



ENGENHEIRO DE QUALIDADE DE SOFTWARE

Thiago Augusto Cavalcante da Silva

Análise de Qualidade

São Paulo

2025

1. RESUMO

O projeto de conclusão de curso em Engenharia de Qualidade de Software da EBAC - Escola Britânica de Artes Criativas e Tecnologia aborda a otimização dos processos de desenvolvimento de software com foco na qualidade.

Utilizando abordagens de engenharia de software, gerenciamento de projetos e técnicas de teste, o projeto visa aprimorar a eficiência e a confiabilidade das aplicações desenvolvidas.

A pesquisa examina a importância da definição de requisitos claros e do uso de metodologias ágeis, como o Scrum, para garantir uma comunicação eficaz entre as equipes de desenvolvimento. Além disso, o estudo se concentra na implementação de estratégias de automação de testes e na adoção de métricas de qualidade para avaliar o desempenho do software.

O objetivo final é demonstrar como a aplicação de práticas de engenharia de qualidade de software pode resultar em produtos mais robustos, reduzindo custos e tempo de desenvolvimento. O projeto destaca a importância crescente da qualidade de software no mercado atual, onde a satisfação do cliente e a competitividade são fundamentais para o sucesso das empresas de tecnologia.

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
4.3 Casos de testes	10
4.4 Repositório no Github.....	15
4.5 Testes automatizados	15
4.6 Integração contínua	16
4.7 Testes de performance.....	16
5. CONCLUSÃO	17
6. REFERÊNCIAS BIBLIOGRÁFICAS	17

3. INTRODUÇÃO

A qualidade de software tem um papel essencial em um mundo cada vez mais digitalizado. Com o avanço contínuo da tecnologia e sua presença crescente em todos os aspectos da vida cotidiana, a demanda por soluções de software confiáveis e de alta qualidade se torna cada vez mais evidente.

Utilizar um sistema ou aplicativo que apresenta falhas constantes, erros ou não corresponde às expectativas do usuário é extremamente frustrante. Além disso, esses problemas podem gerar prejuízos financeiros e afetar negativamente a imagem das empresas. É nesse cenário que se destaca a importância da Engenharia de Qualidade de Software (QA – Quality Assurance).

A QA é a área responsável por assegurar que o software seja desenvolvido, testado e entregue com elevados padrões de qualidade. Esse processo inclui a definição de requisitos bem estruturados, a aplicação de metodologias eficazes de desenvolvimento e a adoção de práticas rigorosas de teste. A automação de testes e a análise de métricas também são fundamentais na busca por excelência contínua.

Nesse contexto, a Engenharia de Qualidade de Software deixa de ser apenas um diferencial competitivo e passa a ser uma exigência indispensável para as organizações que desejam se destacar no ambiente tecnológico atual. Além de garantir a satisfação dos usuários, ela fortalece a confiabilidade, a segurança e a eficiência dos sistemas que utilizamos no dia a dia.

Dessa forma, investir em qualidade de software é essencial para impulsionar a inovação e garantir o sucesso dos negócios no cenário digital.

A qualidade de software é um pilar estratégico para o sucesso empresarial em um mundo digital, com impacto direto na experiência do cliente, na eficiência operacional e na redução de custos. O estudo trará práticas recomendadas e já adotadas por empresas, buscando fomentar inovações em Quality Assurance (QA) e contribuir para a evolução do mercado nacional, alinhado às melhores práticas globais.

4. O PROJETO

Para este trabalho de conclusão de curso **Profissão: Engenheiro de Qualidade de software**, você deve utilizar o conhecimento adquirido ao longo do curso para elaborar uma estratégia de testes adequada para validar o e-commerce EBAC Shop (<http://lojaebac.ebaconline.art.br/>). Você deve considerar as histórias de usuário já refinadas como se você estivesse participando de um time ágil. As funcionalidades devem seguir todo o fluxo de trabalho de um *Quality Engineer* (QE), desde o planejamento até a entrega. Siga as etapas dos sub-tópicos para se orientar no trabalho.

