

**QUALIDADE DE SOFTWARE**

Felipe Rodrigues Bento Dantas

Análise de Qualidade

Jundiaí, SP

2022

# RESUMO

Sabemos que hoje em dia a tecnologia tem estado agressivamente presente em nossas vidas. Mas não sabemos o quanto é importante ter um profissional QA ou Analista de Testes de Software até abordarmos este presente trabalho de conclusão do curso de Qualidade de Software.  
A princípio, passaremos por temas relacionados à fundamentos de testes, como planos, estratégias e técnicas de testes; também veremos sobre critérios de aceitação e casos de testes baseados em três exemplos de funcionalidades do portal web do instituto EBAC: EBAC Shop. E por fim, abordaremos testes automatizados, testes de performance e tudo isso baseado em cultura DevOps aliado às Metodologias Ágeis.

# SUMÁRIO

[1. RESUMO 2](#_Toc85541169)

[2. SUMÁRIO 3](#_Toc85541170)

[3. LISTA DE FIGURAS 3](#_Toc85541170)

[4. INTRODUÇÃO 4](#_Toc85541171)

[5. O PROJETO 5](#_Toc85541172)

[5.1 Estratégia de teste 5](#_Toc85541173)

[5.2 Critérios de aceitação 5](#_Toc85541174)

[5.2.1 História de usuário 1: [US-0001] – Adicionar item ao carrinho 5](#_Toc85541175)

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

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

[5.3 Casos de testes 8](#_Toc85541178)

[5.3.1 História de usuário 1: Adicionar item ao carrinho 8](#_Toc85541179)

[5.3.2 História de usuário 2: Login na plataforma 10](#_Toc85541180)

[5.3.1 História de usuário 3: API de Cupons 13](#_Toc85541181)

[5.4 Repositório no Github 15](#_Toc85541182)

[5.5 Testes automatizados 16](#_Toc85541183)

[5.6 Integração contínua 17](#_Toc85541184)

[5.7 Testes de performance 18](#_Toc85541185)

[6. CONCLUSÃO 19](#_Toc85541186)

[7. REFERÊNCIAS BIBLIOGRÁFICAS 20](#_Toc85541187)

# LISTA DE FIGURAS

[1. IMAGEM 01 - PLANEJAMENTO 2](#_Toc85541169)

[2. IMAGEM 02 – MATRIZ DE RISCO – US-0001 3](#_Toc85541170)

[3. IMAGEM 03 – MATRIZ DE RISCO – US-0002 3](#_Toc85541170)

[4. IMAGEM 04 – MATRIZ DE RISCO – US-0003 3](#_Toc85541170)

[5. IMAGEM 05 – EVIDÊNCIAS – CYPRESS – UI - DASHBOARD 3](#_Toc85541170)

[6. IMAGEM 06 – EVIDÊNCIAS – CYPRESS – UI – GERAR TESTES E VÍDEOS 3](#_Toc85541170)

[7. IMAGEM 07 – EVIDÊNCIAS – CYPRESS – API - GERAR TESTES E VÍDEOS 4](#_Toc85541171)

[8. IMAGEM 08 – EVIDÊNCIAS – CYPRESS – API - DASHBOARD 5](#_Toc85541172)

[9. IMAGEM 09 – JENKINS – ETAPAS DE INTEGRAÇÃO CONTÍNUA – TESTES UI E API 18](#_Toc85541171)

[10. IMAGEM 10 – JMETER – TESTE DE PERFORMANCE COM TEMPLATE RECORDING 18](#_Toc85541171)

# INTRODUÇÃO

Nesse trabalho de conclusão iremos abordar todo o processo que um QA ou Engenheiro Analista de Testes precisa aplicar para garantir a Qualidade de um Software. Passaremos desde fundamentos até as melhores práticas de automatização de testes com metodologias ágeis e cultura DevOps.

Sendo assim, a expectativa é que além de praticarmos todo o nosso aprendizado durante o curso, o conteúdo deste projeto agregue valor ao leitor e às pessoas que se interessam pelo assunto e que de alguma forma as inspirem a conhecerem mais sobre o assunto e até mesmo ter o desejo em serem um QA ou Analista de Testes.

# O PROJETO

## Estratégia de teste

## 

**Imagem 01 - Planejamento**

## Critérios de aceitação

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

Critérios de aceitação:  
  
Cenário 1: Compra com mais de 10 itens de um mesmo produto

Dado que eu acesse a página de carrinho da EBAC-SHOP  
Quando eu inserir a quantidade “11”

E clicar em Concluir a Compra

Então deve exibir uma mensagem de alerta "O limite máximo é de até 10 itens por venda!"

