

UNISOCIESC

Projeto Final de Disciplina: Ciclo Completo de Testes de Software

Grupo: Eduardo Paza Flores(RA:152312248),Gabriel Vieira (RA:152311662)

Guilherme Uliano (RA:152310797), Luan Galvani (RA:152220636)

Pedro Neves(RA:152420539), Thomas Labes (RA:152420965)

Introdução

Neste trabalho, elaboramos o ciclo completo de testes de software como componente do projeto final da matéria. Usamos o site www.demoblaze.com, Site de demonstração para simular interações com uma loja online.

O Cypress é usado para automatizar testes funcionais, enquanto o JMeter é usado para realizar testes de carga e desempenho. Desde o planejamento até a implementação e avaliação dos resultados, o projeto foi estruturado. A documentação e os testes de todo o trabalho será disponibilizada em um repositório no GitHub.

Planejamento e Especificação

Testar as principais funcionalidades do site, como navegação, cadastro, login e compra de produtos, também garantir a qualidade funcional e avaliar o desempenho do sistema em cenários de uso intenso.

Para garantir que tudo esteja bem organizado e que todas as funcionalidades importantes do sistema sejam verificadas. No nosso projeto, vamos testar a aplicação Demoblaze, validando não apenas as operações principais, mas também como o sistema se comporta em situações de maior uso.

Especificação de Casos de Teste

Caso de teste: 1	Objetivo: Realizar cadastro de novo usuário
Passos: 1- Acessar o site Demoblaze 2- Clicar no botão Sign up 3- Preencher os campos necessários 4- Clicar em Sign up	Entradas: Nome de usuário: teste0112 Senha: senha0112
Resultado esperado: Cadastrar o novo usuário com sucesso.	

Caso de teste: 2	Objetivo: Fazer login com usuário e senha cadastrados.
Passos: 1- Acessar o site Demoblaze 2- Clicar no botão Log in 3- Inserir o nome de usuário e senha válido 4- Clicar em Log in	Entradas: Nome de usuário: teste0112 Senha: senha0112
Resultado esperado: Conectar a conta do usuário e aparecer "Welcome teste0112" no menu superior.	

Caso de teste: 3	Objetivo: Adicionar produtos ao carrinho de compras
Passos: 1- Acessar o site Demoblaze 2- Navegar até a página de um item 3- Clicar em "Add to cart" 4- Clicar em Cart	Entradas: Produto: Samsung galaxy s6
Resultado esperado: Quando entrar na página do carrinho haverá o produto adicionado.	

Caso de teste: 4	Objetivo: Realizar uma compra válida
Passos: 1- Acessar o site Demoblaze 2- Navegar até a página do produto 3- Inserir o nome de usuário e senha válido 4- Ir para o carrinho 5- Preencher os campos necessários 6- Clicar em Purchase	Entradas: Nome de usuário: teste0112 Senha: senha0112 Produtos: Nexus 6 Name: João Paulo Country: Brasil City: Blumenau Credit Card: 1230 Month: 12 Year: 2030
Resultado esperado: Adicionar o produto no carrinho, ir para finalizar a compra, preencher os campos necessários e enviar, após enviar receber a mensagem de sucesso "Thank you for your Purchase" e clicar em Ok.	

Caso de teste: 5	Objetivo: Finalizar uma compra inválida
Passos: 1- Acessar o site Demoblaze 2- Ir direto para o carrinho 3- Clicar em finalizar compra 4- Preencher os dados corretos 5- Clicar em Purchase	Entradas: Produtos: Nexus 6 Name: João Paulo Country: Brasil City: Blumenau Credit Card: 1230 Month: 12 Year: 2030
Resultado esperado: Não ser possível finalizar a compra, porque não tem usuário conectado e nem itens no carrinho de compra.	

