carrinho	
0	Não é foi incluído nenhum item	

#### 3.2.2 História de usuário 2: [US-0002] – Login na plataforma

Critérios de aceitação: Login com credenciais válidas

**Cenário 1:** Autenticação bem-sucedida

**Dado** que o cliente está na página de login

**E** possui credenciais válidas

**Quando** preencher os campos com e-mail e senha corretos

**E** clicar no botão "Entrar"

**Então** deve ser redirecionado para a página inicial

**E** deve visualizar seus pedidos

Critério de aceitação 2: Trava de segurança após erros

**Esquema do Cenário 2:** Bloquear login após múltiplas tentativas falhas

**Dado** que o cliente está na página de login

**E** tenta acessar sua conta

**Quando** inserir as credenciais incorretas <tentativas> vezes

**Então** deve exibir a mensagem "<mensagem>"

**E** o login deve ser bloqueado por 15 minutos, se aplicável

Exemplos:

tentativas	mensagem
1	Login ou senha incorretos. Tente novamente.
2	Login ou senha incorretos. Tente novamente.
3	Sua conta está temporariamente bloqueada.

### 3.2.3 História de usuário 2: [US-0003] – API de cupons

Critérios de aceitação: Aplicar cupom com sucesso

**Cenário 1:** Validar cupom de desconto

**Dado** que o cliente possui um carrinho com valor elegível para desconto

**Quando** enviar uma solicitação POST para "/aplicar-cupom"

**E** o código do cupom for válido

**Então** o valor do desconto deve ser aplicado ao total

**E** deve retornar o status 200 na resposta

## **Critério de aceitação 2:** Validação de cupons inválidos

**Cenário 2:** Esquema do Cenário: Cupom inválido ou expirado

**Dado** que o cliente tenta aplicar um cupom

**Quando** enviar uma solicitação POST para "/aplicar-cupom"

**E** o cupom for <status>

**Então** deve retornar o status <codigo\_http>

**E** a mensagem "<mensagem>"

Exemplos:

status	codigo_http	mensagem	
expirado	400	O cupom informado está expirado	
já utilizado	400	O cupom já foi utilizado	
inválido	400	O cupom informado é inválido	

## **3.3 Casos de testes**

### **3.3.1 História de usuário 1:** Adicionar item ao carrinho

**CT01.1:** Adicionar item ao carrinho com sucesso

- **Pré-condição:** Produto disponível na loja.
- **Passos:**
  - Acessar a página do produto.
  - Clicar no botão "Adicionar ao carrinho".
- **Resultado esperado:**
  - O item deve aparecer no carrinho.
  - A quantidade total de itens no carrinho deve ser atualizada.

**CT01.2:** Limite máximo de itens no carrinho

- **Pré-condição:** Estoque disponível para 10 unidades.
- **Passos:**
  - Selecionar 10 unidades do produto.
  - Adicionar os itens ao carrinho.
  - Tentar adicionar mais um item.
- **Resultado esperado:**

- Deve exibir a mensagem "Não é permitido mais de 10 itens".
- O carrinho permanece com 10 itens.

#### **CT01.3: Adicionar item indisponível**

- **Pré-condição:** Produto fora de estoque.
- **Passos:**
  - Acessar a página do produto.
  - Clicar no botão "Adicionar ao carrinho".
- **Resultado esperado:**
  - Exibe a mensagem "Produto indisponível no momento".
  - O carrinho permanece inalterado.

### **3.3.2 História de usuário 2: Login na plataforma**

#### **CT02.1: Login com credenciais válidas**

- **Pré-condição:** Conta existente com e-mail "user@example.com" e senha "password123".
- **Passos:**
  - Acessar a página de login.
  - Inserir o e-mail e senha corretos.
  - Clicar no botão "Entrar".
- **Resultado esperado:**
  - O usuário é redirecionado para a página inicial.
  - As informações do usuário estão visíveis.

#### **CT02.2: Bloqueio após tentativas incorretas**

- **Pré-condição:** Conta existente, mas senha errada.
- **Passos:**
  - Inserir e-mail correto e senha incorreta três vezes.
  - Na terceira tentativa, observar o comportamento.
- **Resultado esperado:**
  - Exibe a mensagem "Sua conta está temporariamente bloqueada".
  - O login é bloqueado por 15 minutos.