ATENÇÃO:

- Conforme a sua estratégia, você pode executar os testes no endereço disponibilizado ou utilizando as imagens disponíveis no Docker Hub:
 - Banco de Dados: [ernestosbarbosa/lojaebacdb](https://hub.docker.com/r/ernestosbarbosa/lojaebacdb)
 - Loja EBAC: [ernestosbarbosa/lojaebac](https://hub.docker.com/r/ernestosbarbosa/lojaebac)
 - Comandos para subir os containers:

```
docker network create --attachable ebac-network

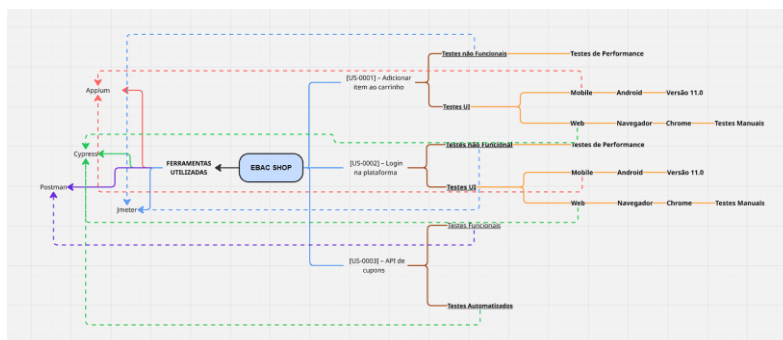
docker run -d --name wp_db -p 3306:3306 --network ebac-network ernestosbarbosa/lojaebacdb:latest

docker run -d --name wp -p 80:80 --network ebac-network ernestosbarbosa/lojaebac:latest
```

Após subir os containers a loja estará em <http://localhost:80>

- Como este trabalho complementa o que criou em seu Trabalho de Consolidação (Módulo 19), você pode utilizá-lo como base para o seu Trabalho de Conclusão.

4.1 Estratégia de teste



4.2 Critérios de aceitação

- Considere as histórias de usuário e para cada uma delas crie pelo menos 4 critérios de aceitação usando a linguagem Gherkin;
 - [US-0001] – Adicionar item ao carrinho

Funcionalidade: Adicionar item ao carrinho

Como cliente da EBAC-SHOP

Quero adicionar produtos no carrinho

Para realizar a compra dos itens

#Limite de quantidade por produto

Cenário: Adicionar mais de 10 unidades do mesmo produto ao carrinho

Dado que o cliente está na página de um produto

Quando ele tenta adicionar 11 unidades do mesmo produto ao carrinho

Então uma mensagem de erro deve ser exibida informando que o limite é de 10 unidades

E o produto não deve ser adicionado

#Valor total no carrinho não pode ultrapassar R\$ 990,00

Cenário: Adicionar produtos que ultrapassam R\$ 990,00 no carrinho

Dado que o cliente tem R\$ 950,00 em produtos no carrinho

Quando ele tenta adicionar um produto de R\$ 50,00

Então uma mensagem de erro deve ser exibida informando que o valor máximo é R\$ 990,00

E o produto não deve ser adicionado ao carrinho

#Cupom de 10% para valores entre R\$ 200 e R\$ 600

Cenário: Receber cupom de 10% ao atingir valor entre R\$ 200 e R\$ 600

Dado que o cliente adicionou produtos no carrinho totalizando R\$ 450,00

Quando ele visualizar o carrinho

Então ele deve receber automaticamente um cupom de 10% de desconto

#Cupom de 15% para valores acima de R\$ 600

Cenário: Receber cupom de 15% ao ultrapassar R\$ 600

Dado que o cliente adicionou produtos no carrinho totalizando R\$ 700,00

Quando ele visualizar o carrinho

Então ele deve receber automaticamente um cupom de 15% de desconto

- [US-0002] – Login na plataforma

Funcionalidade: Login na plataforma

Como cliente da EBAC-SHOP

Quero fazer o login (autenticação) na plataforma

Para visualizar meus pedidos

#Login com usuário ativo

Cenário: Login bem-sucedido com usuário ativo

Dado que o cliente possui um cadastro ativo na EBAC-SHOP

E está na página de login

Quando ele informa um e-mail e senha válidos

Então ele deve ser autenticado com sucesso

E deve ser redirecionado para a página de pedidos

#Erro ao informar login ou senha incorretos

Cenário: Login com e-mail ou senha incorretos

Dado que o cliente está na página de login

Quando ele informa um e-mail ou senha inválidos

Então uma mensagem de erro deve ser exibida informando que os dados estão incorretos

