

**QUALIDADE DE SOFTWARE**

Carlos Henrique Almeida dos Santos

Análise de Qualidade

Brumado

2024

# RESUMO

Aqui você deve fazer um breve resumo do seu projeto. Aborde um pouco de tudo, mas não entre profundamente em nada. O “resumo” em um trabalho acadêmico “serve” para mostrar ao leitor se o conteúdo é de seu interesse ou não. Mas é um resumo, um breve relato de, no máximo 200 palavras.

# 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 5](#_Toc85541175)

[4.2.2 História de usuário 2: [US-0002] – Login na plataforma 6](#_Toc85541176)

[4.2.3 História de usuário 2: [US-0003] – API de cupons 6](#_Toc85541177)

[4.3 Casos de testes 6](#_Toc85541178)

[4.3.1 História de usuário 1: 6](#_Toc85541179)

[4.3.2 História de usuário 2: 6](#_Toc85541180)

[4.3.1 História de usuário 3: API de Cupom 7](#_Toc85541181)

[4.4 Repositório no Github 7](#_Toc85541182)

[4.5 Testes automatizados 7](#_Toc85541183)

[4.6 Integração contínua 8](#_Toc85541184)

[4.7 Testes de performance 8](#_Toc85541185)

[5. CONCLUSÃO 9](#_Toc85541186)

[6. REFERÊNCIAS BIBLIOGRÁFICAS 9](#_Toc85541187)

# INTRODUÇÃO

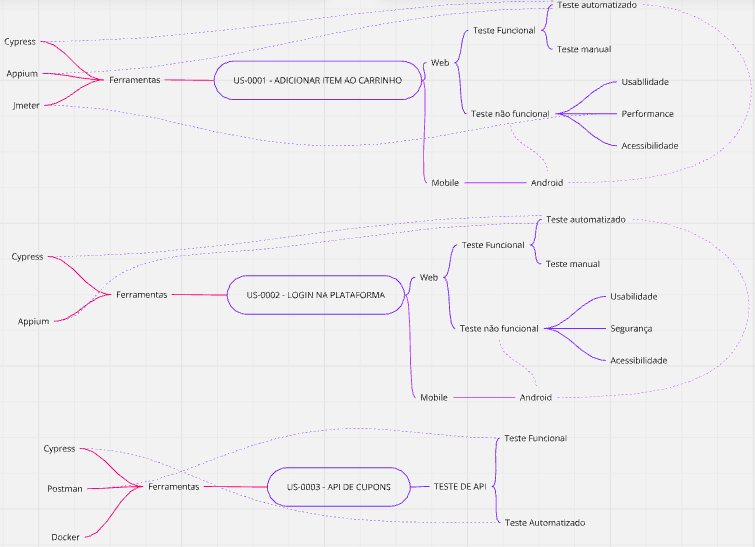
Na introdução você deve fazer um apanhado geral do seu cenário para o leitor. Escreva a prévia do que teríamos no trabalho, o que irá tratar, o que espera do projeto, etc.

# 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

* 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

* Considere as histórias de usuário: [US-0001] – Adicionar item ao carrinho, [US-0002] – Login na plataforma e [US-0003] – API de cupons
* Para cada uma delas crie pelo menos 2 critérios de aceitação usando a linguagem Gherkin;
* Em pelo menos um dos critérios, usar tabela de exemplos ( Esquema do Cenário / Scenario Outline);
* Referência: Módulo 8

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

Critérios de aceitação:  
  
Cenário 1: Inserir mais de itens que o permitido

Dado que eu acesse a página de adicionar itens ao carrinho  
Quando eu inserir mais de 10 itens

Então deve ser exibida mensagem de que o limite de itens é 10

Cenário 2: Cupom 15%

Dado que eu acesse a página de adicionar itens ao carrinho  
Quando eu inserir itens que custem no total mais de R$600,00

Então deve ser aplicado desconto de 15 %

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

Critérios de aceitação: Autenticação  
  
Cenário 1: Autenticação válida

Dado que eu acesse a página de autenticação

Quando eu inserir <usuário>

E a <senha>

Então deve ser exibida página de meus pedidos. Login efetuado com sucesso

Exemplos:

|  |  |
| --- | --- |
| usuario | senha |
| “carlos@gmail.com” | “carlos@123” |
| “joana”@gmail.com” | “joana@123” |
| “ana@hotmail.com” | “ana@123” |