Cenário 2: Compra com valor acima de R$ 600  
Dado que eu acesse a página de carrinho da EBAC-SHOP  
Quando eu inserir a quantidade <quantidade> do produto <produto>

Então deve ser aplicado um desconto de 15%

Exemplos:

|  |  |
| --- | --- |
| quantidade | produto |
| “7” | “Ingrid Running Jacket - XS, Red” |
| “8” | “Ingrid Running Jacket - XS, Red” |

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

Critérios de aceitação:  
  
Cenário 1: Login e senha inválidos

Dado que eu acesse a página de login da EBAC-SHOP  
Quando eu digitar o login “felipe@ebac.com.br”

E a senha “senha@123”

Então deve exibir uma mensagem de alerta "Usuário e/ou senha inválidos!"

Cenário 2: Login com nome, email ou cpf

Dado que eu acesse a página de login da EBAC-SHOP  
Quando eu digitar o login <login>

E a senha <senha>

Então deve exibir uma mensagem de boas vindas <mensagem>

Exemplos:

|  |  |  |
| --- | --- | --- |
| login | senha | mensagem |
| “felipe.dantas@ebac.com.br” | “senha#123” | “Olá, **Felipe Dantas”** |
| “feliperbdantas” | “senha#123” | “Olá, **Felipe Dantas”** |
| 15836995893 | “senha#123” | “Olá, **Felipe Dantas”** |

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

Critérios de aceitação:  
  
Cenário 1: Listagem de cupons

Dado que eu acesse a documentação do serviço da EBAC-SHOP

E esteja autenticado  
Quando eu inserir o id “1”

Então deve retornar status “200” com os dados do cupom de id "1”

Cenário 2: Cadastro de cupons

Dado que eu acesse a documentação do serviço da EBAC-SHOP

E esteja autenticado  
Quando eu inserir o código do cupom <code>

E o valor de desconto <amount>

E o tipo de desconto <discount\_type>

E a descrição <description>

Então deve retornar status <status> com os dados do cupom criado

Exemplos:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| code | amount | discount\_type | description | status |
| “cupom10” | “10” | “fixed\_product” | “Cupom de desconto de 10%” | “201” |
| “cupom15” | “15” | “fixed\_product” | “Cupom de desconto de 15%” | “201” |

## Casos de testes

## História de usuário 1: Adicionar item ao carrinho

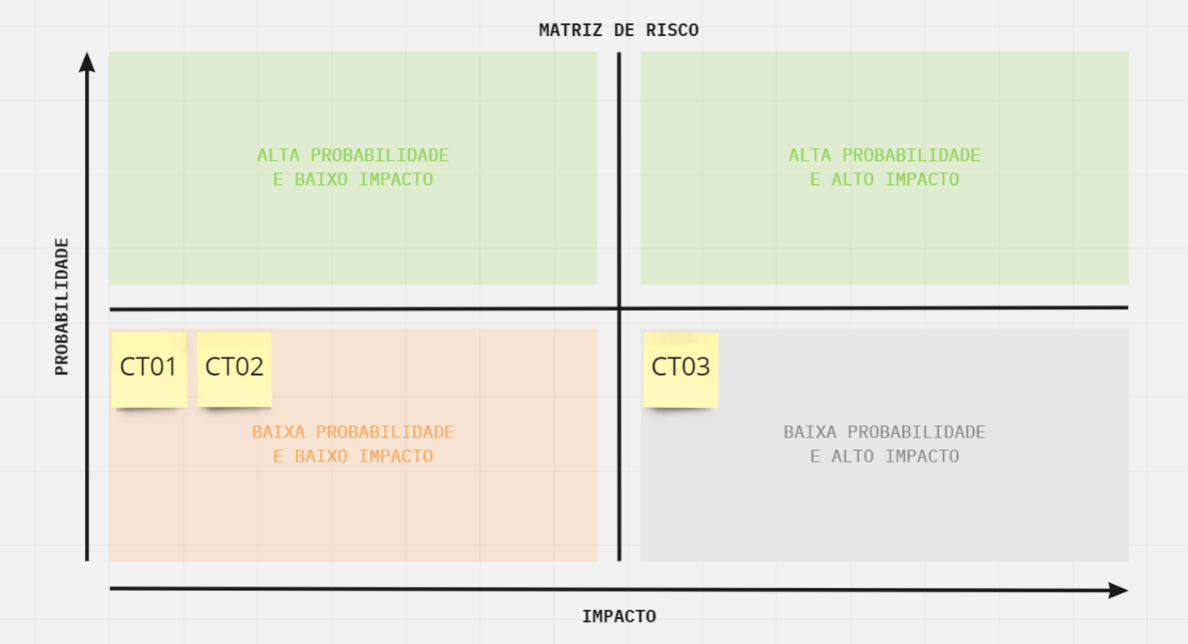
Técnicas de Testes:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PARTICIONAMENTO DE EQUIVALÊNCIA:  |  |  |  | | --- | --- | --- | | Regras | Entrada | Saída | | RN01 |  |  | | RN01 |  |  |   VALOR LIMITE:  |  |  |  | | --- | --- | --- | | Regras | Entrada | Saída | | RN01 |  |  | | RN01 |  |  | | RN01 | Adicionar 11 itens do mesmo produto | Inválido |   TABELA DE DECISÃO:   |  |  |  |  | | --- | --- | --- | --- | | Condições | Regra 1 | Regra 2 | Regra 3 | | > 10 | V | F | F | | < 10 | F | V | F | | 10 | F | F | V | | **Ações** |  |  |  | | Adicionar | - | X | X | |

