

**QUALIDADE DE SOFTWARE**

Allison Cardoso da Silva

Análise de Qualidade

Belém/PA

2024

# RESUMO

Este projeto tem como objetivo garantir a qualidade e a confiabilidade do sistema EBAC-SHOP por meio de uma estratégia de testes abrangente. O foco está na implementação de testes automatizados e manuais para validar as funcionalidades essenciais da plataforma, incluindo a adição de itens ao carrinho, autenticação de usuários e a criação de um serviço de cupons.

A metodologia inclui a criação de critérios de aceitação usando a linguagem Gherkin para definir claramente os requisitos e cenários de teste, bem como a aplicação de técnicas de teste como particionamento de equivalência, valor limite e tabela de decisão. A estratégia de testes envolve diferentes fases, desde testes de unidade até testes de aceitação do usuário, garantindo que todos os aspectos do sistema sejam verificados.

Utilizamos ferramentas como Cypress para testes de interface do usuário, JMeter para testes de desempenho, Postman para testes de API e Jenkins para integração contínua. A abordagem mista de testes manuais e automatizados permite uma cobertura completa e eficiente.

Este projeto visa não apenas identificar e corrigir defeitos, mas também assegurar que o sistema atenda aos requisitos de negócio e proporcione uma experiência de usuário satisfatória.

# SUMÁRIO

[1. RESUMO 2](#_Toc85541169)

[2. SUMÁRIO 3](#_Toc85541170)

[3. INTRODUÇÃO 4](#_Toc85541171)

[4. O PROJETO 5](#_Toc85541172)

[4.1 Estratégia de teste 5](#_Toc85541173)

[4.2 Critérios de aceitação 5](#_Toc85541174)

[4.2.1 História de usuário 1: [US-0001] – Adicionar item ao carrinho 6](#_Toc85541175)

[4.2.2 História de usuário 2: [US-0002] – Login na plataforma 7](#_Toc85541176)

[4.2.3 História de usuário 2: [US-0003] – API de cupons 8](#_Toc85541177)

[4.3 Casos de testes 9](#_Toc85541178)

[4.3.1 História de usuário 1: 9](#_Toc85541179)

[4.3.2 História de usuário 2: 11](#_Toc85541180)

[4.3.1 História de usuário 3: API de Cupom 11](#_Toc85541181)

[4.4 Repositório no Github 13](#_Toc85541182)

[4.5 Testes automatizados 13](#_Toc85541183)

[4.6 Integração contínua 14](#_Toc85541184)

[4.7 Testes de performance 15](#_Toc85541185)

[5. CONCLUSÃO 17](#_Toc85541186)

[6. REFERÊNCIAS BIBLIOGRÁFICAS 17](#_Toc85541187)

# INTRODUÇÃO

A qualidade de software é uma das principais preocupações no desenvolvimento de aplicações modernas, especialmente em plataformas de e-commerce, onde a experiência do usuário e a confiabilidade do sistema são cruciais para o sucesso do negócio

Este trabalho se concentra na criação de uma estratégia de testes abrangente para o sistema EBAC-SHOP, com o objetivo de assegurar que todas as funcionalidades operem conforme esperado e que o sistema seja capaz de lidar com diferentes cenários de uso. A estratégia de testes inclui a definição de critérios de aceitação claros, o uso de técnicas de teste variadas e a aplicação de ferramentas de automação para garantir uma cobertura de testes completa e eficiente.

