



QUALIDADE DE SOFTWARE

Lucas Helvidio Piovezan Rocon

Análise de Qualidade

Vila Velha

2023

1. RESUMO

Neste projeto será demonstrado algumas metodologias de teste e qualidade de software; tais

- Testes automatizados em UI e API;
- Integração contínua;
- Testes de performance;
- Técnicas de teste;

Para isso, documentei e utilizei uma plataforma específica como exemplo durante o projeto.

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	6
4.2.3 História de usuário 2: [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
5. CONCLUSÃO	12
6. REFERÊNCIAS BIBLIOGRÁFICAS.....	12

3. INTRODUÇÃO

Esse documento aborda as estratégias de teste e o uso delas na criação de cenários de teste, embasada nas histórias de usuário.

E tem como objetivo demonstrar os benefícios de adotar as práticas de qualidade em um time de desenvolvimento de software e quanto isso agrega valor ao negócio.

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:

Cenário 1: Teste de aplicação de Cupom

Dado que meu carrinho de compras possui <total> de R\$ em produtos
Quando eu acessar a tela de pagamento

Então será aplicado um <cupom> de <?%> de desconto

Total	Cupom
R\$240	10%
R\$700	15%
R\$100	0% - N/A

Cenário 2: Teste de quantidade de produtos

Dado que estou na página do produto

Quando eu adicionar uma quantidade inválida de produtos no carrinho

Então será exibido um modal de alerta ao cliente

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

Critérios de aceitação:

Usuário Ativo: O sistema deve permitir que apenas usuários ativos façam login na plataforma. Os usuários inativos não devem ser capazes de fazer login.

Exibir Mensagem de Erro: Quando um usuário errar o login ou a senha, o sistema deve exibir uma mensagem de erro clara e informativa, indicando que houve um erro no processo de autenticação.

Cenário 1: Validação de login na plataforma

Dado que o usuário esteja na página de Login

Quando o usuário preencher o campo de login com um <usuário>

E preencher o campo de senha com uma <senha>

Então será validado se a autenticação é válida

E o login será sucedido

OU retornará uma mensagem de erro

<USUÁRIO>	<SENHA>	RESULTADO
lucas.rocon	Teste123	Login sucedido
188742rocon	@123465	Usuário ou senha inválidos!
lucas.rocon	teste@3562	Usuário ou senha inválidos!

Cenário 2: Validação de travamento no login

Critério de aceitação:

Travamento após 3 Tentativas Erradas: Se o usuário errar a senha em três tentativas consecutivas, o sistema deve travar o login desse usuário por um período de 15 minutos. Durante esse tempo, o usuário não deve poder fazer novas tentativas de login.

Dado que o usuário está na tela de login

Quando o usuário errar a senha três vezes consecutivas

Então o usuário será bloqueado respeitando o critério de aceitação

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

Critérios de aceitação:

Cenário 1: Não permitir cadastrar um cupom repetido

Dado que o usuário seja um administrador

Quando o usuário requisitar um POST com o nome do cupom já existente

Então a mensagem de resposta deve mostrar “O código de cupom já existe”

Cenário 2: Cupons cadastrados

Dado que eu seja um usuário administrador

Quando eu requisitar um GET usando o usuário <usuario>

E a senha <senha>

Então o código de status da resposta deve ser <resposta>

<Usuário>	<Senha>	<Resposta>
admin_ebac	@admin!&b@c!2022	200 OK
admin_ebac	@admin!&b@c!2023	400 Bad Request
admin_rocon	@testeteste	400 Bad Request

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:

CAMINHO FELIZ:

CT01: Adicionar Produto ao Carrinho com Valor Dentro do Limite

Ação: Adicionar um produto de R\$ 150,00 ao carrinho.

Resultado Esperado: O produto é adicionado ao carrinho sem problemas.

Caminhos Alternativos/Negativos:

CT02: Tentar Adicionar Produto com Valor Acima de R\$ 990,00

Ação: Adicionar um produto de R\$ 1.000,00 ao carrinho.

Resultado Esperado: Deve exibir uma mensagem de erro informando que os valores não podem ultrapassar R\$ 990,00.

CT03: Tentar Adicionar Produto com Valor Entre R\$ 200 e R\$ 600 sem Cupom

Ação: Adicionar produtos no valor total de R\$ 400,00 ao carrinho sem aplicar um cupom.