|  |  |
| --- | --- |
| Cenários de testes |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RN | Título | Saída |
| CT01 | RN01 | Adicionar 9 itens do mesmo produto | válido |
| CT02 | RN01 | Adicionar 10 itens do mesmo produto | válido |
| CT03 | RN01 | Adicionar 11 itens do mesmo produto | inválido |

Matriz de Risco:



**Imagem 02 – Matriz de Risco – US-0001**

CT01: Adicionar 9 itens de um mesmo produto

|  |  |
| --- | --- |
| Descrição: | Teste para validar a adição de 9 itens de um mesmo produto |
| Pré-condições: | 9 itens do mesmo produto em estoque |
| Massa de dados / Parâmetros: | [Abominable Hoodie,](http://lojaebac.ebaconline.art.br/product/abominable-hoodie/?attribute_size=S&attribute_color=Blue) S, Blue, 9 |
| Prioridade: | Média |
| Ação | Resultado esperado |
| 1. Entrar na página de busca | -- |
| 1. Buscar pelo produto | -- |
| 1. Inserir o tamanho S | -- |
| 1. Inserir a cor Blue | -- |
| 1. Inserir 9 itens | -- |
| 1. Clicar em “Comprar” | Deve exibir uma mensagem: “Abominable Hoodie” foi adicionado no seu carrinho. |

CT02: Adicionar 10 itens de um mesmo produto

|  |  |
| --- | --- |
| Descrição: | Teste para validar a adição de 10 itens de um mesmo produto |
| Pré-condições: | 10 itens do mesmo produto em estoque |
| Massa de dados / Parâmetros: | [Abominable Hoodie,](http://lojaebac.ebaconline.art.br/product/abominable-hoodie/?attribute_size=S&attribute_color=Blue) S, Blue, 10 |
| Prioridade: | Média |
| Ação | Resultado esperado |
| 1. Entrar na página de busca | -- |
| 1. Buscar pelo produto | -- |
| 1. Inserir o tamanho S | -- |
| 1. Inserir a cor Blue | -- |
| 1. Inserir 10 itens | -- |
| 1. Clicar em “Comprar” | Deve exibir uma mensagem: “Abominable Hoodie” foi adicionado no seu carrinho. |

CT03: Adicionar 11 itens de um mesmo produto

|  |  |
| --- | --- |
| Descrição: | Teste para validar a adição de mais de 11 itens de um mesmo produto |
| Pré-condições: | 11 itens do mesmo produto em estoque |
| Massa de dados / Parâmetros: | [Abominable Hoodie,](http://lojaebac.ebaconline.art.br/product/abominable-hoodie/?attribute_size=S&attribute_color=Blue) S, Blue, 11 |
| Prioridade: | Média |
| Ação | Resultado esperado |
| 1. Entrar na página de busca | -- |
| 1. Buscar pelo produto | -- |
| 1. Inserir o tamanho S | -- |
| 1. Inserir a cor Blue | -- |
| 1. Inserir mais de 11 itens | -- |
| 1. Clicar em “Comprar” | Deve exibir uma mensagem de validação: “Não é permitido adicionar mais de 10 itens de um mesmo produto!” |

