

**FACULDADE SENAC GOIÁS
GESTÃO DA TECNOLOGIA DA INFORMAÇÃO.**

Lúcio de Souza Torres
Ordóñez Ribeiro
Vinicius Abadio
Jordy Alecssander

Auditoria e Qualidade de Software

GOIÂNIA/2019

Histórico de Revisões

| Data | Versão | Descrição | Autor |
|-------------|---------------|--|--|
| 13/11/2019 | 1.0 | Documento de Relatório de testes de Loja virtual | Lúcio Torres Ordônez Ribeiro Vinicius Abadio JordyAlecssander |
| | | | |

Sumário

| | | |
|--------|--|----|
| 1. | Introdução | 4 |
| 1.1 | Visão Geral do Documento | 4 |
| 2. | Escopo | 4 |
| 3. | Testes Realizados | 4 |
| 3.1 | Testes Funcionais | 4 |
| 3.1.1 | Objetivo dos Testes | 4 |
| 3.1.2 | Setup dos Experimentos realizados | 4 |
| 3.1.3 | Método Utilizado | 4 |
| 3.1.4 | Técnica | 5 |
| 3.1.5 | CT01 – Realizar Login | 5 |
| 3.1.6 | CT02 – Senha Inválida no Login | 5 |
| 3.1.7 | CT03 – Realizar Cadastro | 6 |
| 3.1.8 | CT04 – Cadastro de Produto | 6 |
| 3.1.9 | CT05 – Exclusão de produto do carrinho | 7 |
| 3.1.10 | CT06 – Pagamento com cartão de crédito | 7 |
| 3.2 | Testes de Respostas da API | 8 |
| 3.2.1 | @GetMapping | 8 |
| 3.2.2 | @GetMapping("/{id}") | 9 |
| 3.2.3 | @DeleteMapping("/{id}") | 10 |
| 3.3 | Testes de Performance | 10 |
| 3.4 | Testes de Configuração | 10 |
| 4. | Referências | 11 |

Relatório de Testes

1. Introdução

Este documento define o projeto de testes do sistema de Loja Virtual, a serem realizados durante e após a implementação do mesmo, visando validar as funcionalidades oferecidas pelo sistema. Neste documento será avaliado se os requisitos do sistema estão sendo satisfeitos, e serão descritos procedimentos para a execução dos testes, métodos utilizados e avaliação dos resultados obtidos.

Um caso de teste especifica uma maneira de testar o sistema: o que testar, quais os valores e pré-condições de entrada e os valores e pós-condições de saída. A realização de um ou mais procedimentos de teste é formado por um conjunto de instruções para execução, que pode ser feito com auxílio de ferramentas ou manualmente.

1.1 Visão Geral do Documento

Seção 1 – Introdução: Esta parte refere-se a visão geral do documento.

Seção 2 – Escopo: Nesta parte se encontra os testes que foram realizados.

Seção 3 – Testes Realizados: Descreve os objetivos dos testes realizados, as condições de realização dos experimentos, a metodologia utilizada.

Seção 4 – Referências: Lista dos documentos utilizados para a elaboração deste documento.

2. Escopo

Neste documento serão descritos os testes que foram realizados na Loja Virtual.

- **Testes Funcionais**
- **Testes de Resposta**
- **Testes de Performance**
- **Testes de Configuração**

3. Testes Realizados

3.1 Testes Funcionais

3.1.1 *Objetivo dos Testes*

Assegurar o correto funcionamento dos recursos oferecidos pelo software, incluindo a navegação, processamento dos dados de entrada, fluxo dos casos de uso e resultados.

3.1.2 *Setup dos Experimentos realizados*

O site deve ser aberto no navegador;

3.1.3 *Método Utilizado*

Os testes foram feitos verificando-se o fluxo dos dados de entrada para cada um deles, verificando-se a resposta do programa, se foi adequada às entradas, reportando erros se os dados eram inválidos ou gerando saídas corretas, a partir de dados válidos.

