

**QUALIDADE DE SOFTWARE**

Guilherme Henrique do Amaral

Análise de Qualidade

São Paulo

2024

# RESUMO

O objetivo da realização deste projeto é apresentar na prática alguns dos conhecimentos que aprendi durante o curso de Teste de Software da EBAC, através da realização de tarefas que são feitas cotidianamente por testadores. Tarefas como: a criação de casos de teste, a automação de testes de interface de usuário (UI), a automação de testes de API e a integração de testes em pipelines de integração contínua.

# SUMÁRIO

[1. RESUMO 2](#_Toc85541188)

[2. SUMÁRIO 3](#_Toc85541189)

[3. INTRODUÇÃO 4](#_Toc85541190)

[4. O PROJETO 5](#_Toc85541191)

[4.1 Estratégia de teste 5](#_Toc85541192)

[4.2 Critérios de aceitação 5](#_Toc85541193)

[4.2.1 História de usuário 1: [US-0001] – Adicionar item ao carrinho 5](#_Toc85541194)

[4.2.2 História de usuário 2: [US-0002] – Login na plataforma 6](#_Toc85541195)

[4.2.3 História de usuário 2: [US-0003] – API de cupons 6](#_Toc85541196)

[4.3 Casos de testes 6](#_Toc85541197)

[4.3.1 História de usuário 1: 6](#_Toc85541198)

[4.3.2 História de usuário 2: 6](#_Toc85541199)

[4.3.1 História de usuário 3: API de Cupom 7](#_Toc85541200)

[4.4 Repositório no Github 7](#_Toc85541201)

[4.5 Testes automatizados 7](#_Toc85541202)

[4.6 Integração contínua 8](#_Toc85541203)

[4.7 Testes de performance 8](#_Toc85541204)

[5. CONCLUSÃO 9](#_Toc85541205)

[6. REFERÊNCIAS BIBLIOGRÁFICAS 9](#_Toc85541206)

# INTRODUÇÃO

No desenvolvimento de um software, a presença de um profissional dedicado ao teste de qualidade é essencial para garantir o sucesso de um produto. Este papel não apenas assegura que os critérios de aceitação definidos para o projeto sejam atendidos, mas também é fundamental para identificar e corrigir falhas antes que o software seja entregue ao cliente. Detectar bugs em fases iniciais do ciclo de desenvolvimento reduz significativamente os custos de correção e previne problemas que poderiam impactar negativamente a experiência do usuário final.

Além disso, a automação de testes desempenha um papel estratégico nesse contexto, permitindo maior eficiência e abrangência na validação de funcionalidades ao testar grandes volumes de dados e cenários de uso em um tempo significativamente menor do que seria possível com testes manuais. Isso não apenas aprimora a qualidade do produto em si, mas também proporciona maior confiança às equipes de desenvolvimento na entrega de versões incrementais aos clientes. Assim, o profissional de Teste de Software, utilizando ferramentas e técnicas de automação, desempenha um papel central no sucesso de projetos.

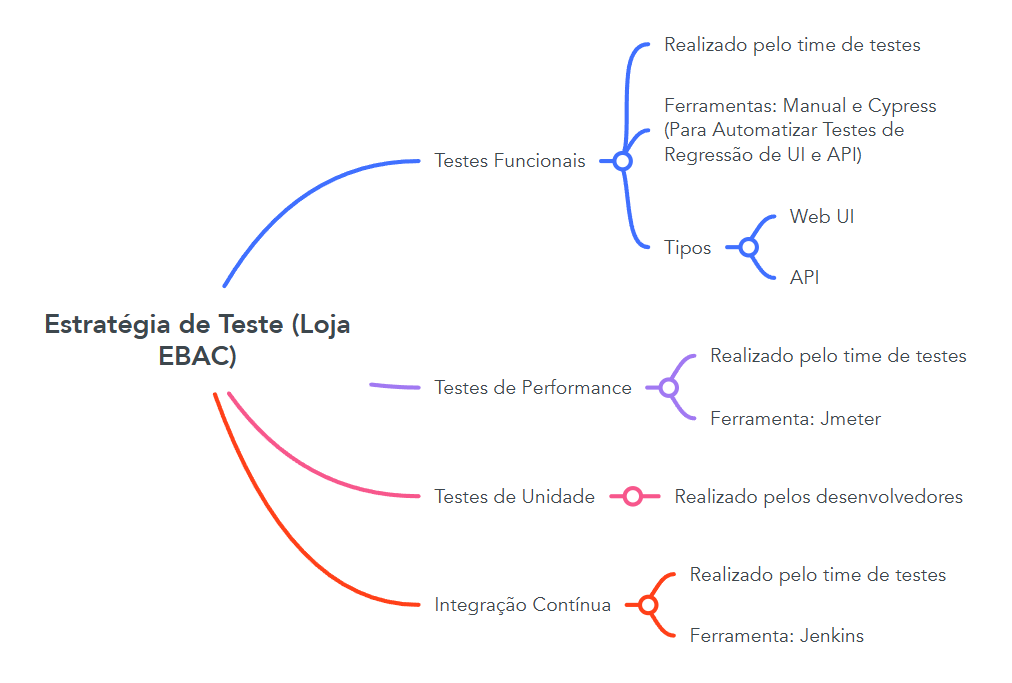
Este trabalho apresenta algumas das principais atividades que compõem o dia a dia de um profissional de Teste de Software, consolidando conhecimentos adquiridos no curso profissionalizante de Teste de Software da EBAC e destacando a importância dessas práticas para a entrega de sistemas confiáveis e de alta qualidade.