Resultado Esperado: Deve exibir uma mensagem de alerta informando que valores entre R\$ 200,00 e R\$ 600,00 ganham um cupom de 10%. Perguntar se o usuário deseja aplicar o cupom.

4.3.2 História de usuário 2:

CT01: Login Bem-Sucedido com E-mail

Ação: Realizar login com um e-mail e senha válidos.

Resultado Esperado: Deve permitir o login e redirecionar o usuário para a visualização de pedidos.

CT02: Login com Credenciais Incorretas

Ação: Tentar fazer login com credenciais (e-mail, nome de usuário ou CPF) incorretas.

Resultado Esperado: Deve exibir uma mensagem de erro informando que as credenciais estão incorretas.

CT03: Login com Usuário Inativo

Ação: Tentar fazer login com um usuário que está marcado como inativo.

Resultado Esperado: Deve exibir uma mensagem de erro informando que somente usuários ativos podem fazer login.

4.3.1 História de usuário 3: API de Cupom

CT01: Listar Todos os Cupons

Ação: Fazer uma solicitação GET para listar todos os cupons.

Resultado Esperado: Deve retornar a lista de todos os cupons cadastrados com sucesso.

CT02: Cadastrar Novo Cupom com Campos Obrigatórios

Ação: Fazer uma solicitação POST para cadastrar um novo cupom com os campos obrigatórios preenchidos corretamente.

Resultado Esperado: Deve cadastrar o cupom com sucesso.

CT03: Tentar Cadastrar Cupom com Código Repetido

Ação: Tentar cadastrar um novo cupom com um código que já existe.

Resultado Esperado: Deve retornar uma mensagem de erro informando que o nome do cupom não pode ser repetido.

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/Irocon18/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

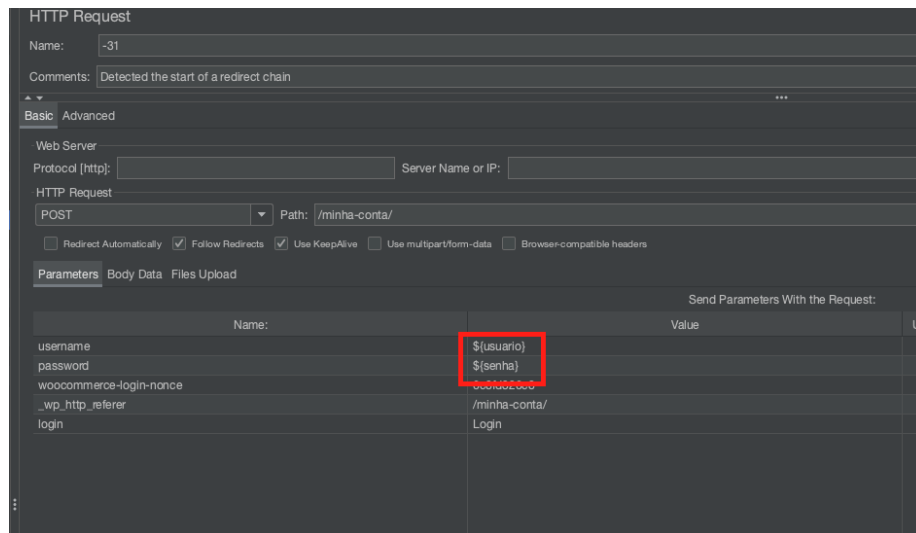
user3_ebac / psw!ebac@test

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:



5. CONCLUSÃO

Estou muito feliz de concluir o curso, após longos meses de esforço e realizando o curso no meu tempo, consegui incluir essa experiência no meu trabalho diário. Meu maior desafio foi entender que mesmo sabendo e tendo experiência com certos pontos abordados no curso, precisei diminuir meu ego e estar aberto a compreender que o conhecimento nunca é pouco, então me deixei guiar pelo aprendizado e então consegui aprender mais e mais a cada dia. Indico muito o curso aos mais novos profissionais que estão ingressando na área.

6. REFERÊNCIAS BIBLIOGRÁFICAS

Apache JMeter - Apache JMeter HTTP(S) Test Script Recorder. Disponível em: <https://jmeter.apache.org/usermanual/jmeter_proxy_step_by_step.html>. Acesso em: 02 out. 2023.

Assertions | Cypress Documentation. Disponível em: <<https://docs.cypress.io/guides/references/assertions>>. Acesso em: 02 out. 2023.