

**QUALIDADE DE SOFTWARE**

Gabriel Alexandre de Oliveira Castilho

Análise de Qualidade

Taubaté

2023

# 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

Abaixo consta os prints dos mapas mentais para cada User Stories, também o link caso seja necessário visualizar por completo [Miro](https://miro.com/app/board/uXjVMYame5U=/?share_link_id=431562226626).

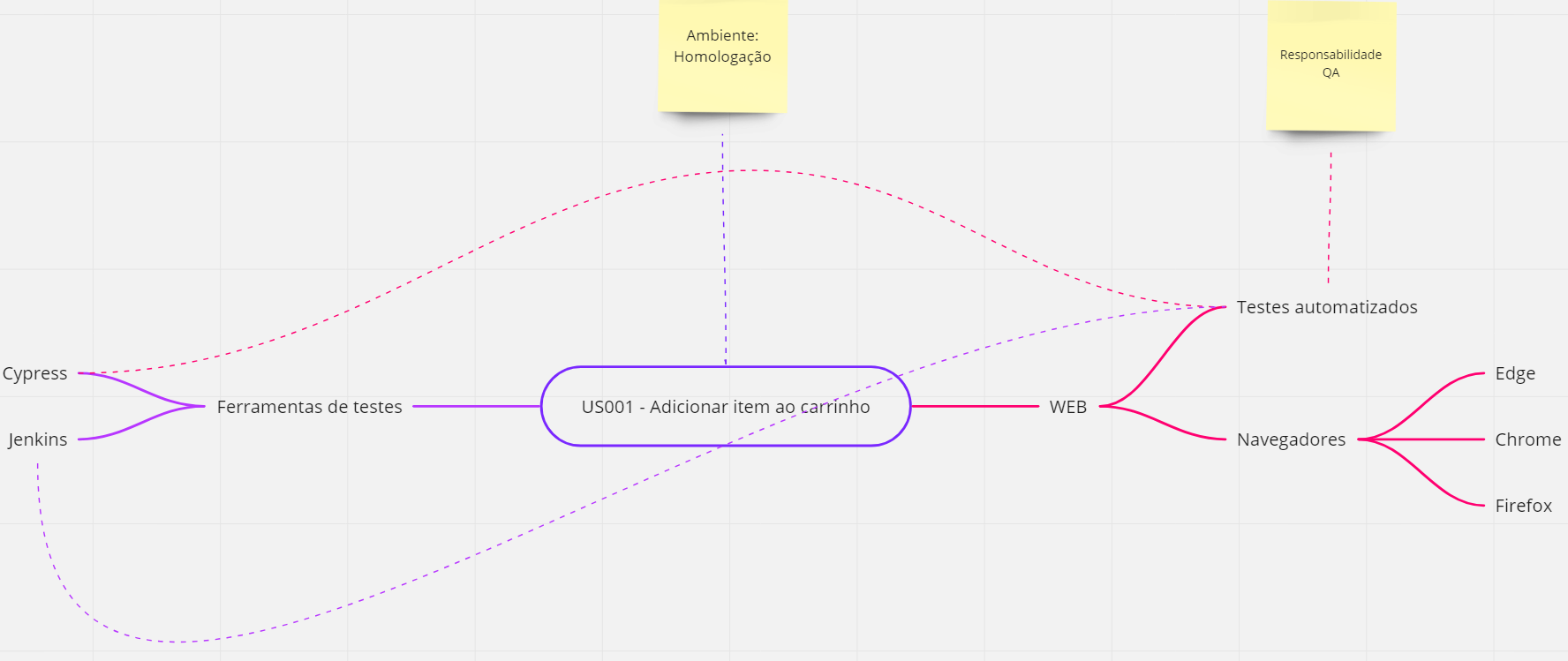


IMAGEM: Estratégia de teste US001 – Adicionar item ao carrinho