E o login não deve ser realizado

#Bloqueio após 3 tentativas inválidas

Cenário: Bloquear login após 3 tentativas inválidas

Dado que o cliente errou o login ou a senha 2 vezes

Quando ele tenta fazer o login pela 3ª vez com dados inválidos

Então uma mensagem deve informar que a conta foi temporariamente bloqueada

E ele não poderá tentar novamente por 15 minutos

#Impedir login de usuário inativo

Cenário: Usuário inativo tenta fazer login

Dado que o cliente possui um cadastro inativo na EBAC-SHOP

Quando ele informa e-mail e senha válidos

Então uma mensagem deve ser exibida informando que o usuário está inativo

E o login não deve ser realizado

- [US-0003] – API de cupons

Funcionalidade: API de cupons

Como admin da EBAC-SHOP

Quero criar um serviço de cupom

Para poder listar e cadastrar os cupons

#Autenticação via Basic Auth com sucesso

Cenário: Realizar autenticação com credenciais válidas

Dado que o admin possui as credenciais corretas

Quando ele realiza uma requisição para a API de cupons com o header Authorization contendo o token Basic válido

Então a API deve aceitar a requisição

E retornar o status 200 (OK)

#Listar todos os cupons cadastrados

Cenário: Listar todos os cupons cadastrados

Dado que o admin está autenticado

Quando ele realiza uma requisição GET para o endpoint de cupons

Então a API deve retornar uma lista com todos os cupons cadastrados

E o status da resposta deve ser 200 (OK)

#Cadastrar novo cupom com dados válidos

Cenário: Cadastrar um novo cupom com todos os campos obrigatórios

Dado que o admin está autenticado

Quando ele envia uma requisição POST com o código "Ganhe10", valor "10.00", tipo "fixed_product" e descrição "Cupom de teste"

Então o cupom deve ser criado com sucesso

E a API deve retornar o status 201 (Created)

#Erro ao cadastrar cupom com nome duplicado

Cenário: Tentar cadastrar um cupom com código já existente

Dado que o cupom com o código "Ganhe10" já está cadastrado

Quando o admin tenta cadastrar um novo cupom com o mesmo código

Então a API deve retornar uma mensagem de erro informando que o nome do cupom já existe

E o status da resposta deve ser 400 (Bad Request)

- Crie histórias de usuário para as funcionalidades:

- **Catálogo de Produtos**

ID: US-0004

Como cliente da EBAC-SHOP

Quero visualizar o catálogo de produtos disponíveis

Para poder navegar, comparar e escolher os produtos que desejo comprar

- **Painel Minha Conta**

ID: US-0005

Como cliente da EBAC-SHOP

Quero acessar o painel “Minha Conta” com minhas informações pessoais, pedidos e configurações

Para gerenciar minha conta de forma prática e centralizada

- **Meus Pedidos**

ID: US-0006

Como cliente da EBAC-SHOP

Quero visualizar todos os meus pedidos realizados

Para acompanhar o status de entrega, histórico de compras e detalhes de cada pedido

- **Endereços**

ID: US-0007

Como cliente da EBAC-SHOP

Quero poder editar as informações do meu endereço na página "Meu Endereço"

Para atualizar dados como nome, rua, número, complemento, CEP, cidade e estado

- **Detalhes da Conta**

ID: US-0008

Como cliente da EBAC-SHOP

Quero poder visualizar as informações da minha conta na página "Detalhes da Conta"

