



QUALIDADE DE SOFTWARE

Márcio Edgar Dalla Torre Filho

Análise de Qualidade

Piracicaba

2025

1. RESUMO

Este trabalho tem como objetivo apresentar as principais técnicas de teste de software abordadas no curso "Teste de Software" da plataforma EBAC. A proposta é demonstrar o papel do profissional de qualidade de software, destacando sua atuação e as práticas utilizadas para garantir a confiabilidade dos sistemas por ele criado.

Ao final, o leitor terá uma visão abrangente sobre a atuação de um profissional de QA em um projeto, desde sua concepção até a conclusão, incluindo as etapas aplicáveis tanto em sistemas novos quanto em sistemas já em uso, com foco na identificação e cobertura de falhas.

2. SUMÁRIO

1. RESUMO.....	2
2. SUMÁRIO	3
3. INTRODUÇÃO	4
4. O PROJETO	5
4.1 Estratégia de teste	5
4.2 Critérios de aceitação.....	6
4.2.1 História de usuário 1: [US-0001] – Adicionar item ao carrinho	6
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	7
4.3 Casos de testes	8
4.3.1 História de usuário 1:	8
4.3.2 História de usuário 2:	9
4.3.1 História de usuário 3: API de Cupom	10
4.4 Repositório no Github	11
4.5 Testes automatizados	11
4.6 Integração contínua	12
4.7 Testes de performance.....	13
5. CONCLUSÃO	14
6. REFERÊNCIAS BIBLIOGRÁFICAS.....	14

3. INTRODUÇÃO

A qualidade de software é um dos pilares fundamentais para o sucesso do desenvolvimento de um software. Com o aumento da complexidade dos projetos e a demanda por soluções confiáveis, o papel do profissional de testes tornou-se essencial.

“No contexto de desenvolvimento de software, qualidade pode ser entendida como um conjunto de características a serem satisfeitas, de modo que o produto de software atenda às necessidades de seus usuários.”(Lenildo,2010, Item 6)

Este trabalho tem por finalidade mostrar as etapas de uma gestão de testes desde estratégias de teste, criterios de aceitação, casos de teste, testes automatizados.

Será levado em consideração três cenários para a realização deste trabalho, Adicionar Item no carrinho, Login na plataforma e API de Cupons.

Este trabalho tem como objetivo explorar as técnicas de teste de software estudadas no curso da EBAC, contextualizando sua aplicação prática. Serão abordadas as etapas envolvidas no processo de testes, desde o planejamento até a execução, com foco na prevenção e identificação de falhas.

4. O PROJETO

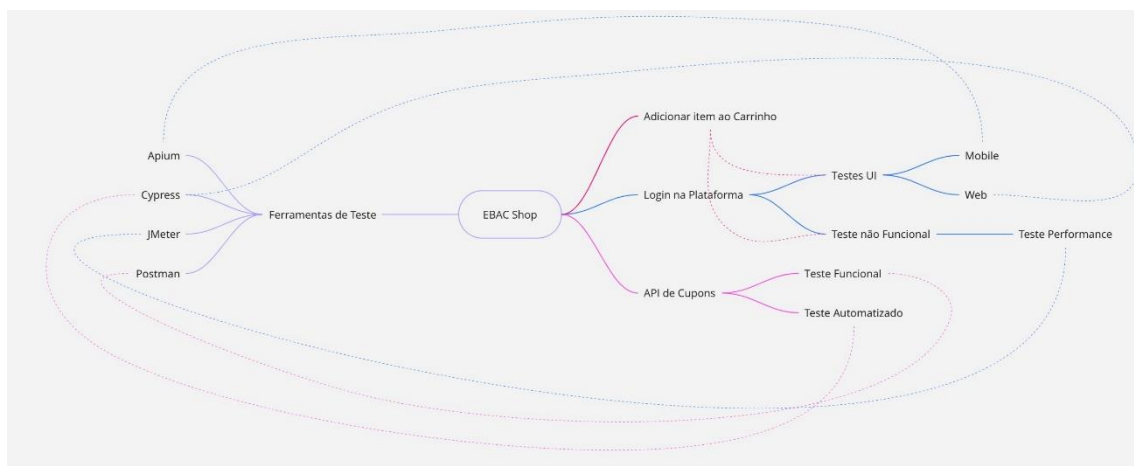
Para o Projeto mestre da primeira parte do curso Profissão: Engenheiro de Qualidade, foram considerados 3 histórias de usuário e o projeto foi feito baseado nestas histórias de usuário.