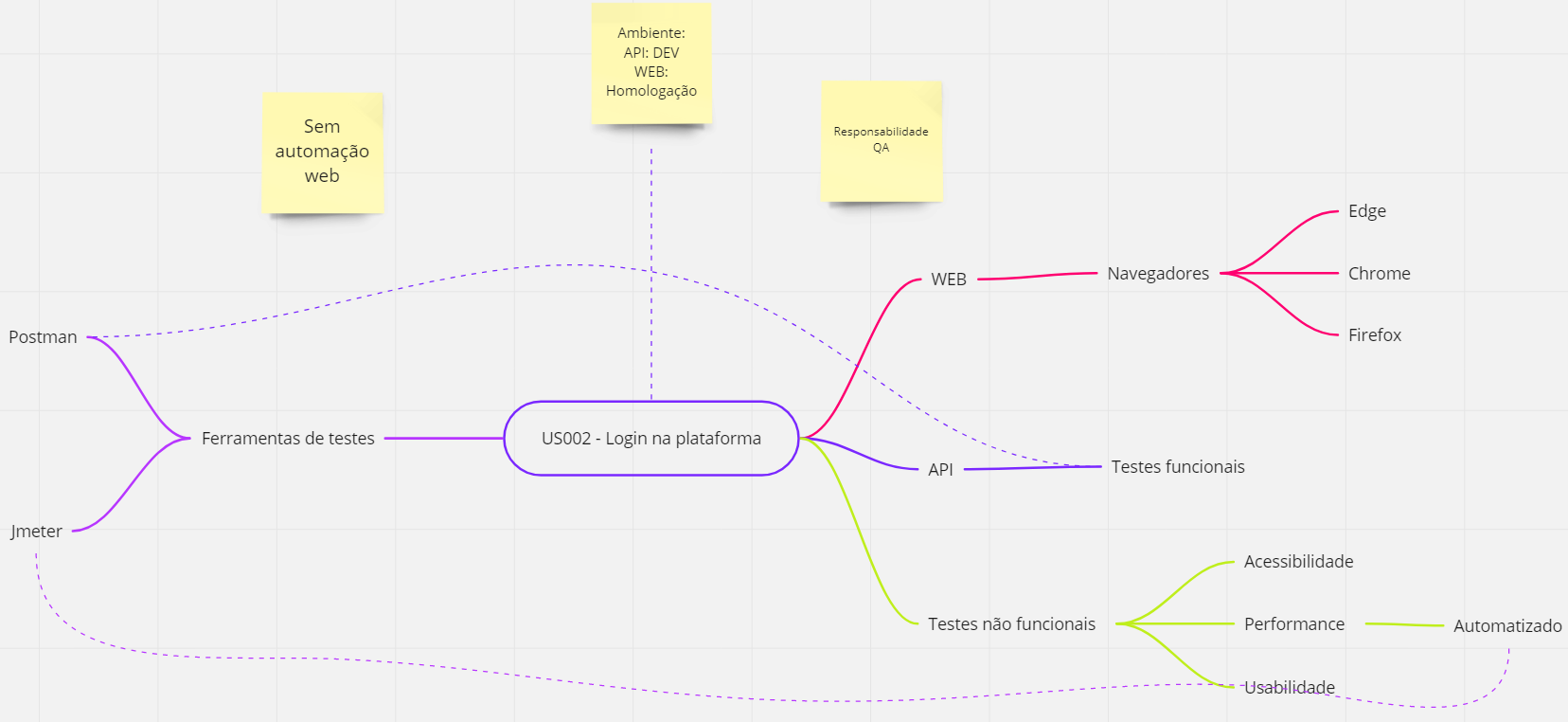


IMAGEM: Estratégia de teste US002 – Login na plataforma

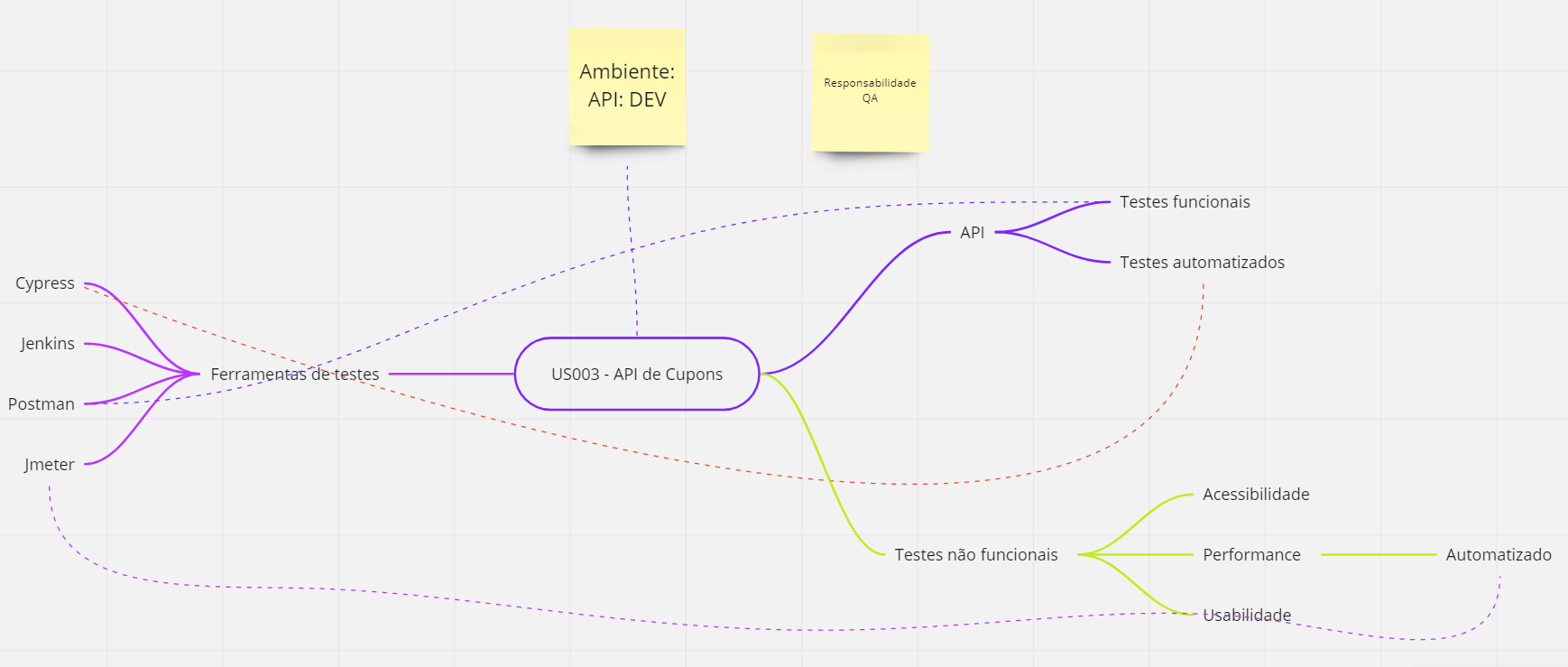


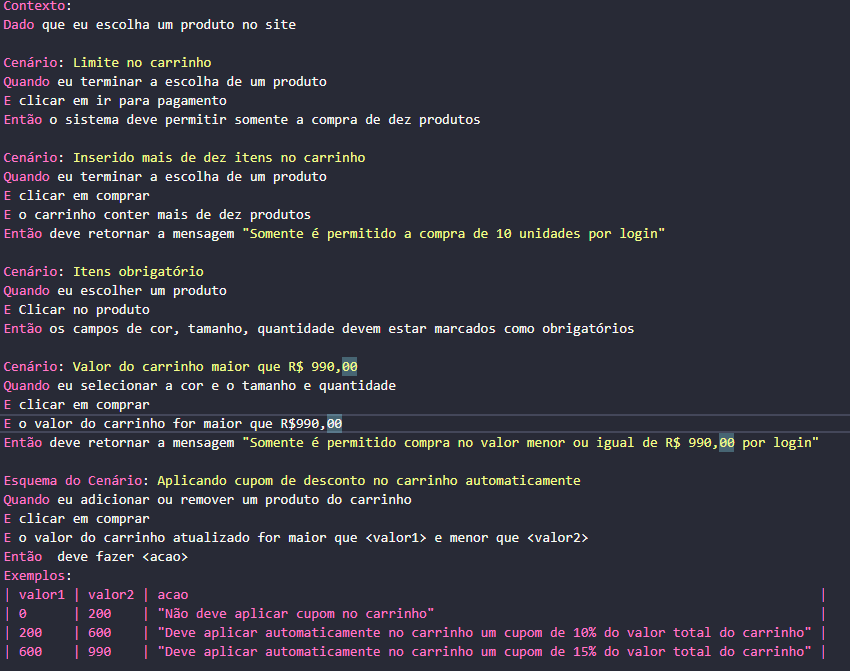
IMAGEM: Estratégia de teste US003 – API de cupons

