



**QUALIDADE DE SOFTWARE**

**PATRICIA CRISTINA SAMPAIO ARAGÃO**

Análise de Qualidade

Manaus - AM

2022

## **1. RESUMO**

Os testes de software são essenciais durante o desenvolvimento de um sistema de software. São os testes que irão verificar se todas as funcionalidades que o sistema deve ter foram implementadas corretamente e também buscam encontrar falhas antes da liberação para o usuário final. Neste trabalho, será possível observar uma análise de Qualidade realizada a partir de 3 Histórias de Usuários de um sistema: Adicionar item ao carrinho, login na plataforma e API de cupons. Serão elaboradas estratégias e casos de testes, e também trará testes automatizados e de performance.

## 2. SUMÁRIO

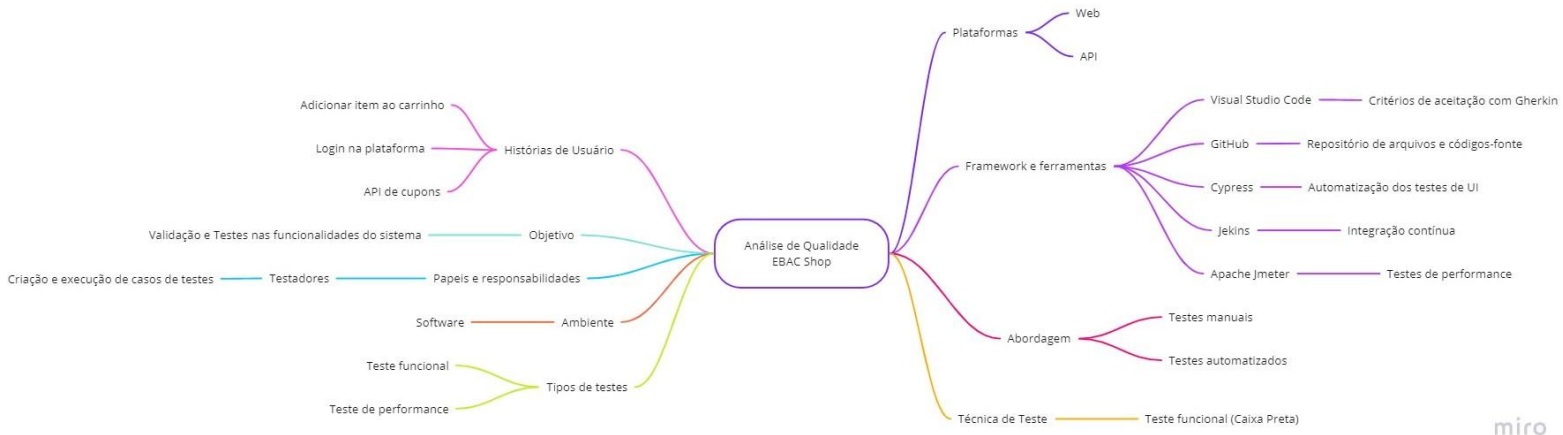
<b>1. RESUMO .....</b>	<b>2</b>
<b>2. SUMÁRIO.....</b>	<b>3</b>
<b>3. INTRODUÇÃO.....</b>	<b>4</b>
<b>4. O PROJETO.....</b>	<b>5</b>
4.1 Estratégia de teste .....	5
4.2 Critérios de aceitação.....	5
4.2.1 História de usuário 1: [US-0001] – Adicionar item ao carrinho .....	5
4.2.2 História de usuário 2: [US-0002] – Login na plataforma .....	6
4.2.3 História de usuário 3: [US-0003] – API de cupons.....	7
4.3 Casos de testes .....	8
4.3.1 História de usuário 1: .....	8
4.3.2 História de usuário 2: .....	9
4.3.1 História de usuário 3: API de Cupom .....	9
4.4 Repositório no Github .....	10
4.5 Testes automatizados .....	10
4.6 Integração contínua .....	11
4.7 Testes de performance.....	11
<b>5. CONCLUSÃO.....</b>	<b>12</b>
<b>6. REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>12</b>

### **3. INTRODUÇÃO**

Este trabalho é uma análise de Qualidade realizado no sistema EBAC Shop, que será realizado em 3 funcionalidades principais do sistema: a primeiro é a de adicionar produtos no carrinho, onde o usuário adicionará todos os itens de seu interesse para depois finalizar a compra dos itens e onde haverá algumas restrições em cada compra e cupons de desconto; a segunda funcionalidade é a de login na plataforma, que tem validação de mensagem e um limite de tentativas para conseguir logar; e por fim, a funcionalidade de API de cupons, que permite cadastro e listagem dos cupons. Para isso, serão elaboradas estratégias de testes, critérios de aceitação com Gherkin, casos de testes que estarão disponíveis no GitHub e testes automatizados de UI, API, integração contínua e de performance.