#### **CT02.3: Login com e-mail inválido**

- **Pré-condição:** Nenhuma conta registrada com o e-mail "fakeuser@example.com".
- **Passos:**
  - Inserir o e-mail "fakeuser@example.com" e uma senha qualquer.
  - Clicar no botão "Entrar".
- **Resultado esperado:**
  - Exibe a mensagem "Usuário ou senha incorretos".
  - O login não é realizado.



### 3.3.3 História de usuário 3: API de cupons

#### CT03.1: Aplicar cupom válido

- **Pré-condição:** Carrinho com valor total de R\$100 e cupom "DESCONTO10" válido.
- **Passos:**
  - Enviar uma requisição POST para "/aplicar-cupom" com o código "DESCONTO10".
- **Resultado esperado:**
  - O desconto de 10% é aplicado.
  - O valor total do carrinho é atualizado para R\$90.
  - Retorna o status HTTP 200 OK.

#### CT03.2: Cupom expirado

- **Pré-condição:** Carrinho com valor total de R\$100 e cupom "PROMO2022" expirado.
- **Passos:**
  - Enviar uma requisição POST para "/aplicar-cupom" com o código "PROMO2022".
- **Resultado esperado:**
  - Retorna o status HTTP 400 Bad Request.
  - Exibe a mensagem "O cupom informado está expirado".

#### CT03.3: Tentar aplicar cupom em carrinho vazio

- **Pré-condição:** Carrinho vazio e cupom "DESCONTO10".
- **Passos:**
  - Enviar uma requisição POST para "/aplicar-cupom" com o código "DESCONTO10".
- **Resultado esperado:**
  - Retorna o status HTTP 400 Bad Request.
  - Exibe a mensagem "Carrinho vazio. Adicione itens antes de aplicar um cupom".

## 1.1 Repositório no Github

- Link do repositório: <https://github.com/HelenCGC/TCC-EBAC.git>

## 1.2 Testes automatizados

### Automação de UI

- Crie um projeto de automação no Cypress;
- Crie uma pasta chamada UI para os testes WEB da História de Usuário [US-0001] – Adicionar item ao carrinho;

- Na automação deve adicionar pelo menos 3 produtos diferentes e validar se os itens foram adicionados com sucesso.

#### Automação de API

- Crie uma pasta chamada API para os testes de API da História de usuário “**Api de cupons**”.
- Faça a automação de **listar** os cupons e **cadastrar** cupom, seguindo as regras da História de usuário.
- Exemplo da automação de Api – GET

```
it('Deve listar todos os cupons cadastrados', () => {
  cy.request({
    method: 'GET',
    url: 'coupons',
    headers: {
      authorization: 'código_da_autorização_aqui'
    }
  }).should((response) => {
    cy.log(response)
    expect(response.status).to.equal(200)
  })
});
```

- Obs.: Considere todas as boas práticas de otimização de cenários (Page Objects, Massa de dados, Custom Commands, elementos etc.).

### 1.3 Integração contínua






- Coloque os testes automatizados na integração contínua com jenkins, criando um job para execução da sua automação;
- Compartilhe o *jenkinsfile* no repositório, junto ao seu projeto.

### 1.4 Testes de performance

- Fluxo de login da História de usuário: [US-0002] – Login na plataforma
- 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

## 2. CONCLUSÃO

A realização deste projeto representou um marco importante no aprendizado e na aplicação prática de conceitos de qualidade de software. Através do desenvolvimento e automação de casos de teste, foi possível compreender o impacto da qualidade no ciclo de vida do produto e a relevância de práticas como integração contínua e testes de performance. Além de reforçar habilidades técnicas, como o uso do Cypress e Jenkins, o trabalho permitiu experimentar o papel crítico do QA em um time ágil. Como profissional em início de carreira, este aprendizado proporciona uma base sólida para enfrentar desafios do mercado, destacando a importância da colaboração, da organização e do foco na entrega de valor ao cliente.

## 3. REFERÊNCIAS BIBLIOGRÁFICAS