## Critérios de aceitação

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

Critérios de aceitação:

1. Deve permitir a inserção de no máximo 10 unidades por produto no carrinho;
2. Seleções de cor, tamanho e quantidade devem ser obrigatórios
3. O valor máximo do carrinho é de R$ 990,00
4. Se um produto for adicionado e o valor do carrinho for entre R$ 200,00 e R$ 600,00 então deve aplicar um cupom de 10% no carrinho
5. Se um produto for removido e o valor do carrinho for menor que R$ 200,00 então o cupom deve ser removido do carrinho.
6. Se um produto for adicionado e o valor do carrinho for maior que R$ 600,00 então deve aplicar um cupom de 15% no carrinho.
7. Se um produto for removido e valor do carrinho for menor que R$ 600 então deve seguir a regra número 3 se for entre R$ 200,00 e R$ 600,00 ou a regra número 4 se for menor que R$ 200,00

  
IMAGEM: Critério de aceitação – US001 – Adicionar item ao carrinho

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

Critérios de aceitação:

1 – Deve permitir a autenticação via CPF, Email ou nome de usuário

2 – Ao ser inserido credenciais inválidas deve retornar a mensagem “Usuário ou senha inválidos”

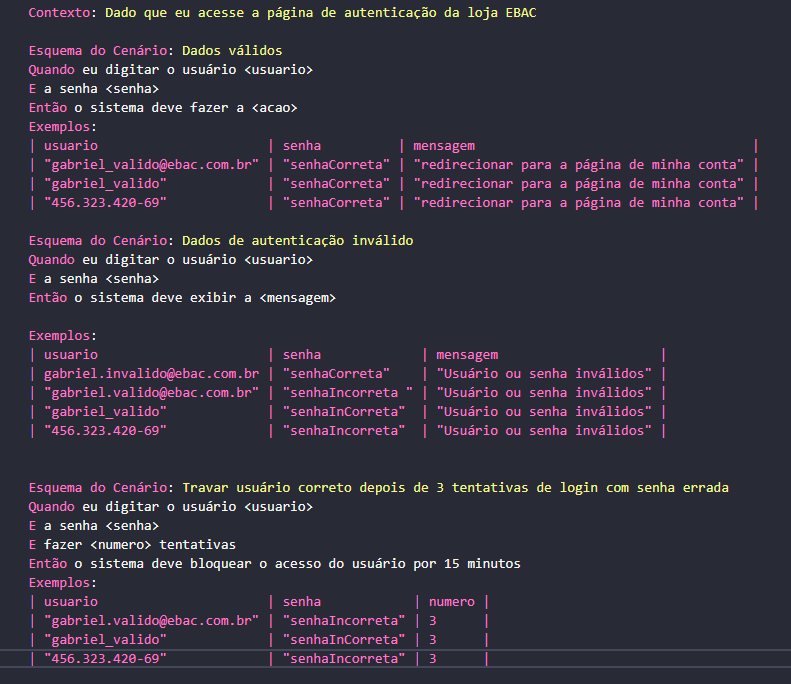
3 – Após um usuário válido errar a senha três vezes, o sistema deve bloquear o login desse usuário por 15 minutos  
  


