



QUALIDADE DE SOFTWARE

Willian Almeida Alves de Souza

Análise de Qualidade

Londrina

2023

1. RESUMO

Este é um projeto de análise de qualidade do site EBAC Shop a partir da realização de testes utilizando diferentes ferramentas e métodos a fim de averiguar como o site corresponde à adição de produtos ao carrinho de compra, como corresponde a autenticação de login, o cadastro de cupons de desconto, além do desempenho na realização de logins em massa.

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.....	6
4.2.1 História de usuário 1: [US-0001] – Adicionar item ao carrinho.....	6
4.2.2 História de usuário 2: [US-0002] – Login na plataforma	7
4.2.3 História de usuário 2: [US-0003] – API de cupons.....	7
4.3 Casos de testes.....	7
4.3.1 História de usuário 1:	8
4.3.2 História de usuário 2:	8
4.3.1 História de usuário 3: API de Cupom	8
4.4 Repositório no Github.....	8
4.5 Testes automatizados	9
4.6 Integração contínua	9
4.7 Testes de performance	9
5. CONCLUSÃO	10
6. REFERÊNCIAS BIBLIOGRÁFICAS.....	10

3. INTRODUÇÃO

Considerando a necessidade de manter o bom funcionamento de uma plataforma, ela precisa ser testada regularmente. Nesse projeto será testada a plataforma da loja virtual EBAC Shop. Utilizaremos de diferentes métodos e ferramentas para averiguar as histórias de usuários indicadas.

Será destacada a estratégia traçada, os critérios de aceitação de cada história de usuário bem como os casos de testes correspondentes a cada uma delas. E além deles serão realizados testes automatizados das respectivas situações, bem como um teste de performance de uma delas.

Os testes automatizados da história de usuário “Adicionar item ao carrinho” e “API Cupons” serão realizados através das ferramentas Cypress e Jenkins. Enquanto para a história “Login na plataforma” será realizado o teste de performance através da plataforma JMeter.

4. O PROJETO

O Projeto consiste em acessar a loja virtual EBAC Shop, analisar e testar três histórias de usuários pré-definidas utilizando-se de ferramentas e métodos específicos em cada uma delas. A primeira história de usuário trata de adicionar itens ao carrinho de compras, a segunda da validação do login na plataforma e a terceira consiste no cadastro de cupons de desconto.

Antes de colocar em prática os respectivos testes, é preciso traçar a estratégia para cada um deles e definir qual o melhor caminho trilhar a fim de obter os melhores resultados possíveis a partir do que já é esperado para cada situação.

4.1 Estratégia de teste

- [US-0001] – Adicionar item ao carrinho: a abordagem manual deste teste consiste basicamente em acessar o site, escolher um produto e adicioná-lo ao carrinho de compras. O mesmo procedimento pode ser realizado de forma automatizada através da plataforma Cypress. O resultado esperado para tal teste é que desde que os produtos escolhidos estejam disponíveis no estoque, eles serão adicionados ao carrinho de compras de acordo com as características e quantidade solicitadas pelo usuário;
- [US-0002] – Login na plataforma: acessando a tela de login do usuário na loja virtual, deve autenticar o acesso através do nome de usuário ou email e a respectiva senha. Para isso, o teste a ser realizado será o do preenchimento desses mesmos campos e confirmação da autenticação do acesso. Considerando a possibilidade de controle dos logins a serem testado e assim a utilização de vários deles para teste, para essa história de usuário é possível realizar o teste de performance com massa de dados, esperando que a plataforma responda com a autenticação de todos os cadastros testados.
- [US-0003] – API de Cupons: considerando agora um usuário interno da plataforma que deva ser capaz de gerenciar os cupons de desconto da loja. Para a realização do teste é necessária a utilização do sistema API e respeitar as regras e comandos já especificados para criação e listagem dos cupons. No caso, os cupons podem ser pesquisados por ID ou código e o cadastro deve ter conter obrigatoriamente código, valor, tipo de desconto e descrição.



4.2 Critérios de aceitação

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

Critérios de aceitação:

Cenário 1: Item adicionado com sucesso

Dado que eu escolha o <produto>

E defina cor, tamanho e quantidade

Quando clicar em “Comprar”

Então deve exibir a mensagem “<produto> foi adicionado ao seu carrinho”

Cenário 2: Produto indisponível

Dado que eu escolha o <produto> e suas especificações

E não constar o produto em estoque

Então botão “Comprar” permanecerá desabilitado

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

Critérios de aceitação:

Cenário 1: Autenticação válida

Quando eu digitar o <email> ou <usuário>

E a <senha>

Então deve exibir uma mensagem de boas-vindas "Olá, <nome>"

email	usuário	senha	nome
joao@ebac.com.br	joaoEBAC	12345	João
maria@ebac.com.br	maria123	54321	Maria
willian@gmail.com	willianTeste	teste123	Willian
fulano@hotmail.com	ful@no	senh@	Fulano

Cenário 2: Usuário inexistente

Quando eu digitar o usuário "xxxyyyzzz@ebac.com.br"

E a senha "senha@123"

Então deve exibir uma mensagem de alerta "Usuário inexistente."