Caso de teste: 6	Objetivo: Cadastro de usuário já existente
Passos: 1- Acessar o site Demoblaze 2- Clicar em "Sign up". 3- Inserir nome de usuário e senha existente. 4- Clicar em "Sign up".	Entradas: Nome de usuário: teste0112 Senha: senha0112
Resultado esperado: Aparecer um alerta de erro com a mensagem de usuário já cadastrado.	

Caso de teste: 7	Objetivo: Fazer logout do usuário conectado
Passos: 1- Acessar o site Demoblaze 2- Fazer o log in 3- Clicar em Log out	Entradas: Nome de usuário: teste0112 Senha: senha0112
Resultado esperado: Entrar na conta do usuário e depois conseguir desconectar sem problemas.	

Caso de teste: 8	Objetivo: Excluir item do carrinho de compras
Passos: 1- Acessar o site Demoblaze 2- Adicionar o produto no carrinho 3- Ir para o carrinho 4- Clicar em Delete	Entradas: Produto: Iphone 6 32gb
Resultado esperado: O produto do carrinho seja excluído e limpando o carrinho de compras.	

Caso de teste: 9	Objetivo: Adicionar o mesmo produto ao carrinho
Passos: 1- Acessar o site Demoblaze 2- Adicionar o produto no carrinho 3- Adicionar novamente o produto no carrinho 4- Ir para carrinho	Entradas: Produtos: Samsung galaxy s6
Resultado esperado: Ter dois produtos iguais no carrinho de compras, já que o site não trabalha com quantidade.	

Caso de teste: 10	Objetivo: Testar link de contato
Passos: 1- Acessar o site Demoblaze 2- Clicar em Contact 3- Preencher os campos 4- Clicar em Send Message	Entradas: Contact Email: joaopaulo@teste.com Contact Name: João Paulo Message: Teste de e-mail
Resultado esperado: Abrir o link de contato, preencher os campos e depois enviar mensagem sem dar erros.	

Caso de teste: 1	Objetivo: Simular 1.000 usuários navegando ao mesmo tempo pelo site, acessando páginas principais.
Passos: 1- Simular acesso a página inicial	Entradas: Number of Threads (users): 1.000 Ramp-Up Period: 1 segundos Loop Count: 1
Resultado esperado: Todas as requisições devem retornar status positivo. O servidor deve suportar 1.000 usuários sem falhas. Porcentagem de erros: 0%	

Caso de teste: 2	Objetivo: Simular 1.000 usuários navegando ao mesmo tempo pelo site, acessando páginas principais.
Passos: 1- Simular acesso a página inicial	Entradas: Number of Threads (users): 1.000 Ramp-Up Period: 500 milissegundos Loop Count: 1
Resultado esperado: Todas as requisições devem retornar status positivo. O servidor deve suportar 1.000 usuários sem falhas. Porcentagem de erros: 100%	

Caso de teste: 3	Objetivo: Simular 10.000 usuários navegando ao mesmo tempo pelo site, acessando o carrinho.
Passos: 1- Simular acesso a página do carrinho	Entradas: Number of Threads (users): 10.000 Ramp-Up Period: 10 segundos Loop Count: 3
Resultado esperado: Todas as requisições devem retornar status positivo. O servidor deve suportar 10.000 usuários sem falhas. Porcentagem de erros: 28,74%	

Caso de teste: 4	Objetivo: Simular 20.000 usuários navegando ao mesmo tempo pelo site, acessando o carrinho.
Passos: 1- Simular acesso a página do carrinho	Entradas: Number of Threads (users): 20.000 Ramp-Up Period: 30 segundos Loop Count: 1
Resultado esperado: Todas as requisições devem retornar status positivo. O servidor deve suportar 20.000 usuários sem falhas. Porcentagem de erros: 36,85%	

Tipos de testes

1. Funcionais (usando **Cypress**):

- Automação de cenários de CRUD.
- Validação dos retornos (códigos de status e dados).
- Testes de interface gráfica, caso aplicável.
- Testar funcionalidades principais, como login, navegação e compras

2. Não funcionais (usando **JMeter**):

- Testes de carga com múltiplas requisições simultâneas nos endpoints mais críticos.
 - Análise de tempos de resposta, desempenho sob carga e limites do site.
-

Critérios para os testes serem realizados

● Antes de começar:

- O site demoblaze precisa estar disponível e funcional.
- Ferramentas configuradas: **Cypress** e **JMeter**.