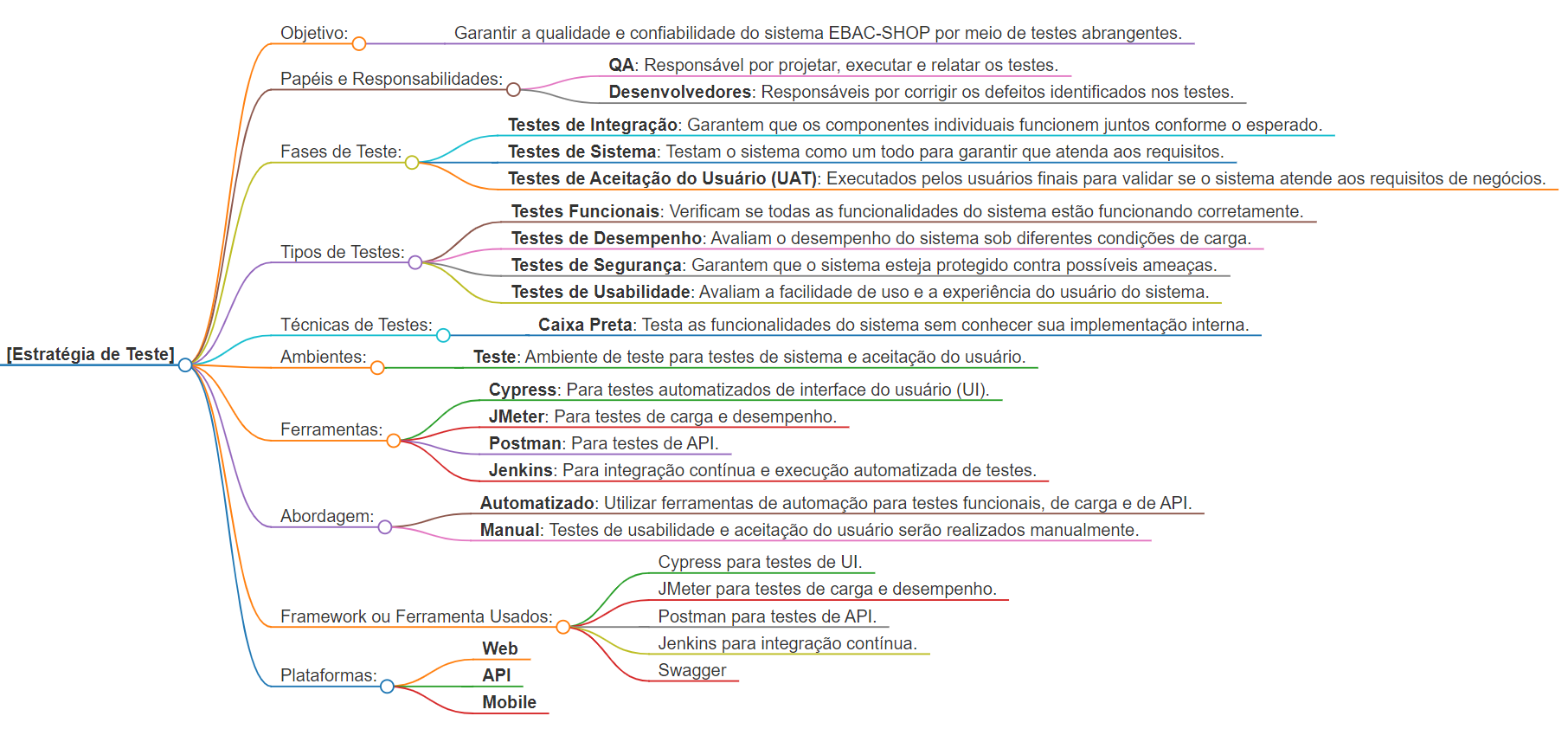
Abordaremos diferentes aspectos do processo de teste, incluindo a autenticação de usuários, a adição de itens ao carrinho e a implementação de um serviço de cupons. Utilizaremos a linguagem Gherkin para definir os critérios de aceitação, facilitando a comunicação entre as partes interessadas e a equipe de desenvolvimento.

Esperamos que, ao final deste projeto, o sistema EBAC-SHOP tenha passado por um rigoroso processo de validação, garantindo não apenas a correção dos defeitos identificados, mas também a conformidade com os requisitos de negócio e a satisfação dos usuários. O uso de ferramentas como Cypress, JMeter, Postman e Jenkins permitirá uma integração contínua e eficiente dos testes, resultando em um produto final de alta qualidade.

# O PROJETO

Para o Trabalho de Conclusão de Curso Qualidade de Software, 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.

## Estratégia de teste



## Critérios de aceitação

Considere as histórias de usuário: [US-0001] – Adicionar item ao carrinho, [US-0002] Login na plataforma e [US-0003] – API de cupons

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

**Critérios de aceitação:  
Contexto**: Dado que estou na página de um produto

**Cenário 1: Adicionar menos de 10 itens de um produto ao carrinho**

**Quando** eu adicionar 9 itens do produto ao carrinho

**Então** o sistema deve adicionar os itens ao carrinho

**E** o total de itens no carrinho deve ser 9

**Cenário 2: Tentar adicionar mais de 10 itens de um produto ao carrinho**

**Quando** eu tentar adicionar 11 itens do produto ao carrinho

**Então** o sistema deve exibir uma mensagem de erro "Não é permitido inserir mais de 10 itens de um mesmo produto ao carrinho"