As funcionalidades são as seguidas no fluxo de trabalho de um QA, desde o planejamento até a entrega, isso irá incluir Estratégias de teste, Critérios de aceitação, Casos de testes, Testes automatizados, Integração contínua e teste de performance.

O projeto no final será postado no GutHub e ficará disponível para download.

4.1 Estratégia de teste

- Faça uma estratégia de testes em um mapa mental, seguindo algumas diretrizes como objetivos, papéis 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



4.2 Critérios de aceitação

Abaixo estão alguns criterios de aceitação com base em 3 historias de usuario, para cada uma das historias foi criado 2 critérios de aceitação usando a linguagem Gherkin.

Na historia [US-0002], foi usado tabela de exemplos (Esquema do Cenário / Scenario Outline);

Referência: Módulo 8

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

Critérios de aceitação:

Como cliente da EBAC-SHOP

Quero adicionar produtos no carrinho

Para realizar a compra dos itens

Contexto:

Dado que o cliente adicione um item no carrinho

Cenário 1: Valor limite

Quando O valor total dos itens ultrapassar R\$ 990,00

Então exibir um alerta de valor limite atingido

E não inserir o item no carrinho

Cenário 2: Incluir mais de 10 itens iguais

Quando a quantidade do mesmo item ultrapassar 10 unidades

Então exibir um alerta de quantidade limite do item atingida

E não adicionar o item ao carrinho

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

Critérios de aceitação:

Como cliente da EBAC-SHOP

Quero fazer o login (autenticação) na plataforma

Para visualizar meus pedidos

Cenário 1: Login efetuado com sucesso

Dado que o usuario possui um login ativo no sistema

E esta na tela de login

Quando digitar o login usando o usuario <usuario>

E digitar a senha <senha>

Então deve-se exibir uma <mensagem>

E deve ser redirecionado para a tela de menu da plataforma

usuario	senha	mensagem
marcio@ebac.com	1234	"Seja bem vindo Marcio"
Maria.Silva	3578	"Seja bem vindo Maria"
422.345.000-35	3214	"Seja bem vindo José"

Cenário 2: Login Incorreto

Dado que o usuário esta na tela de login do sistema

E já tenha errado mais de 3 vezes o login

Quando digitar o usuario e senha

E clicar para efetuar o login

Então exibir uma mensagem de erro "Login bloqueado por 15min"

E não efetuar o login

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

Critérios de aceitação:

Como admin da EBAC-SHOP

Quero criar um serviço de cupom

Para poder listar e cadastrar os cupons

Contexto:

Dado que o admin tenha um código de autenticação válido

Cenário 1: Realizar uma requisição GET

Quando o admin realizar uma requisição GET

Então deve retornar uma lista com todos os cupons cadastrados

E o status da operação deve ser 200

Cenário 2: Cadastrar um novo cupom com sucesso

Quando o admin realizar uma requisição POST com todos os campos obrigatórios preenchidos e válidos

Então o cupom deve ser criado com sucesso

E o status da operação deve ser 201

4.3 Casos de testes

- Crie pelo menos 3 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.).
- Considere sempre o caminho feliz (fluxo principal) e o caminho alternativo e negativo (fluxo alternativo). Exemplo de cenário negativo: “Ao preencher com usuário e senha inválidos deve exibir uma mensagem de alerta...”
- Referência: Módulo 4 e 5

4.3.1 História de usuário 1: Adicionar item ao carrinho

RN01: Não é permitido inserir mais de 10 itens de um mesmo produto ao carrinho

≤ 10	>10
Válido	Inválido
0,1,2,...,8,9,10	11,12,13,14,15...