## 4. O PROJETO

### 4.1 Estratégia de teste



### 4.2 Critérios de aceitação

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

Critérios de aceitação:

- 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%.

```
#language: pt
```

Funcionalidade: Adicionar item ao carrinho

Como cliente da EBAC-SHOP

Quero adicionar produtos no carrinho

Para realizar a compra dos itens

Cenário: Inserir mais de 10 itens do mesmo produto

Dado que eu acesse o sistema EBAC-SHOP

E eu adiciono 10 itens do mesmo produto ao carrinho

Quando eu clicar para adicionar mais 1 item no mesmo produto

Então a ação não será permitida

E uma mensagem de erro será exibida

Esquema do Cenário: Receber cupom

Dado que eu acesse o sistema EBAC-SHOP

E adicione itens ao carrinho

Quando o valor da minha compra for <valor>

Então devo ganhar <desconto> % de desconto

valor	desconto
150	0
205	10
400	10
580	10
690	15

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

Critérios de aceitação:

- 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.

```
#language: pt
```

Funcionalidade: Login na plataforma

Como cliente da EBAC-SHOP

Quero fazer o login na plataforma

Para visualizar meus pedidos

Cenário: Senha inválida

Dado que eu acesse o sistema EBAC-SHOP na seção de login

Quando eu inserir um login válido

E uma senha inválida

Então uma mensagem de erro será exibida

Esquema do Cenário: Login por e-mail, login ou cpf

Dado que eu acesse o sistema EBAC-SHOP na seção de login

Quando eu inserir o login <login>

E a senha <senha>

Então deve ser permitido acesso ao sistema

login	senha	
patricia@gmail.com	Eb@c2022	
Patricia	Eb@c2022	
123.456.789-01	Eb@c2022	

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

Critérios de aceitação:

- Deve listar todos os cupons cadastrado ou listar buscando por ID do cupom;
- 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.

```
#language: pt
```

Funcionalidade: API de Cupons

Como admin da EBAC-SHOP

Quero criar um serviço de cupom

Para poder listar e cadastrar os cupons

Cenário: Buscar cupom por ID

Dado que eu acesse a API de cupons

Quando eu informar um ID de cupom no método GET

Então as informações do cupom devem ser listadas

Esquema do Cenário: Cadastrar cupom

Dado que eu acesse a API de cupons

E acesse o método POST

E informe o código do cupom <codigo>

E o valor <valor>

E o tipo de desconto <desconto>

E a descrição <descricao>

Quando eu clicar para cadastrar

Então o cupom deve ser cadastrado corretamente

codigo	valor	desconto	descricao
Ganhe10	10.00	fixed_product	Cupom de teste
Desconto15	15.00	fixed_product	Desconto de R\$15.00
Mega50	50.00	fixed_product	Desconto de R\$50.00
Brinde2	2.00	fixed_product	Desconto de R\$2.00

## 4.3 Casos de testes

### 4.3.1 História de usuário 1:

CT01: Inserir 9 unidades do mesmo produto no carrinho, deve permitir a realização da compra;

CT02: Inserir 10 unidades do mesmo produto no carrinho, deve permitir a realização da compra;

CT03: Inserir 11 unidades do mesmo produto no carrinho, não deve permitir a realização da compra e exibir mensagem de erro;

CT04: Inserir produtos até o valor R\$989,99, deve permitir a finalização da compra;

CT05: Inserir produtos até o valor R\$990,00, deve permitir a finalização da compra;

CT06: Inserir produtos até o valor R\$990,10, não deve permitir a finalização da compra e exibir uma mensagem de alerta;



CT07: Compras até R\$199,99 não devem receber desconto;  
CT08: Compras de R\$200,00 devem receber 10% de desconto;  
CT09: Compras de R\$400,00 devem receber 10% de desconto;  
CT10: Compras de R\$600,00 devem receber 10% de desconto;  
CT09: Compras de R\$601,00 devem receber 15% de desconto.

#### **4.3.2 História de usuário 2:**

CT01: Ao logar com um usuário ativo, com login e senha corretos, deve ser possível visualizar pedidos;  
CT02: Ao inserir um login correto e senha incorreta, deve exibir uma mensagem de erro;  
CT03: Ao inserir um login incorreto e senha correta, deve exibir uma mensagem de erro;  
CT04: Ao inserir um login incorreto e senha incorreta, deve exibir uma mensagem de erro;  
CT05: Ao inserir um login incorreto e senha incorreta, deve exibir uma mensagem de erro;  
CT06: Ao inserir um login com formato de e-mail e senha correta, deve permitir acesso ao sistema;  
CT07: Ao inserir um login de nome e senha correta, deve permitir acesso ao sistema;  
CT08: Ao inserir um login com formato de CPF e senha correta, deve permitir acesso ao sistema;  
CT09: Ao inserir um login ou senha incorretos por mais de 3 vezes, não deve ser possível fazer outra tentativa pelos próximos 15 minutos.