## História de usuário 2: Login na plataforma

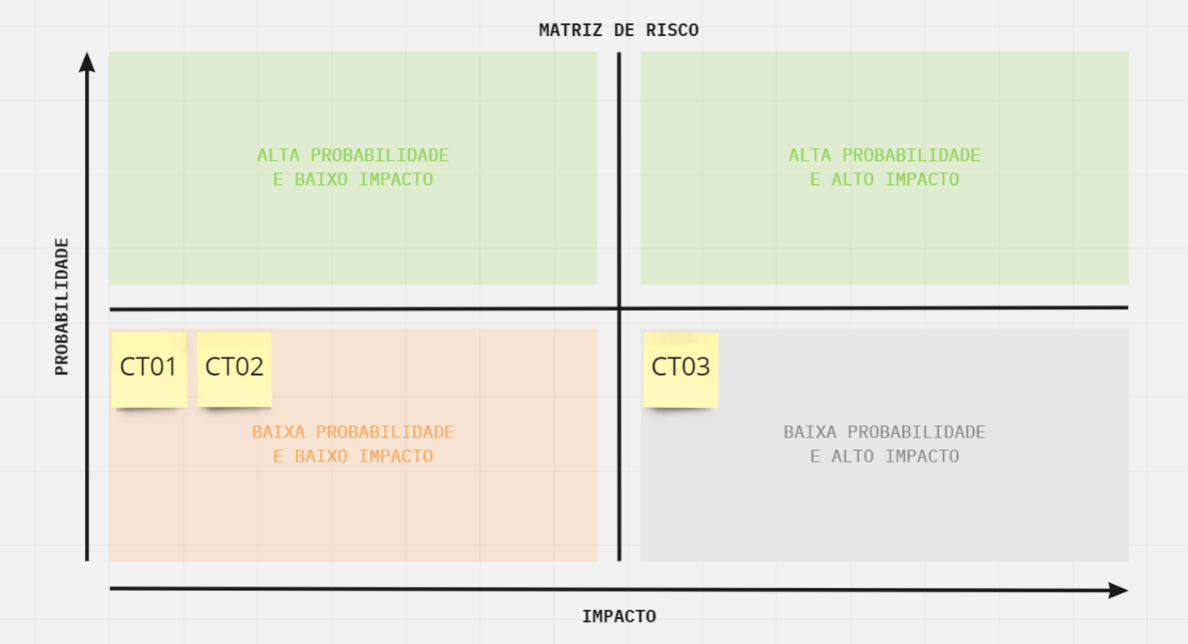
# Técnicas de Testes:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PARTICIONAMENTO DE EQUIVALÊNCIA:  |  |  |  | | --- | --- | --- | | Regras | Entrada | Saída | | RN04 |  |  | | RN04 |  |  |   VALOR LIMITE:  |  |  |  | | --- | --- | --- | | Regras | Entrada | Saída | | RN04 |  |  | | RN04 |  |  |   TABELA DE DECISÃO:   |  |  |  |  | | --- | --- | --- | --- | | Condições | Regra 1 | Regra 2 | Regra 3 | | 1 | V | F | F | | 2 | F | V | F | | 3 | F | F | V | | **Ações** |  |  |  | | Errar a senha | X | X | - | |

|  |  |
| --- | --- |
| Cenários de testes |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RN | Título | Saída |
| CT01 | RN04 | Errar a senha 1 vez | válido |
| CT02 | RN04 | Errar a senha 2 vezes | válido |
| CT03 | RN04 | Errar a senha 3 vezes | inválido |

Matriz de Risco:



**Imagem 03 – Matriz de Risco – US-0002**

CT01: Errar a senha 1 vez

|  |  |
| --- | --- |
| Descrição: | Teste para validar se o usuário errar 1 vez a senha, exibir mensagem de validação, mas não travar por 15 minutos o login |
| Pré-condições: | Usuário e senha cadastrados e ativo |
| Massa de dados / Parâmetros: | [felipe.dantas@ebac.com.br](mailto:felipe.dantas@ebac.com.br), senha@123 |
| Prioridade: | Média |
| Ação | Resultado esperado |
| 1. Entrar na página de login | -- |
| 1. Inserir o usuário | -- |
| 1. Inserir a senha | -- |
| 1. Clicar em “LOGIN” | Deve exibir a mensagem: “Erro: A senha fornecida para o e-mail felipe.dantas@ebac.com.br está incorreta. Perdeu a senha?” |

CT02: Errar a senha 2 vezes

|  |  |
| --- | --- |
| Descrição: | Teste para validar se o usuário errar 2 vezes a senha, exibir mensagem de validação, mas não travar por 15 minutos o login |
| Pré-condições: | Usuário e senha cadastrados e ativo |
| Massa de dados / Parâmetros: | [felipe.dantas@ebac.com.br](mailto:felipe.dantas@ebac.com.br), senha@123 |
| Prioridade: | Média |
| Ação | Resultado esperado |
| 1. Entrar na página de login | -- |
| 1. Inserir o usuário | -- |
| 1. Inserir a senha | -- |
| 1. Clicar em “LOGIN” | Deve exibir a mensagem: “Erro: A senha fornecida para o e-mail felipe.dantas@ebac.com.br está incorreta. Perdeu a senha?” |

CT03: Errar a senha 3 vezes

|  |  |
| --- | --- |
| Descrição: | Teste para validar se o usuário errar 3 vezes a senha, deve travar por 15 minutos o login |
| Pré-condições: | Usuário e senha cadastrados e ativo |
| Massa de dados / Parâmetros: | [felipe.dantas@ebac.com.br](mailto:felipe.dantas@ebac.com.br), senha@123 |
| Prioridade: | Média |
| Ação | Resultado esperado |
| 1. Entrar na página de login | -- |
| 1. Inserir o usuário | -- |
| 1. Inserir a senha | -- |
| 1. Clicar em “LOGIN” | Deve travar por 15 minutos o login |