Para manter meus dados corretos e atualizados na plataforma

- Referência: Módulo 8

4.3 Casos de testes

- Crie pelo menos 4 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...”
- Identifique quais os casos de teste serão automatizados, sendo ao menos 1 caminho feliz e 1 caminho alternativo.
- Referência: Módulos 4 e 5

○ [US-0001] – Adicionar item ao carrinho

➤ RN01

Não é permitido inserir mais de 10 itens de um mesmo produto ao carrinho;

Dados Válidos	Dados Inválidos
0 até 10	>10
0,1,2,3,4,5,6,7,8,9,10	11,12,13,14...97,98,99,100

➤ Casos de Testes

ID	Entrada	Saída
TC_001	Validar inserir 5 itens de um mesmo produto ao carrinho	Válido
TC_002	Validar inserir 10 itens de um mesmo produto ao carrinho	Válido
TC_003	Validar inserir mais de 11 itens de um mesmo produto ao carrinho	Inválido
TC_004	Validar inserir 100 itens de um mesmo produto ao carrinho	Inválido

➤ RN02

Os valores não podem ultrapassar a R\$ 990,00

Dados Válidos	Dados Inválidos
R\$ 0 até R\$ 899,99	>R\$ 990,00
0,10,00 ,20,00 ,...,100,00,..., 899,99	990,00,1.000,000,...

➤ Casos de Testes

ID	Entrada	Saída
TC_001	Total do valor de R\$ 80,00	Válido
TC_002	Total do valor de R\$ 899,99	Válido
TC_003	Total do valor de R\$ 990,01	Inválido
TC_004	Total do valor de R\$ 1.000,00	Inválido

➤ RN03

Valores entre R\$ 200 e R\$ 600, ganham cupom de 10%

Dados Inválidos	Dados Válidos	Dados Inválidos
< R\$199,99	R\$ 200,00 até R\$ 600,00	> R\$600,00
10,00 ,20,00 ...,100,00,..., 199,00	200,00,300,00,..., 600,00	700,00,800,00,...

➤ Casos de Testes

ID	Entrada	Saída
TC_001	Total do valor de R\$ 599,99	Válido
TC_002	Total do valor de R\$ 200,00	Válido
TC_003	Total do valor de R\$ 150,00	Inválido
TC_004	Total do valor de R\$ 600,01	Inválido

➤ RN04

Valores acima de R\$ 600 ganham cupom de 15%

Dados Válidos	Dados Inválidos
> 990,00	0 até 599,99
100,00,500,000,..., 800,000	900,00, 901...1.000,000,...

➤ Casos de Testes

ID	Entrada	Saída
TC_001	Total do valor de R\$ 1.000,00	Válido
TC_002	Total do valor de R\$ 600,001	Válido
TC_003	Total do valor de R\$ 80,00	Inválido
TC_004	Total do valor de R\$ 599,99	Inválido

○ [US-0002] – Login na plataforma

➤ **RN01**

Somente usuários ativos podem fazer login;

Condições	User 1	User 2	User 3	User 4
Usuário inativo	Não	Sim	Não	Sim
Usuário Bloqueado	Não	Não	Sim	Sim
Ações				
Permitir acesso?	Sim	Não	Não	Não

➤ **Casos de Testes**

ID	Entrada	Saída
TC_001	Login com usuário Ativo e Usuário desbloqueado , deve permitir acesso	Válido
TC_002	Login com usuário Inativo e Usuário desbloqueado , não deve permitir acesso	Inválido
TC_003	Login com usuário Ativo e Usuário bloqueado , não deve permitir acesso	Inválido
TC_004	Login com usuário Inativo e Usuário bloqueado , não deve permitir acesso	Inválido

➤ **RN02**

Somente usuários ativos podem fazer login;

Condições	User 1	User 2	User 3	User 4
Usuário válido	Sim	Sim	Não	Não
Senha válida	Sim	Não	Sim	Não
Ações				
Permitir acesso?	Sim	Não	Não	Não