● Quando terminamos:

- Todos os cenários de teste precisam ser executados.
 - Resultados organizados e analisados em um relatório final.
-

Ferramentas utilizadas

- **Cypress**: Para testes funcionais automatizados.
 - **JMeter**: Para realizar testes de carga e performance.
 - **GitHub**: Repositório para armazenar códigos, relatórios e documentação.
-

Ambiente de testes

- **Demoblaze**: <https://www.demoblaze.com/>
 - **Configuração local**:
 - Ferramentas instaladas: **Cypress** e **JMeter**.
 - Ambiente de execução: Máquina com internet estável, JavaScript e Java configurados.
-

Cenários para Automação com Cypress:

1. Realizar cadastro.
2. Realizar login/logout
3. Adicionar produtos ao carrinho.
4. Remover produtos do carrinho.
5. Finalizar uma compra.
6. Finalizar uma compra inválida.
7. Enviar mensagem no Contato
8. Adicionar dois itens iguais ao carrinho.

Testes Não Funcionais com JMeter:

1. Simular múltiplos usuários navegando simultaneamente pelo site
2. Testar a carga no carrinho.

Resultados e Análises

Testes Funcionais

Total de Casos Executados: 8

Passaram: 7

Falharam: 1

Análise: O caso de erro foi identificado na compra de um produto sem precisar de uma conta conectada e com o carrinho de compras vazio.

Testes Não Funcionais

Gráficos e Estatísticas:

Usuários: 1000

Tempo de resposta: Tempo médio de resposta por número de usuários é de 6237

Análise:

Erros: 0,00%

Usuários: 1000

Tempo de resposta: Tempo médio de resposta por número de usuários é de 7018

Análise:

Erros: 100%

Usuários: 10000

Tempo de resposta: Tempo médio de resposta por número de usuários é de 327223

Análise:

Erros: 28,74%

Usuários: 20000

Tempo de resposta: Tempo médio de resposta por número de usuários é de 59821

Análise:

Erros: 36,85%

Conclusão

Os testes realizados permitiram validar a funcionalidade e o desempenho do site selecionado dentro dos cenários propostos

O site atendeu aos requisitos principais para usuários regulares, mas apresentou problemas de desempenho em cenários de alta carga.

De modo geral, o sistema demonstrou estabilidade e confiabilidade em operações essenciais, como navegação, login e finalização de compras.

O desempenho do site é aceitável para até 1000 usuários simultâneos. Acima disso, o tempo de resposta ultrapassa 3 segundos, indicando necessidade de otimização.

A automação dos testes funcionais demonstrou ser eficiente na validação de cenários comuns, porém teve um erro em um cenário crítico que seria na finalização de uma compra sem usuário.

Recomendações

- Melhorar o sistema de validação do carrinho de compras.
- Garantir que o usuário esteja conectado ao finalizar uma compra ou dar a opção para criar conta após ir para o carrinho de compras.
- Adicionar quantidade de produtos no carrinho, para não precisar adicionar o mesmo produto várias vezes
- Criar um botão para excluir todos itens do carrinho de uma vez.
- Melhorar a comunicação de erros com o usuário, substituindo os alerts
- Otimizar o site para suportar mais de 1000 usuários simultâneos sem que de erros de conexão.
- Realizar uma melhoria no servidor de host para que suporte mais usuários simultaneamente.

Repositório no GitHub:

https://github.com/guilhermeuliano/A3_Gestao_e_qualidade_de_software