CT01: Cadastrar Itens

Teste	Entrada	Saída
01	Inserir 3 itens iguais no carrinho	Válido
02	Inserir 10 itens iguais no carrinho	Válido
03	Inserir 11 itens iguais no carrinho	Inválido
04	Inserir 99 itens iguais no carrinho	Inválido

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

990	
Válido	Inválido
899,99	990,01

CT02: Valor Limite

Teste	Entrada	Saída
01	Valor total do carrinho R\$ 899,99	Válido
02	Valor total do carrinho R\$ 10,00	Válido
03	Valor total do carrinho R\$ 990,01	Inválido
04	Valor total do carrinho R\$ 9990,00	Inválido

RN03: Valores entre R\$ 200 e R\$ 600 , ganham cupom de 10%

Condições	Regra 1	Regra 2	Regra 3
Valor maior que R\$ 200	Não	Sim	Sim
Valor menor ou igual a R\$ 600	Sim	Sim	Sim
Ações			
Permitir desconto de 10%	Não	Sim	Sim

CT03: Cupom de 10% de desconto

	Entrada	Saída
Regra1	Valor total do carrinho R\$ 99,00	Inválido
Regra2	Valor total do carrinho R\$ 500,00	Válido
Regra3	Valor total do carrinho R\$ 600,00	Válido

4.3.2 História de usuário 2: Login na plataforma

RN01: Somente usuários ativos e desbloqueados podem fazer login

Condições	Regra 1	Regra 2	Regra 3
Usuário Ativo	Sim	Sim	Não
Usuário Desbloqueado	Sim	Não	Sim
Ações			
Permitir Login	Sim	Não	Não

CT01: Login no sistema

Teste	Entrada	Saída
01	Usuário ativo e desbloqueado	Válido
02	Usuário ativo e bloqueado	Inválido
03	Usuário inativo e desbloqueado	Inválido

RN02: Deve exibir uma mensagem de erro caso o usuário erre o login e senha

Condições	Regra 1	Regra 2	Regra 3
Usuário válido	Sim	Sim	Não
Senha válida	Sim	Não	Não
Ações			
Mensagem de erro	Sim	Não	Não

CT02: Mensagem de “usuário ou senha inválida”

	Entrada	Saída
Regra 01	Usuário válido e senha válida	Válido
Regra 02	Usuário válido e senha inválida	Inválido
Regra 03	Usuário inválido e senha inválida	Inválido

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

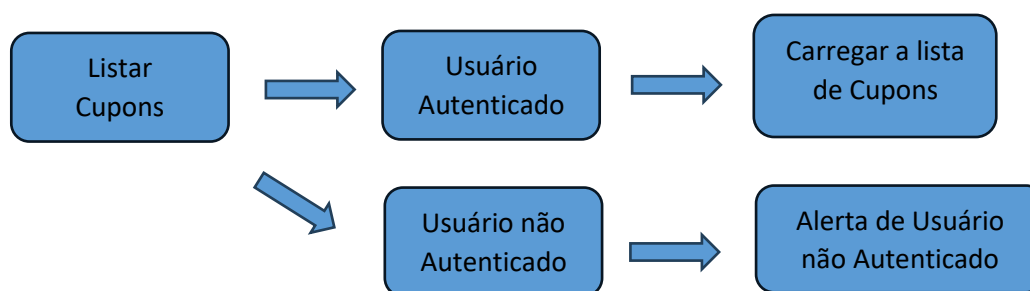
Condições	Regra 1	Regra 2	Regra 3
Tentativas de Login	2	3	>3
Ações			
Bloquear Usuário		X	X
Não Bloquear Usuário	X		

CT03: Bloqueio por tentativas de login incorreto

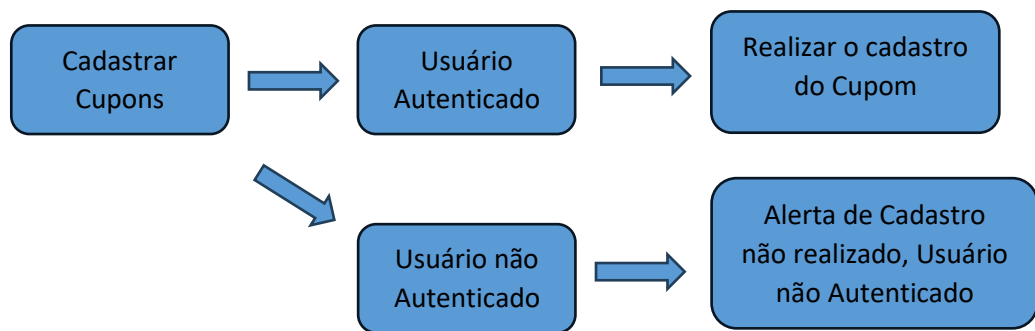
	Entrada	Saída
Regra 1	Usuário errou a senha na segunda tentativa	Válido
Regra 2	Usuário errou a senha na terceira tentativa	Inválido
Regra 3	Usuário acertou a senha na quarta tentativa	Inválido