**E** o total de itens no carrinho deve ser 10  
  
**Cenário 3:** Não permitir que o valor total dos itens ultrapasse R$ 990,00

**Dado** que o carrinho está vazio

**Quando** eu adicionar itens ao carrinho cujo valor total exceda R$ 990,00

**Então** o sistema deve exibir uma mensagem de erro "Os valores não podem ultrapassar R$ 990,00"

**E** o valor total no carrinho deve permanecer em R$ 990,00

**Cenário 4:** Aplicar desconto ao adicionar itens no carrinho

**Dado** que o carrinho está vazio

**Quando** eu adicionar itens ao carrinho com valor total de <valorTotal>

**Então** devo receber um desconto de <percentualDesconto>%

**E** o valor do desconto deve ser de R$ <valorDesconto>

**Exemplos**:

| valorTotal | percentualDesconto | valorDesconto |

| 250 | 10 | 25 |

| 650 | 15 | 97.5 |

**Cenário 5:** Não permitir adicionar mais de 10 itens do mesmo produto ao carrinho

**Dado** que o carrinho está vazio

**Quando** eu adicionar mais de 10 itens do mesmo produto ao carrinho

**Então** o sistema deve exibir uma mensagem de erro "Não é permitido inserir mais de 10 itens de um mesmo produto ao carrinho"

**E** a quantidade de itens desse produto no carrinho deve permanecer em 10

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

**Critérios de aceitação:  
Cenário 1:** Login com sucesso

**Dado** que sou um usuário ativo com o CPF "123.456.789-00" e senha "senha123"

**Quando** eu inserir o CPF "123.456.789-00"

**E** eu inserir a senha "senha123"

**E** eu clicar no botão "Login"

**Então** devo ser redirecionado para a página de meus pedidos

**Cenário 2:** Tentativa de login com credenciais inválidas

**Dado** que eu sou um usuário

**Quando** eu inserir o e-mail "usuario@exemplo.com"

**E** eu inserir a senha "senhaErrada"

**E** eu clicar no botão "Login"

**Então** devo ver uma mensagem de erro "Usuário ou senha inválidos"

**Esquema do Cenário**: Travar login após 3 tentativas com senha errada

**Dado** que eu sou um usuário ativo

**E** eu já errei a senha 2 vezes

**Quando** eu inserir o e-mail ou nome de usuário ou CPF "<login>"

**E** eu inserir a senha "senhaErrada"

**E** eu clicar no botão "Login"

**Então** devo ver uma mensagem de erro "Usuário ou senha inválidos"

**E** ao tentar logar novamente

**Quando** eu inserir o e-mail ou nome de usuário ou CPF "<login>"

**E** eu inserir a senha correta "senha123"

**E** eu clicar no botão "Login"

**Então** devo ver uma mensagem de erro "Conta bloqueada por 15 minutos devido a múltiplas tentativas falhas"

**Exemplos**:

| login |

| usuario@exemplo.com|

| usuario123 |

| 123.456.789-00 |

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

**Cenário1:** Cadastrar um novo cupom

**Dado** que estou autenticado na API como admin\_ebac

**Quando** eu enviar uma requisição POST para "/cupons" com os seguintes dados:

| código | desconto |

| CUPOM10 | 10 |

**Então** devo receber um código de status 201 Created

**E** o cupom "CUPOM10" deve estar cadastrado no sistema

**Cenário 2:** Tentativa de criar cupom inválido

**Dado** que estou autenticado na API como admin\_ebac

**Quando** eu enviar uma requisição POST para "/cupons" com os seguintes dados inválidos:

| código | desconto |

| 123 | 10 |

**Então** devo receber um código de status 400 Bad Request

**E** uma mensagem de erro deve ser retornada informando que o campo "código" é obrigatório

## Casos de testes

## História de usuário 1:

**CT01**: Verificar se é possível adicionar até 10 itens de um mesmo produto no carrinho.

**Caminho Feliz:**

Acessar a página do produto.

Adicionar 10 itens do produto ao carrinho.

Verificar se o carrinho aceita os 10 itens sem exibir mensagens de erro.

**Caminho Alternativo/Negativo:**

Tentar adicionar mais de 10 itens do mesmo produto e verificar se é exibida uma mensagem de erro informando que não é permitido adicionar mais de 10 itens.

**CT02**:Valor total dos itens não deve ultrapassar R$ 990,00

**Caminho Feliz:**

Acessar a página de diversos produtos.