➤ **Casos de Testes**

ID	Entrada	Saída
TC_001	Login com usuário válido e senha válida , deve permitir acesso	Válido
TC_002	Login com usuário válido e senha inválida , não deve permitir acesso	Inválido
TC_003	Login com usuário inválido e senha válida , não deve permitir acesso	Inválido
TC_004	Login com usuário inválido e senha inválida , não deve permitir acesso	Inválido

➤ **RN03**

Caso o usuário erre a senha 3 vezes, deve bloquear o acesso por 15 minutos;

Condições	User 1	User 2
Bloqueio por tentativas erradas	<2	>3
Ações		
Bloquear	Não	Sim

➤ **Casos de Testes**

ID	Entrada	Saída
TC_001	Permitir acesso do usuário se na terceira tentativa de login ele inserir a senha correta	Válido
TC_002	Não permitir acesso do usuário se na quarta tentativa ele não inserir a senha correta	Inválido

➤ **RN04**

Apenas alunos entre 16 e 80 anos podem ser cadastrados;

Condições	Regra 1	Regra 2	Regra 3
Idade	<16	16 a 18	>80
Ações			
Cadastrar	Não	Sim	Não

➤ **Casos de Testes**

ID	Entrada	Saída
TC_001	Deve permitir cadastro de um usuário com a idade entre 16 e 80 anos	Válido
TC_002	Não deve permitir o cadastro de um usuário com a idade Menor que 16 anos	Inválido
TC_003	Não deve permitir o cadastro de um usuário com a idade Maior que 80 anos	Inválido

○ [US-0003] – API de cupons

➤ **RN01**

Apenas usuários autenticados podem realizar o cadastro de cupom

➤ **Casos de Testes**

Testes	Teste 1	Teste 2	Teste 3
Estado Inicial:	Login e senha corretos	Login correto e senha incorreta	Login e senha incorretos
Evento:	Integrar	Não integrar	Não integrar
Estado Final:	Cadastrar cupom	Não cadastrar cupom	Não cadastrar cupom

➤ **RN02**

Apenas requisições informando novos nomes podem cadastrar os cupons

➤ **Casos de Testes**

Testes	Teste 1	Teste 2
Estado Inicial:	Novo nome	Nome repetido
Evento:	Integrar	Não integrar
Estado Final:	Cadastrar cupom	Não cadastrar cupom

➤ **RN03**

Apenas novos nomes de cupons podem ser cadastrados

➤ **Casos de Testes**

Testes	Teste 1	Teste 2
Estado Inicial:	Novo nome	Nome repetido
Evento:	Integrar	Não integrar
Estado Final:	Cadastrar cupom	Não cadastrar cupom

➤ **RN04**

Deve cadastrar os cupons informando os campos obrigatórios

➤ **Casos de Testes**

Testes	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5
Estado Inicial:	Informar os campos: Código do cupom, Valor, Tipo do desconto e Descrição	Informar apenas os campos: Valor, Tipo do desconto e Descrição	Informar apenas os campos: Código do cupom, Tipo do desconto e Descrição	Informar apenas os campos: Código do cupom, Valor e Descrição	Informar apenas os campos: Código do cupom, Valor e Tipo do desconto
Evento:	Integrar	Não integrar	Não integrar	Não integrar	Não integrar
Estado Final:	Cadastrar cupom	Não cadastrar cupom	Não cadastrar cupom	Não cadastrar cupom	Não cadastrar cupom

4.4 Repositório no Github

- Crie um repositório no GitHub com o nome TCC-EBAC-QE;
- Deixe o repositório público até a análise dos tutores;
- Neste repositório você deve subir este arquivo e todos os código fontes das automações que criar.
- Referência: Módulo 10
- Link do repositório: <https://github.com/thiagoacavalcantes/TCC-EBAC-QE>

4.5 Testes automatizados

4.5.1 Automação de UI

Ao considerar ferramentas de automação de testes para nossos projetos, é crucial avaliar cuidadosamente as opções disponíveis, levando em consideração fatores como facilidade de uso, velocidade de execução, manutenção de código e integração contínua. Após uma análise aprofundada, decidimos adotar o Cypress como nossa ferramenta de automação preferida, superando o Selenium e o Playwright, pelas seguintes razões:

- **Facilidade de Uso e Produtividade Eficiente:**

O Cypress se destaca pela sua abordagem amigável e intuitiva para a automação de testes. Sua sintaxe simplificada e estrutura de teste clara permitirão que nossos desenvolvedores escrevam cenários de teste de maneira eficaz e rápida. Isso acelera nosso processo de desenvolvimento de testes e economiza tempo valioso.