# 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 subtópicos 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:



## 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

Os critérios de aceitação escritos em Gherkin dessa *user story* podem ser encontrados no arquivo [carrinho.feature](https://github.com/guilhermeha7/TCC-EBAC/blob/main/criterios-de-aceitacao-em-gherkin/carrinho.feature) do repositório do projeto

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

Os critérios de aceitação escritos em Gherkin dessa *user story* podem ser encontrados no arquivo [login.feature](https://github.com/guilhermeha7/TCC-EBAC/blob/main/criterios-de-aceitacao-em-gherkin/login.feature) do repositório do projeto

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

Os critérios de aceitação escritos em Gherkin dessa *user story* podem ser encontrados no arquivo [api-cupons.feature](https://github.com/guilhermeha7/TCC-EBAC/blob/main/criterios-de-aceitacao-em-gherkin/api-cupons.feature) do repositório do projeto

## 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:

CT01: Deve adicionar produto ao carrinho

|  |  |
| --- | --- |
| Descrição: | Verificar se o produto é adicionado ao carrinho |
| Pré-condições: | Produto em estoque |
| Massa de dados: | 0, 2, 4 (ids de itens da lista de produtos) |
| Prioridade: | Alta |

|  |  |
| --- | --- |
| Ação | Resultado Esperado |
| 1. Abrir lista de produtos |  |
| 1. Selecionar um produto |  |
| 1. Escolher uma cor, tamanho e quantidade para o produto |  |
| 1. Clicar no botão Comprar | O produto deve ser adicionado ao carrinho |

CT02: Deve impedir adição de 11 itens de um mesmo produto ao carrinho

|  |  |
| --- | --- |
| Descrição: | Verificar se a regra de negócio “Não é permitido inserir mais de 10 itens de um mesmo produto ao carrinho” está sendo obedecida |
| Pré-condições: | Produto em estoque |
| Massa de dados: | 1 (id do segundo item da lista de produtos) |
| Prioridade: | Média |

|  |  |
| --- | --- |
| Ações: | Resultado Esperado: |
| 1. Abrir lista de produtos |  |
| 1. Selecionar um produto |  |
| 1. Escolher uma cor, tamanho e 11 unidades para o produto |  |
| 1. Clicar no botão Comprar | Deve aparecer um aviso e impedir a adição do produto ao carrinho |

CT03: Deve impedir adição de unidades acima do estoque

|  |  |
| --- | --- |
| Descrição: | Verificar se o sistema não permite adicionar ao carrinho mais unidades de um produto do que a quantidade disponível em estoque. |
| Pré-condições: | Produto em estoque |
| Massa de dados: | 2 (id do terceiro item da lista de produtos) |
| Prioridade: | Alta |

|  |  |
| --- | --- |
| Ações: | Resultado Esperado: |
| 1. Abrir lista de produtos |  |
| 1. Selecionar um produto |  |
| 1. Escolher uma cor, tamanho e uma quantidade de unidades maior que o estoque disponível |  |
| 1. Clicar no botão Comprar | Deve aparecer uma mensagem informando que não há estoque suficiente para a quantia desejada. |

## História de usuário 2:

CT01: Deve fazer login com dados válidos

|  |  |
| --- | --- |
| Descrição: | Verificar se o login funciona corretamente |
| Pré-condições: | Conta cadastrada |
| Massa de dados: | robertoteste@teste.com, teste123 |
| Prioridade: | Alta |

|  |  |
| --- | --- |
| Ações: | Resultado Esperado: |
| 1. Clicar no ícone de fazer login |  |
| 1. Digitar um e-mail e uma senha válidos |  |
| 1. Clicar no botão Login | Deve fazer login com sucesso |

