



QUALIDADE DE SOFTWARE

Kelly Regina da Rosa Martins

Análise de Qualidade

Itapetininga/ SP

2023

1. RESUMO

Nesse projeto foi abordado o conhecimento passado nas aulas do curso de teste de software da EBAC, como estratégia de testes, como elabora critérios de aceite usando a linguagem Gherkin, a criar casos de teste, Automação com Cypress testes de UI e API a realizar um integração usando o Jenkins e usar o Jmeter na realização de testes de performance.

2. SUMÁRIO

1. RESUMO.....	2
2. SUMÁRIO.....	3
3. INTRODUÇÃO.....	4
4. O PROJETO.....	5
4.1 Estratégia de teste.....	5
4.2 Critérios de aceitação.....	5
4.2.1 História de usuário 1: [US-0001] - Adicionar item ao carrinho	5
4.2.2 História de usuário 2: [US-0002] - Login na plataforma.....	6
4.2.3 História de usuário 2: [US-0003] - API de cupons	6
4.3 Casos de testes.....	6
4.3.1 História de usuário 1.....	6
4.3.2 História de usuário 2.....	6
4.3.1 História de usuário 3: API de Cupom.....	7
4.4 Repositório no Github.....	7
4.5 Testes automatizados.....	7
4.6 Integração contínua.....	8
4.7 Testes de performance.....	8
5. CONCLUSÃO.....	9
6. REFERÊNCIAS BIBLIOGRÁFICAS.....	9

3. INTRODUÇÃO

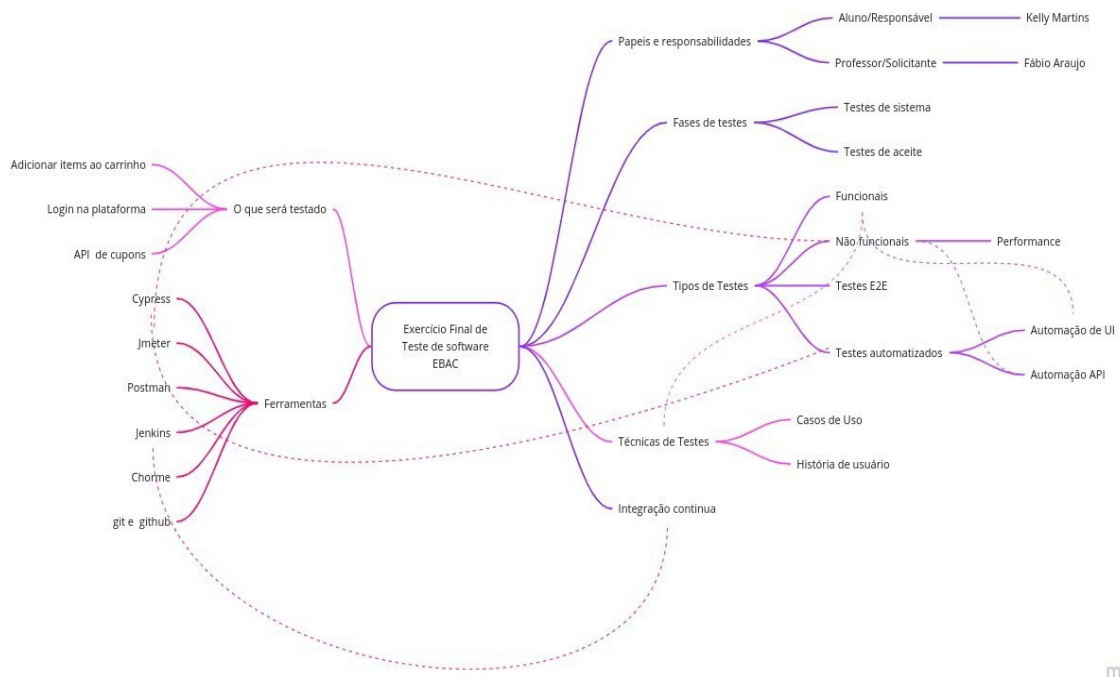
Esse trabalho tem com objetivo demonstra o fluxo de trabalho de um QA desde o planejamento até entrega, passando desde a elaboração da estratégia de testes e de critérios de aceites a automação de cenário e integração continua até sua entrega na plataforma do github

4. 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 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.