## História de usuário 3: API de Cupons

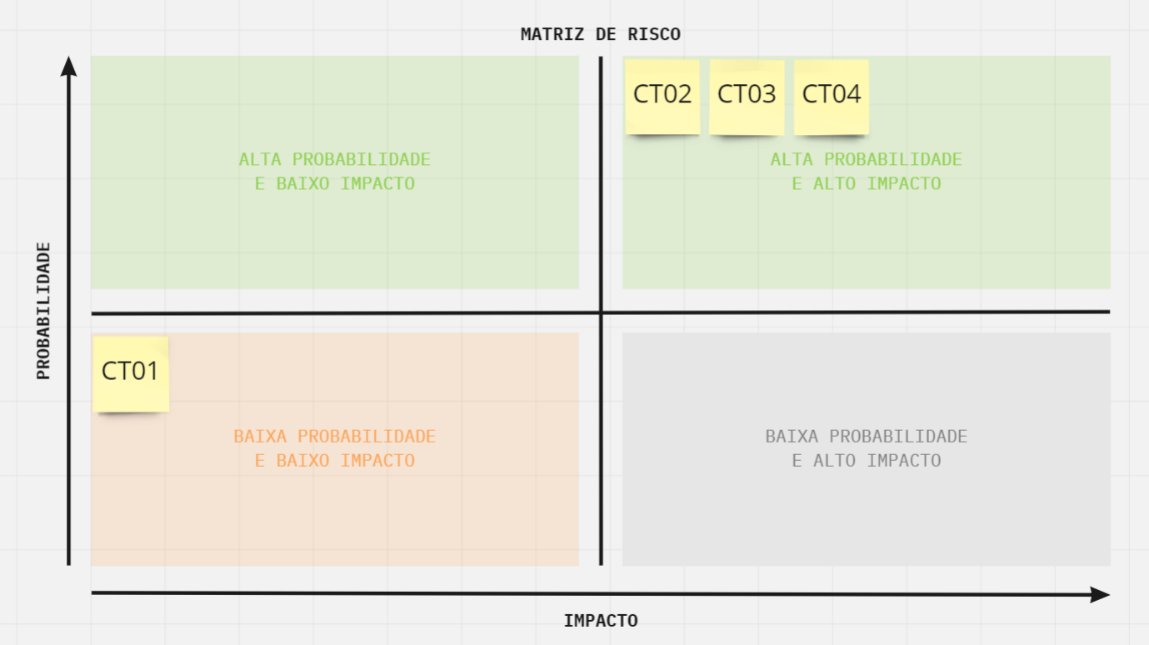
# Técnicas de Testes:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TABELA DE DECISÃO:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Condições | Regra 1 | Regra 2 | Regra 3 | Regra 4 | | Inseriu os campos obrigatórios? | V | F | F | V | | Nome do cupom é repetido? | F | V | F | V | | **Ações** |  |  |  |  | | Inserir | - | - | - | X | |

|  |  |
| --- | --- |
| Cenários de testes |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | RN | Título | Saída |
| CT01 | RN01/RN02 | Inseriu os campos obrigatórios sem repetir o nome do cupom | válido |
| CT02 | RN01/RN02 | Inseriu os campos obrigatórios e repetiu o nome do cupom | inválido |
| CT03 | RN01/RN02 | Não inseriu todos os campos obrigatórios sem repetir o nome do cupom | inválido |
| CT04 | RN01/RN02 | Não inseriu todos os campos obrigatórios e repetiu o nome do cupom | inválido |

Matriz de Risco:



**Imagem 04 – Matriz de Risco – US-0003**

CT01: Inseriu os campos obrigatórios sem repetir o nome do cupom

|  |  |
| --- | --- |
| Descrição: | Teste para validar na criação de cupom se os dados obrigatórios foram preenchidos sem repetir o nome do cupom |
| Pré-condições: | Acesso ao serviço (API de Cupons) e estar autenticado |
| Massa de dados / Parâmetros: | “code”: “cupomTeste”, “amount”: “10”, “discount\_type”: “fixed\_product”, “description”: “Descrição do cupom de teste” |
| Prioridade: | Média |
| Ação | Resultado esperado |
| 1. Acessar o serviço (API de Cupons) | -- |
| 1. Inserir o código (nome do cupom) | -- |
| 1. Inserir o valor de desconto | -- |
| 1. Inserir o tipo de desconto | -- |
| 1. Inserir a descrição | -- |
| 1. Clicar em “LOGIN” | Deve exibir o status “201” com os dados criados do cupom |

