

**QUALIDADE DE SOFTWARE**

Victor Vinicius Machado

Análise de Qualidade

Uberlândia

2025

# RESUMO

Este projeto tem como objetivo aplicar conceitos e práticas de testes automatizados, integração contínua e testes de performance em um ambiente realista, utilizando como base o site virtual da loja EBAC. A abordagem adotada visa garantir a qualidade do software por meio da implementação de testes eficientes e bem estruturados. Ao longo deste trabalho, serão apresentados os testes desenvolvidos, destacando sua funcionalidade e impacto na estabilidade da aplicação. O projeto reforça a importância da automação de testes no ciclo de desenvolvimento de software e demonstra a evolução técnica adquirida durante o curso.

# 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](#_Toc85541175) 6

[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 7](#_Toc85541177)

[4.3 Casos de testes 8](#_Toc85541178)

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

[4.3.2 História de usuário 2: 9](#_Toc85541180)

[4.3.3 História de usuário 3: API de Cupom 10](#_Toc85541181)

[4.4 Repositório no Github 11](#_Toc85541182)

[4.5 Testes automatizados 12](#_Toc85541183)

[4.6 Integração contínua 12](#_Toc85541184)

[4.7 Testes de performance 12](#_Toc85541185)

[5. CONCLUSÃO 14](#_Toc85541186)

[6. REFERÊNCIAS BIBLIOGRÁFICAS 14](#_Toc85541187)

# INTRODUÇÃO

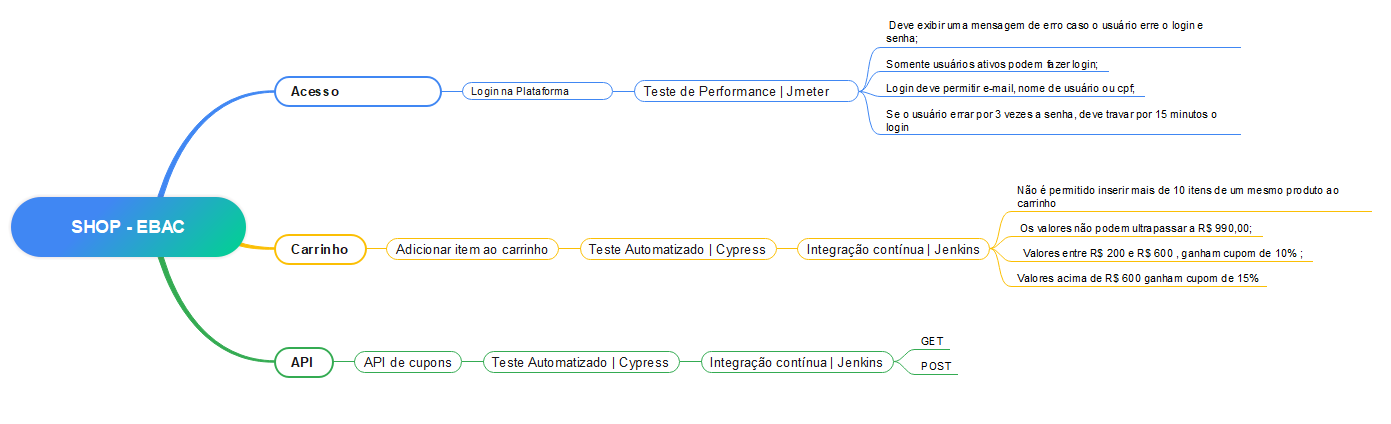
A garantia da qualidade de um software é essencial para sua confiabilidade e usabilidade. Com o avanço das metodologias ágeis e da cultura DevOps, a automação de testes tornou-se uma prática indispensável para otimizar o desenvolvimento, reduzir falhas e acelerar entregas. Este projeto final tem como propósito demonstrar a aplicação de testes automatizados, integração contínua e análise de performance no site virtual da loja EBAC. A implementação desses testes reflete a importância da automação na detecção precoce de erros e na manutenção da estabilidade da aplicação. Através deste trabalho, serão exploradas as estratégias utilizadas, as ferramentas empregadas e os resultados obtidos, evidenciando o conhecimento adquirido ao longo do curso.