Adicionar itens ao carrinho até que o valor total seja exatamente R$ 990,00.

Verificar se o carrinho aceita os itens sem exibir mensagens de erro.

**Caminho Alternativo/Negativo:**

Tentar adicionar itens ao carrinho que façam o valor total ultrapassar R$ 990,00 e verificar se é exibida uma mensagem de erro informando que os valores não podem ultrapassar R$ 990,00.

**CT03:** Aplicação de cupons de desconto baseado no valor total

**Descrição:** Verificar a aplicação correta de cupons de desconto baseado no valor total dos itens no carrinho.

**Caminho Feliz:**

Adicionar itens ao carrinho cujo valor total esteja entre R$ 200 e R$ 600.

Verificar se é aplicado um cupom de desconto de 10%.

Adicionar itens ao carrinho cujo valor total seja acima de R$ 600.

Verificar se é aplicado um cupom de desconto de 15%.

**Caminho Alternativo/Negativo:**

Adicionar itens ao carrinho com valores fora das faixas de desconto e verificar se nenhum desconto é aplicado.

**Particionamento de Equivalência:**

|  |  |  |
| --- | --- | --- |
| **Regras** | **Entrada** | **Saída** |
| RN01 | Adicionar 11 itens do mesmo produto | Inválida |
| RN01 | Adicionar 10 itens do mesmo produto | Válida |
| RN01 | Adicionar 1 item do mesmo produto | Válida |
| RN02 | Adicionar itens ao carrinho com valor total de R$ 1000,00 | Inválida |
| RN02 | Adicionar itens ao carrinho com valor total de R$ 990,00 | Válida |
| RN02 | Adicionar itens ao carrinho com valor total de R$ 500,00 | Válida |
| RN03 | Adicionar itens ao carrinho com valor total de R$ 250,00 | Válida |
| RN03 | Adicionar itens ao carrinho com valor total de R$ 150,00 | Inválida |
| RN04 | Adicionar itens ao carrinho com valor total de R$ 650,00 | Válida |
| RN04 | Adicionar itens ao carrinho com valor total de R$ 599,00 | Inválida |

## História de usuário 2:

**CT01:** Login com usuário ativo (Caminho Feliz)

**Dado** que eu sou um usuário ativo  
**Quando** eu faço login com minhas credenciais válidas (e-mail, nome de usuário ou CPF)  
**Então** eu devo ser autenticado com sucesso e redirecionado para a página de pedidos

**CT02**: Login com usuário inativo (Caminho Alternativo)

**Dado** que eu sou um usuário inativo  
**Quando** eu tento fazer login com minhas credenciais válidas (e-mail, nome de usuário ou CPF)  
**Então** eu devo ver uma mensagem de erro indicando que o usuário está inativo

**CT03**: Trava de login após 3 tentativas falhas (Caminho Negativo)

**Dado** que eu sou um usuário ativo  
**Quando** eu insiro uma senha incorreta 3 vezes  
**Então** meu login deve ser travado por 15 minutos  
**E** eu devo ver uma mensagem de erro "Você errou a senha 3 vezes. Tente novamente em 15 minutos."

**Valor limite:**

|  |  |  |
| --- | --- | --- |
| **Regras** | **Entrada** | **Saída** |
| RN04 | Errar a senha 2 vezes | Válida |
| RN04 | Errar a senha 3 vezes | Inválida e trava o login por 15 minutos |
| RN04 | Errar a senha 4 vezes | Inválida e trava o login por 15 minutos |

## História de usuário 3: API de Cupom

**CT01**: Cadastrar cupom com campos obrigatórios válidos

**Dado** que sou um admin autenticado na EBAC-SHOP  
**Quando** envio uma requisição POST para cadastrar um novo cupom com os campos obrigatórios válidos  
**Então** o cupom deve ser cadastrado com sucesso  
**E** a resposta deve ter o status code 201

**CT02**: Cadastrar cupom com nome repetido

**Dado** que sou um admin autenticado na EBAC-SHOP  
**Quando** envio uma requisição POST para cadastrar um novo cupom com um nome já existente  
**Então** devo receber uma mensagem de erro "Nome do cupom já existe"  
**E** a resposta deve ter o status code 400

**CT03**: Listar todos os cupons com autenticação válida

**Dado** que sou um admin autenticado na EBAC-SHOP  
**Quando** envio uma requisição GET para "/wc/v3/coupons"  
**Então** devo receber uma lista de todos os cupons cadastrados  
**E** a resposta deve ter o status code 200