CT02: Deve impedir login com senha incorreta

|  |  |
| --- | --- |
| Descrição: | Verificar se o sistema impede a entrada do usuário se a senha digitada estiver incorreta |
| Pré-condições: | Conta cadastrada |
| Massa de dados: | robertoteste@teste.com, teste31232 |
| Prioridade: | Alta |

|  |  |
| --- | --- |
| Ações: | Resultado Esperado: |
| 1. Clicar no ícone de fazer login |  |
| 1. Digitar um e-mail válido e uma senha inválida |  |
| 1. Clicar no botão Login | Deve aparecer uma mensagem informando que a senha digitada está incorreta |

CT03: Deve impedir login com endereço de e-mail não cadastrado

|  |  |
| --- | --- |
| Descrição: | Verificar se o sistema impede a entrada do usuário caso o e-mail digitado não estiver cadastrado |
| Pré-condições: |  |
| Massa de dados: | qewrrrqweqwreqw@teste.com, teste123 |
| Prioridade: | Baixa |

|  |  |
| --- | --- |
| Ações: | Resultado Esperado: |
| 1. Clicar no ícone de fazer login |  |
| 1. Digitar um e-mail não cadastrado e uma senha qualquer |  |
| 1. Clicar no botão Login | Deve aparecer uma mensagem dizendo que o endereço de e-mail digitado não está cadastrado |

## História de usuário 3: API de Cupom

CT01: Deve listar todos os cupons

|  |  |
| --- | --- |
| Descrição: | Verificar se são listados todos os cupons criados |
| Pré-condições: |  |
| Massa de dados: |  |
| Prioridade: | Média |

|  |  |
| --- | --- |
| Ações: | Resultado Esperado: |
| 1. Realizar autenticação de admin |  |
| 1. Enviar uma requisição do tipo GET para o endpoint “http://lojaebac.ebaconline.art.br/wp-json/wc/v3/coupons” | Deve aparecer todos os cupons criados |

CT02: Deve listar um cupom específico

|  |  |
| --- | --- |
| Descrição: | Validar a listagem de um cupom específico |
| Pré-condições: | Cupom já criado |
| Massa de dados: | Id dinâmico |
| Prioridade: | Baixa |

|  |  |
| --- | --- |
| Ações: | Resultado Esperado: |
| 1. Realizar autenticação de admin |  |
| 1. Enviar uma requisição do tipo GET para o endpoint “http://lojaebac.ebaconline.art.br/wp-json/wc/v3/coupons/{id}” (trocando o {id} pelo id do cupom procurado) | Deve aparecer o cupom procurado |

CT03: Deve criar novo cupom

|  |  |
| --- | --- |
| Descrição: | Validar a criação de cupom |
| Pré-condições: | Nome do cupom não pode já existir |
| Massa de dados: | Código do cupom e sua porcentagem são criados dinamicamente |
| Prioridade: | Média |

|  |  |
| --- | --- |
| Ações: | Resultado Esperado: |
| 1. Realizar autenticação de admin |  |
| 1. Enviar uma requisição do tipo POST para o endpoint “http://lojaebac.ebaconline.art.br/wp-json/wc/v3/coupons/”, com o body preenchido como no exemplo abaixo:   {  "code": "nomeCupom", "amount": "10", "discount\_type": "fixed\_product", "description": "Cupom de desconto de 10%" } | Deve criar o cupom com sucesso |

## 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: [Repositório do TCC-EBAC](https://github.com/guilhermeha7/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 usados. Substitua esses parâmetros pela sua massa de dados, conforme aprendido em aula:

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

# CONCLUSÃO

Realizar este projeto foi uma experiência muito enriquecedora, pois me ajudou a consolidar e colocar em prática vários dos conhecimentos adquiridos durante o curso de Teste de Software. Durante este trabalho, consegui entender melhor a importância dos testes na garantia de qualidade dos sistemas e aprimorar minhas habilidades em automação de testes e integração contínua.

Gostaria de agradecer ao professor Fábio Araújo por todo o conhecimento compartilhado ao longo do curso, fundamental para o meu aprendizado e para a realização deste projeto. Meu objetivo é continuar estudando e me aperfeiçoando na área de automação de testes, aprendendo novas ferramentas e tecnologias.

# REFERÊNCIAS BIBLIOGRÁFICAS

VERICODE. *Testes de software: tudo que você precisa saber*. Disponível em: https://blog.vericode.com.br/testes-de-software/. Acesso em: 25 nov. 2024.

ACCURATE. Automação de testes: o que é, como funciona e quais são suas vantagens. Disponível em: https://blog.accurate.com.br/automacao-de-testes/. Acesso em: 25 nov. 2024.