#### **4.3.3 História de usuário 3: API de Cupom**

CT01: Ao ir no método GET e não passar parâmetro, ao clicar para listar, devem ser retornados todos os cupons cadastrados;  
CT02: Ao ir no método GET e passar um ID como parâmetro, ao clicar para listar, ser listado apenas o cupom com o ID informado;  
CT03: Ao ir no método POST e informar corretamente o Código do cupom, valor, tipo de desconto e descrição como parâmetros, ao clicar para cadastrar o cupom, ser cadastrado com sucesso;

CT04: Ao ir no método POST e informar corretamente o Código do cupom, valor e tipo de desconto, mas não informar a descrição, ao clicar para cadastrar o cupom, deve ser exibida mensagem de alerta para informar campo obrigatório;

CT05: Ao ir no método POST e informar corretamente o Código do cupom, valor e descrição, mas não informar o tipo de desconto, ao clicar para cadastrar o cupom, deve ser exibida mensagem de alerta para informar campo obrigatório;

CT06: Ao ir no método POST e informar corretamente o Código do cupom, tipo de desconto e descrição, mas não informar o valor, ao clicar para cadastrar o cupom, deve ser exibida mensagem de alerta para informar campo obrigatório;

CT07: Ao ir no método POST e informar corretamente o valor, tipo de desconto e descrição, mas não informar o Código do cupom, ao clicar para cadastrar o cupom, deve ser exibida mensagem de alerta para informar campo obrigatório;

CT08: Ao ir no método POST e tentar cadastrar cupom com nome igual a um já previamente cadastrado, deve ser exibida mensagem de erro informando que não pode repetir o nome.

#### 4.4 Repositório no Github

- Link do repositório: <https://github.com/PatriciaCris/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.
- **Link da atividade no repositório:**  
**<https://github.com/PatriciaCris/TCC-EBAC/tree/main/Atividades%20TCC/UI>**

##### 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.

- **Link da atividade no repositório:**  
**<https://github.com/PatriciaCris/TCC-EBAC/tree/main/Atividades%20TCC/API>**

#### 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.
- **Link do repositório:** **<https://github.com/PatriciaCris/TCC-EBAC/tree/main/Atividades%20TCC/API>**

#### 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;
- 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 usados. Substitua esses parâmetros pela sua massa de dados, conforme aprendido em aula:

The screenshot shows an HTTP Request configuration window. The 'Name' field is set to '-31'. The 'Comments' field contains 'Detected the start of a redirect chain'. The 'Basic' tab is selected. The 'Web Server' section shows 'Protocol [http:]' and 'Server Name or IP:'. The 'HTTP Request' section shows 'POST' as the method and '/minha-conta/' as the path. Below this, there are checkboxes for 'Redirect Automatically', 'Follow Redirects' (checked), 'Use KeepAlive' (checked), 'Use multipart/form-data', and 'Browser-compatible headers'. The 'Parameters' tab is active, showing a table of parameters to be sent with the request.

Name:	Value	UP
username	\$(usuario)	
password	\$(senha)	
woocommerce-login-nonce	6c0b023c0	
_wp_http_referer	/minha-conta/	
login	Login	

- Link da atividade no repositório: <https://github.com/PatriciaCris/TCC-EBAC/tree/main/Atividades%20TCC/Teste%20Performance%20TCC>

## 5. CONCLUSÃO

Este trabalho de conclusão do Curso de Teste de Software da plataforma EBAC foi um compilado de tudo o que foi estudado ao longo dos capítulos e foi uma oportunidade de reforçar ainda mais os conhecimentos aprendidos. O curso foi acrescentou bastante aos conhecimentos e irá agregar ao trabalho de testes executados no dia-a-dia, com foco aos testes automatizados, que é de grande importância no mercado atual e que é capaz de agilizar os testes. Por fim, foi uma oportunidade de conhecer outros profissionais da área e compartilhar experiências.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

EBAC – Escola Britânica de Artes Criativas & Tecnologia. **Curso Teste de Software**. Disponível em: <https://lms.ebaconline.com.br/courses/6b500edd-7012-4bb5-9e6c-3af6bd7e9806/>. Acesso em: 20 nov. 2022.