# O PROJETO

Este Trabalho de Conclusão de Curso em Qualidade de Software segue uma abordagem baseada em metodologias ágeis, considerando histórias de usuário previamente refinadas. O desenvolvimento das funcionalidades seguirá todo o fluxo de trabalho de um QA, desde o planejamento até a entrega final.

## 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:



## 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:  
  
**Como** cliente da EBAC-SHOP  
**Quero** adicionar produtos no carrinho   
**Para** realizar a compra dos itens

**Regras de negócio:**

* Não é permitido inserir mais de 10 itens de um mesmo produto ao carrinho;
* Os valores não podem ultrapassar a R$ 990,00;
* Valores entre R$ 200 e R$ 600, ganham cupom de 10%
* Valores acima de R$ 600 ganham cupom de 15%

**Cenário: Ganhar cupom de 10%**

**Quando** eu inserir 10 itens de um mesmo produto ao carrinho

**E** a soma der R$300,00

**Então** ganharei um cupom de 10%

**Cenário: Não ganhar cupom**

**Quando** eu inserir 10 itens produtos diferentes ao carrinho

**E** a soma der R$100

**Então** não ganharei cupom devido ao valor total da compra

## 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

**Regras de negócio:**

* Somente usuários ativos podem fazer login;
* Deve exibir uma mensagem de erro caso o usuário erre o login e senha;
* Login deve permitir e-mail, nome de usuário ou cpf;
* Se o usuário errar por 3 vezes a senha, deve travar por 15 minutos o login

**Contexto:** Dado que eu acesse a plataforma virtual da EBAC-SHOP

**Cenário: Autenticação válida**

**Quando** eu digitar o usário "victor@ebac.com.br"

**E** a senha "senha@123"

**Então** serei direcionado a tela de checkout

**Cenário: Usuário com senha inválida**

**Quando** eu digitar o usário “victor@ebac.com.br”

**E** a senha “xyz@123”

**Então** deve exibir a mensagem de alerta “Usuário ou senha inválidos”

**Cenário: Usuário inexistente**

**Quando** eu digitar o usário "victor@ebac.com.br"

**E** a senha "abc@123"

**Então** deve exibir a mensagem de alerta "Usuário inexistente"

## 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

**Regras de negócio:**

**GET:**

* Deve listar todos os cupons cadastrado ou listar buscando por ID do cupom
* Documentação do serviço:   
  <http://lojaebac.ebaconline.art.br/rest-api/docs/#/coupons/get_wc_v3_coupons>   
  <http://lojaebac.ebaconline.art.br/rest-api/docs/#/coupons/get_wc_v3_coupons__id_>

**POST:**

* Deve cadastrar os cupons com os campos obrigatórios abaixo:

- Código do cupom: Exemplo: “Ganhe10”   
- Valor: “10.00”  
- Tipo do desconto: “fixed\_product”  
-Descrição: “Cupom de teste”

* Nome do cupom não pode ser repetido;
* Os outros campos são opcionais.
* Documentação do serviço:   
  <http://lojaebac.ebaconline.art.br/rest-api/docs/#/coupons/post_wc_v3_coupons>
* Exemplo do body:

**body: {  
 "code": "nomeCupom",  
 "amount": "10",  
 "discount\_type": "fixed\_product",  
 "description": "Cupom de desconto de teste"  
 }**  
**Contexto:** Dado que eu acesse API para cadastrar cupons na loja virtual EBAC-SHOP