IMAGEM: Critério de aceitação – US002 – Login na plataforma

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

Critérios de aceitação:

Não pode ser possível cadastrar um cupom com código repetido

O cupom deve aceitar o desconto no valor fixo

O cupom deve aceitar o desconto no valor em porcentagem

Os campos: Code, amout, discount\_type, description devem ser obrigatórios

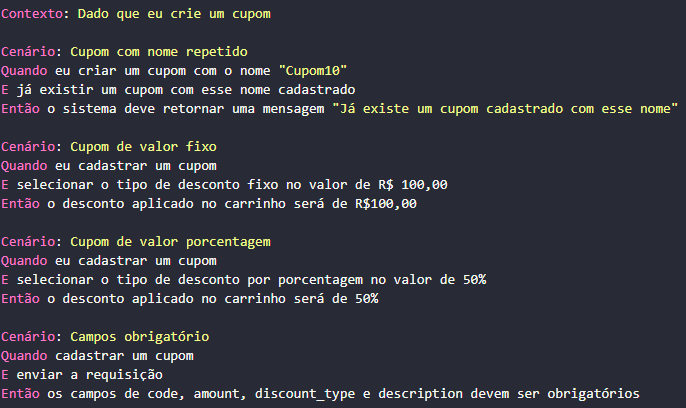


IMAGEM: Critério de aceitação – US003 – API de cupons

## 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:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RN | Título | Saída |
| CT01 | RN01 | Inserção de 10 itens no carrinho | válido |
| CT02 | RN01 | Inserção de 9 itens no carrinho | Válido |
| CT03 | RN01 | Inserção de 11 itens no carrinho | Inválido |
| CT04 | RN02 | Pagamento com carrinho no valor de R$ 990,01 | Inválido |
| CT05 | RN02 | Pagamento com carrinho no valor de R$ 990,00 | Válido |
| CT06 | RN02 | Pagamento com carrinho no valor de R$ 989,00 | Válido |

**RN03**: Valores no carrinho entre R$ 200,00 e R$ 600,00 ganham cupom de 10%

**RN04**: Valores no carrinho acima de R$ 600,00 ganham cupom de 15%

|  |  |  |  |
| --- | --- | --- | --- |
| **Condição** | **Regra 1 – Sem cupom** | **Regra 2 - Cupom 10%** | **Regra 3 – Cupom 15%** |
| Carrinho < R$ 200,00 | Sim | Não | Não |
| Carrinho >= 200 E Carrinho <600 | Não | Sim | Não |
| Carrinho >= 600 e Carrinho <=990,00 | Não | Não | Sim |
| **Ação** |  |  |  |
| Aplicar cupom? | Não | Sim | Sim |

## História de usuário 2:

|  |
| --- |
| **Fluxo principal** |
| 1 – Acessar a tela de login |
| 2 – Preencher com email, CPF, ou nome de usuário já cadastrado e clicar em Login |
| 3 – Exibir a tela de minha conta |

|  |
| --- |
| **Fluxo Alternativo** |
| 1 – Inserir usuário ou senha inválido |
| 2 – O sistema deve mostrar a mensagem “Usuário senha inválido” |
| 3 – Inserir o usuário correto, mas com a senha errada três vezes |
| 4 – O sistema deve bloquear o usuário por quinze minutos |

## História de usuário 3: API de Cupom

|  |
| --- |
| **Fluxo Principal** |
| 1 – Fazer GET na rota de cupons |
| 2 – Retornar todos os cupons cadastrados |
| 3 - Fazer POST na rota cupons com os dados obrigatórios inseridos |
| 4 – O cupom deve ser cadastrado e retornado o status code 200 |

|  |
| --- |
| **Fluxo Alternativo** |
| 1 – Fazer GET sem estar com autenticação |
| 2 – Deve retornar o status code 400 |
| 3 – Fazer POST sem estar com autenticação |
| 4 – Deve retornar o status code 400 |
| 5 – Fazer um POST passando um código já existente |
| 6 – Deve retornar o status code 400 |

## 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: <cole o link aqui>

## 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