CT02: Inseriu os campos obrigatórios e repetiu o nome do cupom

|  |  |
| --- | --- |
| Descrição: | Teste para validar na criação de cupom se o nome do cupom já existe |
| Pré-condições: | Acesso ao serviço (API de Cupons) e estar autenticado |
| Massa de dados / Parâmetros: | “code”: “cupomTeste”, “amount”: “10”, “discount\_type”: “fixed\_product”, “description”: “Descrição do cupom de teste” |
| Prioridade: | Média |
| Ação | Resultado esperado |
| 1. Acessar o serviço (API de Cupons) | -- |
| 1. Inserir o código (nome do cupom) | -- |
| 1. Inserir o valor de desconto | -- |
| 1. Inserir o tipo de desconto | -- |
| 1. Inserir a descrição | -- |
| 1. Clicar em “LOGIN” | Deve exibir o status “400” com a mensagem “O código de cupom já existe” |

CT03: Não inseriu todos os campos obrigatórios sem repetir o nome do cupom

|  |  |
| --- | --- |
| Descrição: | Teste para validar na criação de cupom se os dados obrigatórios não foram preenchidos sem repetir o nome do cupom |
| Pré-condições: | Acesso ao serviço (API de Cupons) e estar autenticado |
| Massa de dados / Parâmetros: | “code”: “cupomTeste”, “amount”: “”, “discount\_type”: “”, “description”: “” |
| Prioridade: | Média |
| Ação | Resultado esperado |
| 1. Acessar o serviço (API de Cupons) | -- |
| 1. Inserir o código (nome do cupom) | -- |
| 1. Clicar em “LOGIN” | Deve exibir o status “422” com a mensagens: “O campo ‘amount’ é obrigatório”, “O campo ‘discount\_type’ é obrigatório”, “O campo ‘description’ é obrigatório” |

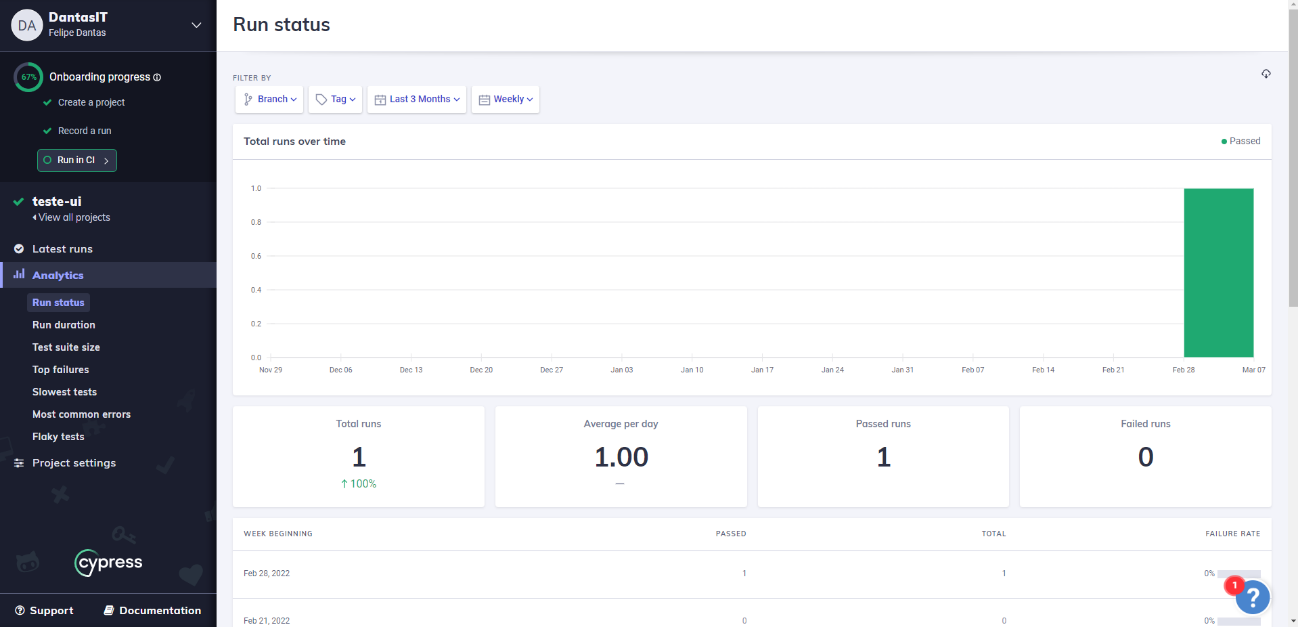
## Repositório no Github

Link do repositório: https://github.com/FelipeRBDantas/TCC-EBAC.git

## Testes automatizados

* + 1. Automação de UI:

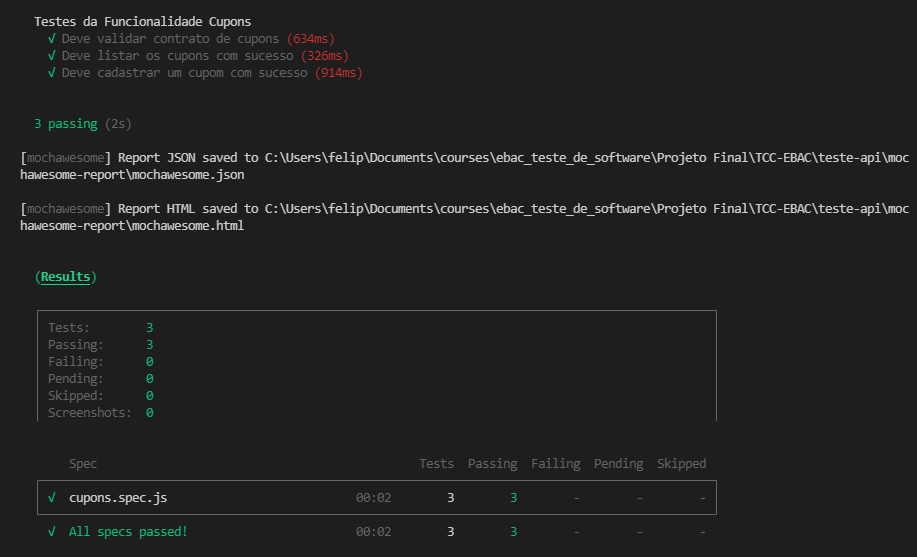
Projeto – Teste UI: <https://github.com/FelipeRBDantas/TCC-EBAC/teste-ui>

  
  
**Imagem 05 – Evidências – Cypress – UI - Dashboard**

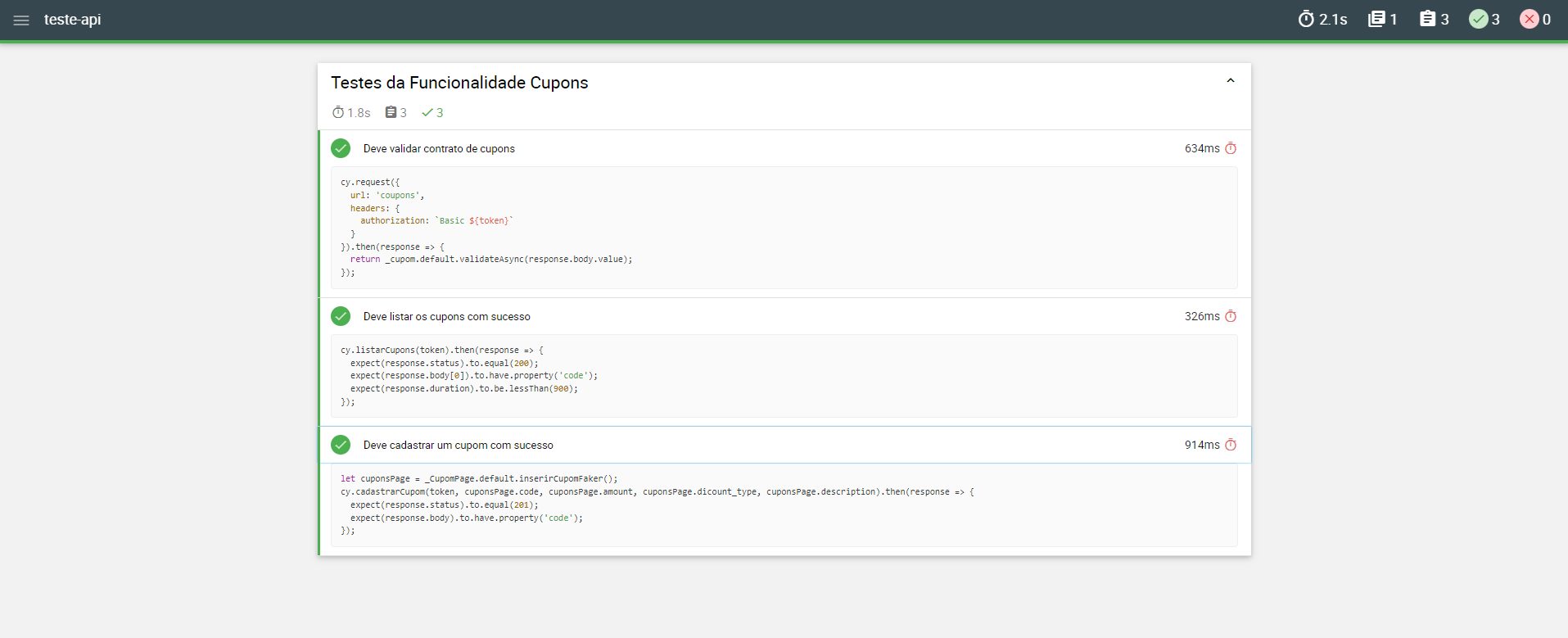
  
**Imagem 06 – Evidências – Cypress – UI – Gerar Testes e Vídeos**

* + 1. Automação de API

Projeto – Teste API: <https://github.com/FelipeRBDantas/TCC-EBAC/teste-api>



**Imagem 07 – Evidências – Cypress – API - Gerar Testes e Vídeos**



**Imagem 08 – Evidências – Cypress – API - Dashboard**

## Integração contínua

Jenkinsfile - UI: <https://github.com/FelipeRBDantas/TCC-EBAC/tree/main/teste-ui>