**Cenário: Cadastro de cupons repetidos**

**Quando** eu tentar cadastrar um cupom

**E** for repetido

**Então** serei alertado que este cupom já está cadastrado

**Cenário: Listar cupons já cadastrados**

**Quando** eu finalizar os cadastros dos cupons

**E** tentar busca-los

**Então** deve ser listados todos os cupons já cadastrados

## 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: Adicionar item ao carrinho

**Regras de negócio:**

RN01: Não é permitido inserir mais de 10 itens de um mesmo produto ao carrinho;  
RN02: Os valores não podem ultrapassar a R$ 990,00;  
RN03: Valores entre R$ 200 e R$ 600, ganham cupom de 10%  
RN04: Valores acima de R$ 600 ganham cupom de 15%

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RN | TÍTULO | SAÍDA |
| CT01 | RN01 | Inserir mais de 10 itens de um mesmo produto ao carrinho | Inválido |
| CT02 | RN01 | Inserir mais de 10 itens de produtos diferentes no carrinho | Válido |
| CT03 | RN02 | Valor total do carrinho no valor de R$1.000,00 | Inválido |
| CT04 | RN03 | Liberar cupom de desconto de 10% em compras no valor de R$100,00 | Inválido |
| CT05 | RN04 | Liberar cupom de desconto de 15% em compras no valor de R$700,00 | Válido |

## História de usuário 2: Login na Plataforma

**Regras de negócio:**

RN01: Somente usuários ativos podem fazer login;  
RN02: Deve exibir uma mensagem de erro caso o usuário erre o login e senha;  
RN03: Login deve permitir e-mail, nome de usuário ou cpf;  
RN04: Se o usuário errar por 3 vezes a senha, deve travar por 15 minutos o login

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RN | TITULO | SAÍDA |
| CT01 | RN01 | Quando usuário inativo tenta realizar login na plataforma recebe o alerta "Usuário inexistente" | Válido |
| CT02 | RN02 | Usuário não recebe mensagem de erro "Usuário inexistente" ao informar usuário e senha incorretos | Inválido |
| CT03 | RN04 | Usuário erra a senha por 3 vezes e não ocorre a trava de 15 minutos | Inválido |

## História de usuário 3: API de Cupom

**Regras de negócio:**

**GET:**

RN01: Deve listar todos os cupons cadastrado ou listar buscando por ID do cupom. Documentação do serviço:   
<http://lojaebac.ebaconline.art.br/rest-api/docs/#/coupons/get_wc_v3_coupons>   
<http://lojaebac.ebaconline.art.br/rest-api/docs/#/coupons/get_wc_v3_coupons__id_>

**POST:**

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

- Código do cupom: Exemplo: “Ganhe10”   
- Valor: “10.00”  
- Tipo do desconto: “fixed\_product”  
-Descrição: “Cupom de teste”

RN03: Nome do cupom não pode ser repetido;  
RN04: Os outros campos são opcionais. Documentação do serviço:   
<http://lojaebac.ebaconline.art.br/rest-api/docs/#/coupons/post_wc_v3_coupons>

* Exemplo do body:

**body: {  
 "code": "nomeCupom",  
 "amount": "10",  
 "discount\_type": "fixed\_product",  
 "description": "Cupom de desconto de teste"**

**}**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RN | TITULO | SAÍDA |
| CT01 | RN01 | Não é listado os cupons cadastrados | Inválido |
| CT02 | RN02 | Cupom cadastrado sem preencher o campo: Descrição | Inválido |
| CT03 | RN03 | Nome do cupom cadastrado repetido | Inválido |
| CT04 | RN04 | Alerta de erro ao cadastrar cupom já cadastrado | Válido |

## 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/victor1991-prog/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: 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

Ao longo do curso, pude compreender os testes de software de ponta a ponta, adquirindo uma base teórica inicial para ingressar no mercado de trabalho. A prática com ferramentas modernas ampliou minha visão sobre automação e qualidade. No entanto, alguns conteúdos do curso precisam ser atualizados para refletir melhor as tecnologias e práticas atuais.

# REFERÊNCIAS BIBLIOGRÁFICAS

<https://chatgpt.com/>