- **Execução Rápida e Eficiente:**

A arquitetura de execução direta no navegador do Cypress oferece uma vantagem significativa em termos de velocidade de execução em comparação com o Selenium. Isso nos permitirá realizar testes de regressão e validação de maneira mais rápida, melhorando nossa eficiência geral no ciclo de desenvolvimento.

- **API Simplificada e Manutenção Acessível:**

A API coesa e direta do Cypress reduz a complexidade do nosso código de teste, resultando em testes mais legíveis e fáceis de manter. Ao evitar a verbosidade associada ao Selenium e ao Playwright, conseguiremos criar e atualizar cenários de teste com mais facilidade, reduzindo o esforço de manutenção ao longo do tempo.

- **Integração Contínua e Suporte Amplo:**

O Cypress possui integração sólida com várias ferramentas de integração contínua amplamente utilizadas, como Jenkins e Travis CI. Isso nos permitirá incorporar nossos testes de forma transparente em nosso pipeline de entrega contínua, garantindo que a qualidade seja mantida em cada etapa do processo de desenvolvimento.

- **Visibilidade de Testes e Depuração Simplificada:**

A interface de controle interativo do Cypress oferece visibilidade abrangente sobre a execução dos testes, facilitando a identificação e solução rápida de problemas. Isso reduzirá o tempo necessário para depurar e resolver quaisquer problemas que possam surgir durante a execução dos testes.

Baseados nessas considerações, a escolha do Cypress como nossa ferramenta de automação de testes é uma decisão estratégica que visa aumentar a qualidade de

nossos produtos, otimizar nossos processos de teste e melhorar nossa produtividade geral. Estamos confiantes de que o Cypress nos ajudará a atingir nossos objetivos de garantir uma experiência confiável e consistente para nossos usuários.

4.5.2 Automação de API

- Crie uma pasta chamada API para os testes de API dos casos de teste que forem automatizados
- Você deve utilizar a ferramenta Supertest para criar seus testes de API
- Não esqueça de validar os contratos! 😊

4.5.3 Automação Mobile

- Considere para os APPs apenas a funcionalidade de Catálogo de Produtos
 - Você pode encontrar os APPs em:
 - *Android*: <https://github.com/EBAC-QE/testes-mobile-ebac-shop/tree/main/app/android>
 - *iOS*: <https://github.com/EBAC-QE/testes-mobile-ebac-shop/tree/ios-tests/app/ios>
 - Crie uma pasta chamada Mobile para os testes em aplicativos dos casos de teste que forem automatizados
 - Utilize ao menos um *Testing Pattern* (à sua escolha) na implementação dos testes.
 - Você deve implementar testes para ao menos uma das plataformas Mobile (*Android* ou *iOS*)
- Observações:
 - Considere todas as boas práticas aprendidas até aqui
 - Não esqueça de implementar a geração de relatórios
 - Referência: Módulos 11, 12, 14, 16, 17, 22, 23, 24, 29 e 30

4.6 Integração contínua

- Execute os testes automatizados em integração contínua utilizando o Github Actions
- Referência: Módulo 26






4.7 Testes de performance

- Usando o K6, implemente um teste de performance em ao menos 2 casos de testes
- Referência: Módulo 28

- 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

Coloque sua experiência na realização do trabalho, o que aprendeu, quais lições pode aplicar em sua vida profissional etc.

6. REFERÊNCIAS BIBLIOGRÁFICAS

Seguir regras ABNT