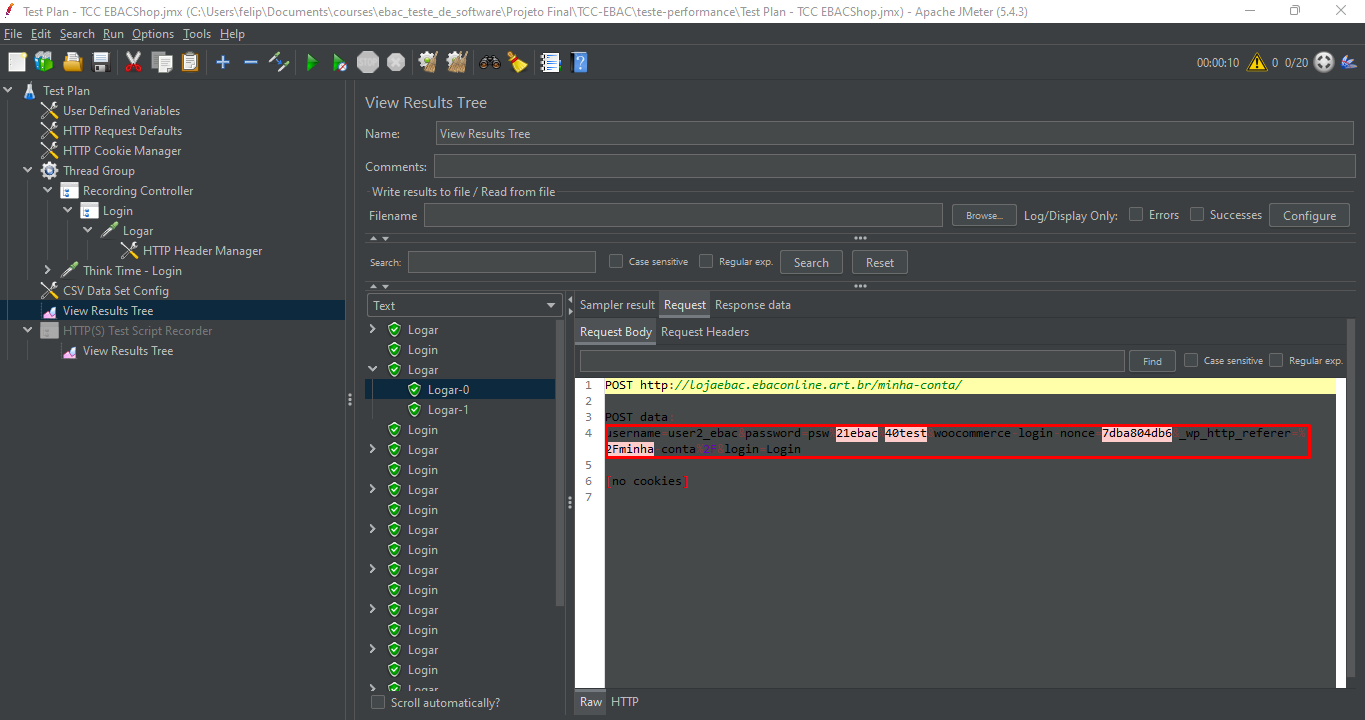
Jenkinsfile- API: <https://github.com/FelipeRBDantas/TCC-EBAC/tree/main/teste-api>



**Imagem 09 - Jenkins – Etapas de Integração Contínua – Testes UI e API**

## Testes de performance

Projeto – Teste Performance: <https://github.com/FelipeRBDantas/TCC-EBAC/teste-performance>



**Imagem 09 – JMeter – Teste de Performance com Template Recording**

# CONCLUSÃO

Em suma, posso dizer que com o decorrer deste presente trabalho pude ter a oportunidade de enrijecer cada vez mais o meu conhecimento e a minha experiência, seja com fundamentos, automatização e tudo que contribua para a carreira de teste de software.

E acredito também que com a concretização deste projeto, possa vir a ser um portfólio para futuras oportunidades no mercado de trabalho.

# REFERÊNCIAS BIBLIOGRÁFICAS

Miro (Ferramenta de Colaboração): <https://www.miro.com>

Qualidade de Produto de Software: http://www.bianchi.pro.br/edutec/qualsoft.php