Cenário 2: Login inválido

Dado que eu acesse a página de autenticação

Quando eu inserir usuário e senha não cadastrados

Então deve ser exibida mensagem de erro de login e senha inválidos

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

Critérios de aceitação:  
  
Cenário 1: Listar cupons

Dado que eu autentique na API de cupons

Quando eu enviar o comando GET

Então deve ser exibida lista de cupons cadastrados

Cenário 2: Cadastrar cupons

Dado que eu autentique na API de cupons

Quando eu enviar o comando SET com Código do cupom “Ganhe30”   
E Valor: “30.00”  
E Tipo do desconto: “fixed\_product”  
E Descrição: “Cupom de teste”

Então deve ser exibida mensagem de cupom cadastrado com sucesso

## 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

## História de usuário 1:

Nesta história utilizada a técnica de valor limite aplicada ao valor dos produtos.

CT01: Valor dos produtos R$989,99

1 – Faça login com usuário válido

2 - Adicione itens no carrinho em que a soma seja R$989,99

3 – Confirme a compra

Resultado esperado: Compra confirmada com sucesso

CT02: Valor dos produtos R$990,01

1 – Faça login com usuário válido

2- Adicione itens no carrinho em que a soma seja R$990,01

3 – Confirme a compra

Resultado esperado: Mensagem de limite de valor excedido deve ser apresentada

CT03: Valor dos produtos R$990,00

1 – Faça login com usuário válido

2 - Adicione itens no carrinho em que a soma seja R$990,00

3 – Confirme a compra

Resultado esperado: Compra confirmada com sucesso

## História de usuário 2:

CT01: Login e senha válidos

1 – Na página inicial, digite um e-mail, nome ou CPF válido

2 – Digite uma senha válida

3 – Faça o login

Resultado esperado: Deve ser exibida a página de boas vindas

CT02: Login inválido

1 – Na página inicial, digite um e-mail, nome, CPF ou senha inválidos

2 – Faça o login

Resultado esperado: Deve ser exibida mensagem de usuário ou senha inválidos

CT03: Login ou senha inválidos 3 vezes

1 – Na página inicial, digite um e-mail, nome, CPF ou senha inválidos

2 – Faça o login

3 – Repita os passos anteriores 3 vezes

Resultado esperado: Deve ser exibida mensagem de usuário bloqueado por 15 minutos

## História de usuário 3: API de Cupom

CT01: Listar cupons cadastrados

1 – Faça autenticação na API utilizando usuário “admin\_ebac” e senha “@admin!&b@c!2022”

2 – Faça uma consulta com o método GET

Resultado esperado: Todos os cupons listados devem ser exibidos

CT02: Cadastrar um cupom

1 – Faça autenticação na API utilizando usuário “admin\_ebac” e senha “@admin!&b@c!2022”

2 – Cadastre um cupom com o método POST inserindo os campos como segue:

- Código do cupom: Exemplo: “Ganhe30”  
- Valor: “30.00”  
- Tipo do desconto: “fixed\_product”  
-Descrição: “Cupom de 30”

3 – Faça uma consulta com o método GET

Resultado esperado: Cupom cadastrado deve ser exibido

CT03: Cadastrar cupom sem campo obrigatório

1 - Faça autenticação na API utilizando usuário “admin\_ebac” e senha “@admin!&b@c!2022”

2 – Cadastre um cupom com o método POST inserindo os campos como segue:

- Código do cupom: Exemplo: “Ganhe30”  
- Tipo do desconto: “fixed\_product”  
-Descrição: “Cupom de 30”

Resultado esperado: Deve ser exibida mensagem de erro ao cadastrar o cupom

## Repositório no Github

* Crie um repositório no github com o nome TCC-EBAC;
* Deixe o repositório público 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/carloshenriqueas/TCC-EBAC

## Testes automatizados

* + 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.
  + 1. 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.).
* 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 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

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

Coloque sua experiência na realização do trabalho, o que aprendeu, quais lições pode aplicar em sua vida profissional etc.

# REFERÊNCIAS BIBLIOGRÁFICAS

Seguir regras ABNT