Cenário 3: Usuário com senha inválida

Quando eu digitar o usuário "joao@ebac.com.br"

E a senha "123456"

Então deve exibir uma mensagem de alerta "Usuário ou senha inválidos."

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

Critérios de aceitação:

Cenário 1: Cadastro de novo cupom

Dado que eu digite os campos obrigatórios do cupom

E no formato válido

Então deve apresentar código de resposta 200 e descrição "OK"

Cenário 2: Nome de cupom repetido

Dado que eu digite o nome de um cupom já existente

Quando enviar o comando GET

Então deve apresentar código de resposta 400 e descrição "Bad request"

4.3 Casos de testes

4.3.1 História de usuário 1:

CT01: Ao inserir até 10 itens de um mesmo produto no carrinho deve exibir mensagem de “Produto adicionado ao carrinho”.

CT02: Ao tentar inserir mais que 10 itens de um mesmo produto no carrinho, não deve habilitar o botão “Comprar”.

CT03: Caso a soma do valor do produto a ser adicionado no carrinho com o dos produtos já inseridos for maior do que R\$ 990,00 não deve habilitar o botão “Comprar”.

CT04: Caso a soma total do carrinho for de até R\$199,99, não deve ser liberado nenhum cupom de desconto.

CT05: Caso a soma total do carrinho for entre R\$200 e R\$600, deve ser liberado o cupom de desconto de 10%.

CT06: Caso a soma total do carrinho for entre R\$600,01 e R\$990,00 deve ser liberado o cupom de desconto de 15%.

4.3.2 História de usuário 2:

CT01: Ao inserir login e senha corretos, a plataforma deve permitir visualizar os pedidos realizados.

CT02: Ao inserir login e senha incorretos, deve exibir mensagem de erro.

CT03: Caso digite a senha incorreta por 3 vezes seguidas, não deve permitir realizar o login durante 15min.

4.3.1 História de usuário 3: API de Cupom

CT01: Ao digitar o comando GET, permitir visualizar todos os cupons cadastrados.

CT02: Ao digitar todos os campos obrigatórios de cadastro do cupom, deve aparecer mensagem de confirmação do cadastro.

CT03: Caso o nome do cupom já esteja cadastrado, não permitir o cadastro repetido.

4.4 Repositório no Github

- Link do repositório: <https://github.com/willian2as/TCC-EBAC.git>

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)  
  })  
});
```

4.6 Integração contínua






- Coloque os testes automatizados na integração contínua com jenkins, criando um job para execução da sua automação;

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

5. CONCLUSÃO

A realização dos testes solicitados permitiu explorar diferentes ferramentas e métodos e estratégias mesmo se tratando de apenas uma plataforma em questão. Logo, nota-se quão amplo pode ser o ambiente de testes, pois a cada função explorada dentro de uma plataforma é possível que novos questionamentos sejam analisados e testados e assim criando uma gama de ramificações já que no final das contas, está tudo interligado.

Para garantir o bom funcionamento das funções sejam elas de um software, plataforma, site ou mesmo processos de trabalho é preciso analisar a situação em questão definir uma estratégia do que deve ser observado e testado em busca de resultados que satisfaçam todos os usuários envolvidos no processo.

6. REFERÊNCIAS BIBLIOGRÁFICAS

Apache JMeter – Apache JMeter. Disponível em: <<https://jmeter.apache.org/>>. Acesso em: 29 dez. 2023.

EBAC – Shop – API. Disponível em: <http://lojaebac.ebaconline.art.br/rest-api/docs/#/coupons/post_wc_v3_coupons>. Acesso em: 29 dez. 2023.

EBAC – Shop – Página de testes. Disponível em:
<<http://lojaebac.ebaconline.art.br/>>. Acesso em: 29 dez. 2023.

JavaScript Component Testing and E2E Testing Framework. Cypress.
Disponível em: <<https://www.cypress.io/>>. Acesso em: 29 dez. 2023.

Jenkins. Disponível em: <<https://www.jenkins.io/>>. Acesso em: 29 dez. 2023.

Software de criação de mapas mentais: mapas mentais. MIND meister.
Disponível em: <<https://www.mindmeister.com/app/maps/public>>. Acesso em:
29 dez. 2023.