4.1 Estratégia de teste

- Faça uma estratégia de testes em um mapa mental, seguindo algumas diretrizes como objetivos, papéis 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:



4.2 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

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

Critérios de aceitação:

Dado que usuário acesse página de carrinho

Cenário 1: Quantidade excedida

Quando usuário adicionar 11 itens do mesmo produto ao carrinho

Então exibir um alerta "quantidade ultrapassou o limite de 10 produto"

Cenário 2: Limite de compra excedido

Quando a soma do carrinho do usuário ultrapassar R\$ 990.00

Então exibir mensagem "Limite de compra é de R\$ 990.00, retire um produto"

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

plataforma Critérios de aceitação:

Contexto:

Dado que usuário acesse a página de login da EBAC-SHOP

Cenário 1: Login válido

Quando Usuário informar usuário "kelly@ebacshop.com.br"

E senha "ebacshop123"

Então ir para página de checkout

Cenário 2: Usuário inválido

Quando usuário informar usuário "usuario@usuario.com

E senha "ebacshop123"

Então deve exibir uma mensagem de alerta "usuário inválido"

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

cupons Critérios de aceitação:

Contexto

Dado que usuário autenticado como admin

Cenário 1: Autenticação admin

Quando usuário informar usuário " admin_ebac"

E senha "@admin!&b@c!2022"

Então usuário admin poderá cadastrar os cupons de descontos

Cenário 2: Cadastro de cupom

Quando usuário admin informar o código do cupom <codigocupom>

E o valor do cupom

<valorcupom> **Então** cadastrar

cupom **Example:**

codigocupom	valorcupom
Ganhou10	10
Ganhou15	15

4.3 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

4.3.1 História de usuário 1:

CT01: Carrinho com 10 produtos diferentes e valor total ≤ 990.00 , finalizar compra

CT02: Carrinho com 10 produtos iguais e valor total ≤ 990.00 , não finalizar compra

CT03: Carrinho com 11 produtos diferentes e valor total ≤ 990.00 , não finalizar compra

4.3.2 História de usuário 2:

CT01: Login com usuário válido e senha válido, deve permitir o acesso

CT02: Login com usuário válido e senha inválida, não deve permitir acesso

CT03: Login com usuário inválido e senha válida, não deve permitir acesso

4.3.1 História de usuário 3: API de Cupom

CT01: Login com usuário admin válido e senha válida, autenticação permitida

CT02: Login com usuário não admin válida e senha válida, autenticação não permitida

CT03: Login com usuário admin inválido e senha válida, autenticação não permitida

4.4 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: <https://github.com/kellyrrmartins/tcc-ebac>

4.5 Testes automatizados

4.5.1 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.

4.5.2 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

4.6 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

4.7 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

```

<input type="checkbox"/>	Nome de usuário	Nome	E-mail	Função
<input type="checkbox"/>	 user1_ebac	—	user1_ebac@ebac.com	Assinante
<input type="checkbox"/>	 user2_ebac	—	user2_ebac@ebac.com	Assinante
<input type="checkbox"/>	 user3_ebac	—	user3_ebac@ebac.com	Assinante
<input type="checkbox"/>	 user4_ebac	—	user4_ebac@ebac.com	Assinante
<input type="checkbox"/>	 user5_ebac	—	user5_ebac@ebac.com	Assinante

- 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:

HTTP Request

Name: -31

Comments: Detected the start of a redirect chain

Basic Advanced

Web Server

Protocol [http]: Server Name or IP:

HTTP Request

POST Path: /minha-conta/

☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data ☐ Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value	UP
username	\$(usuario)	
password	\$(senha)	
woocommerce-login-nonce	0c0130c6c0	
_wp_http_referer	/minha-conta/	
login	Login	

5. CONCLUSÃO

Ao finalizar esse trabalhos alias ao finalizar esse curso e fazendo um review, tenho sim muito o que aprender, mais já consigo realizar um teste de software, realizar casos de testes, possui um conhecimento em mtodologias ageis, e como todo a caminha quando se chega ao final é na verdade o começo de tudo

6. REFERÊNCIAS BIBLIOGRÁFICAS

ESCOLA EBAC, curso Teste de Software

SITE www.atlassian.com/br/git/tutorials/git-submodule Submódulos do Git