4.3.1 História de usuário 3: API de Cupom

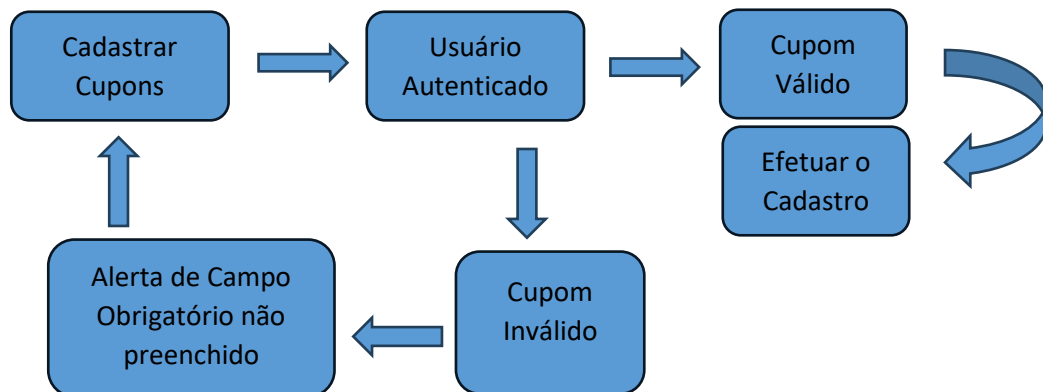
CT01: Somente usuários autenticados podem listar os cupons cadastrados



CT02: Somente usuários autenticados podem cadastrar cupons



CT03: Somente realizar o cadastro se os campos obrigatórios estiverem preenchidos.



4.4 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/MarcioDT/TCC-EBAC.git>

4.5 Testes automatizados

Testes automatizados consistem na utilização de software para a verificação da funcionalidade de um sistema, simulando as interações a fim de encontrar inconsistências e falhas.

Esse processo de teste é fundamental para uma melhor aproveitamento do tempo quando se diz respeito aos testes em um sistema, os testes automatizados podem ser mais rápidos e com uma maior precisão, não tendo interação humana.

4.5.1 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.

4.5.2 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)  
  })  
});
```

- Referência: Módulo 11, 12 e 14

4.6 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

4.7 Testes de performance

- Usando o Apache Jmeter, faça um teste de performance com o fluxo de login da 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

<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

- 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:

HTTP Request

Name: -31

Comments: Detected the start of a redirect chain

Basic: Advanced

Web Server

Protocol (http): Server Name or IP:

HTTP Request

POST Path: /minha-conta/

☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data ☐ Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value	UF
username	\$(usuario)	
password	\$(senha)	
woocommerce-login-nonce	0c0b00c000	
_wp_http_referer	/minha-conta/	
login	Login	

5. CONCLUSÃO

A realização deste trabalho me fez entender o quanto a Qualidade de software esta presente no dia a dia das empresas de desenvolvimento e o quando ela contribui para um desenvolvimento de um software robusto, seguro e que atenda totalmente as demandas que o cliente solicita.

A elaboração deste trabalho foi de grande ajuda na fixação do conteúdo aprendido no curso Qualidade de software oferecido pela EBAC, fazendo com que eu possa fixar todas as etapas de testes durante o desenvolvimento de um sistema.

Ao fim consegui compreender integralmente como o profissional de QA trabalha e desenvolve suas atividades, podendo eu aplicar estes conhecimentos na minha vida profissional.

6. REFERÊNCIAS BIBLIOGRÁFICAS

Lenildo, Qualidade de Software - Engenharia de Software 29, Devmedia, 2010, Disponível em: <https://www.devmedia.com.br/qualidade-de-software-engenharia-de-software-29/18209>, Acesso em: 19 Set. 2025