**CT04**: Listar todos os cupons com autenticação inválida

**Dado** que sou um admin da EBAC-SHOP  
**Quando** envio uma requisição GET para "/wc/v3/coupons" com autenticação inválida  
**Então** devo receber uma mensagem de erro de autenticação  
**E** a resposta deve ter o status code 401

**CT05**: Cadastrar cupom com valor no limite inferior

**Dado** que sou um admin autenticado na EBAC-SHOP  
**Quando** envio uma requisição POST para cadastrar um novo cupom com valor "0.01"  
**Então** o cupom deve ser cadastrado com sucesso  
**E** a resposta deve ter o status code 201

**CT06**: Listar cupons por ID com autenticação válida

**Dado** que sou um admin autenticado na EBAC-SHOP  
**Quando** envio uma requisição GET para "/wc/v3/coupons/{id}" com um ID válido  
**Então** devo receber os detalhes do cupom com o ID especificado  
**E** a resposta deve ter o status code 200

**Tabela de Decisão:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Cenário** | **Usuário** | **Senha** | **Resultado** |
| Usuário e senha corretos | admin\_ebac | @admin!&b@c!2022 | Acesso permitido |
| Usuário correto, senha errada | admin\_ebac | errada | Mensagem de erro de autenticação |
| Usuário errado, senha correta | errado | @admin!&b@c!2022 | Mensagem de erro de autenticação |

## Repositório no Github

Crie um repositório no github com o nome TCC-EBAC;

Deixe o repositório publico até a análise dos tutores;

Neste repositório você deve subir este arquivo e todos os código fontes da automação WEB, API, Mobile, Performance e CI.

Referência: Módulo 10

Link do repositório: < https://github.com/all1soncs/TCC-EBAC/tree/main >

## 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, CustomCommands, elementos etc.).

Referência: Módulo 11, 12 e 14

## 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.

Referência: Módulo 15

## Testes de performance

Usando o Apache Jmeter, faça um teste de performance com o fluxo de loginda História de usuário: [US-0002] – Login na plataforma

Crie um template de gravação no jmeter (recording);

Use massa de dados dinâmica em arquivo CSV;

Referência: Módulo 18

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

DICA: Em uma das requisições, após a gravação, vai aparecer os parâmetros usado. Substitua esses parâmetros pela sua massa de dados, conforme aprendido em aula:



# CONCLUSÃO

Durante a realização deste projeto, foi possível adquirir uma compreensão mais profunda sobre a importância da qualidade de software e a necessidade de estratégias de teste abrangentes e eficientes. Através da definição de critérios de aceitação claros e da aplicação de técnicas de teste adequadas, pudemos garantir que o sistema EBAC-SHOP atendesse aos requisitos de negócio e oferecesse uma experiência de usuário satisfatória.

A utilização de ferramentas como Cypress, JMeter, Postman e Jenkins proporcionou uma integração contínua dos testes, permitindo uma validação rápida e eficiente das funcionalidades do sistema. Além disso, a abordagem centrada no usuário nos ensinou a priorizar os aspectos mais relevantes para os clientes, garantindo que suas necessidades fossem atendidas de forma adequada.

Este projeto também ressaltou a importância da comunicação eficaz entre as partes interessadas e a equipe de desenvolvimento, especialmente ao definir os critérios de aceitação e elaborar os cenários de teste. A linguagem Gherkin se mostrou uma ferramenta poderosa para facilitar essa comunicação e garantir uma compreensão comum das expectativas em relação ao sistema.

No geral, esta experiência foi enriquecedora e proporcionou insights valiosos que poderão ser aplicados em minha vida profissional, especialmente no que diz respeito à garantia da qualidade de software e ao desenvolvimento de estratégias de teste eficazes. Aprendi a importância de adotar uma abordagem proativa em relação à qualidade, buscando constantemente melhorias e refinamentos nos processos de desenvolvimento e teste.

# REFERÊNCIAS BIBLIOGRÁFICAS

Cypress Documentation: <https://docs.cypress.io/>

JMeter Documentation: <https://jmeter.apache.org/usermanual/index.html>

Postman Documentation: <https://learning.postman.com/docs/getting-started/introduction/>

Documentação da API da EBAC-SHOP: <http://lojaebac.ebaconline.art.br/rest-api/docs/>  
Documentação da AP: <https://serverest.dev/>