3.1.4 Técnica

Executar cada caso de uso, o fluxo de caso de uso ou função, utilizando dados válidos ou inválidos, para verificar o seguinte:

- O resultado esperado ocorre quando entradas válidas são utilizadas.
- Erros apropriados ocorrem quando entradas inválidas são utilizadas.
- Cada regra de negócio é aplicada corretamente.

3.1.5 CT01 – Realizar Login

| [CT01] | |
|----------------------|--|
| Caso de teste | CT 01 – Realizar Login |
| Pré-condições | O usuário deverá ter iniciado o navegador, estar com a página web da loja aberta e estar devidamente cadastrado. |
| Procedimento | <ol style="list-style-type: none">1) O ator deverá clicar no botão de Login2) O ator deverá informar o seu usuário3) O ator deverá informar sua senha4) O sistema verifica os dados |
| Resultado esperado | Login com êxito e direcionamento para página inicial |
| Dados de entrada | Um login válido |
| Prioridade | Alta |

3.1.6 CT02 – Senha Inválida no Login

| [CT02] | |
|----------------------|--|
| Caso de teste | CT 02 – Senha Inválida |
| Pré-condições | Estar na tela de login. |
| Procedimento | <ol style="list-style-type: none">1) O ator informa uma senha inválida e preenche um login válido2) O ator seleciona a opção OK3) O sistema verifica se os campos obrigatórios foram preenchidos4) O sistema verifica se o login do usuário está cadastrado no sistema e se a senha é correta5) O sistema exibe a mensagem “Login/Senha inválidos” |
| Resultado esperado | Mensagem de erro do sistema |
| Dados de entrada | Um login inválido |
| Prioridade | Alta |

3.1.7 CT03 – Realizar Cadastro

[CT03]

| | |
|----------------------|---|
| Caso de teste | CT 03 – Realizar cadastro |
| Pré-condições | Estar na tela de cadastro |
| Procedimento | <ol style="list-style-type: none">1) O ator deverá informar um Nome e Sobrenome2) O ator deverá informar um e-mail para vínculo3) O ator deverá escolher um nome de usuário4) O ator deverá preencher informações de pagamento5) O ator deverá criar uma senha6) O sistema validará se os campos obrigatórios foram preenchidos7) Um e-mail será enviado para confirmação do cadastro |
| Resultado esperado | Mensagem de aprovação do sistema e solicitação para confirmar cadastro no e-mail |
| Dados de entrada | Informações pessoais e bancárias |
| Prioridade | Alta |

3.1.8 CT04 – Cadastro de Produto

[CT04]

| | |
|----------------------|---|
| Caso de teste | CT 04 – Cadastro de Produto |
| Pré-condições | Estar logado como administrador e na tela de produtos |
| Procedimento | <ol style="list-style-type: none">1) O ato deverá informar o nome do produto2) O ator deverá informar o preço3) O ator deverá informar uma breve descrição do produto4) O ator deverá informar quantidade em estoque |
| Resultado esperado | Mensagem de confirmação do cadastro de produto |
| Dados de entrada | Informações do produto como nome e preço |
| Prioridade | Alta |

3.1.9 CT05 – Exclusão de produto do carrinho

[CT05]

| | |
|----------------------|--|
| Caso de teste | CT 05 – Exclusão de produto do carrinho |
| Pré-condições | Possuir produto no carrinho de compras |
| Procedimento | <ol style="list-style-type: none">1) O ator deverá clicar no carrinho de compras2) O ator deverá escolher os produtos a serem excluídos3) O ator deverá clicar em retirar do carrinho4) A lista de produtos será atualizada e o valor total do carrinho recalculado |
| Resultado esperado | Produto excluído da lista de compras do carrinho |
| Dados de entrada | Produtos selecionados na loja para o carrinho |
| Prioridade | Média |

3.1.10 CT06 – Pagamento com cartão de crédito

[CT06]

| | |
|----------------------|---|
| Caso de teste | CT 06 – Pagamento com cartão de crédito |
| Pré-condições | Ter produtos no carrinho e estar na tela do carrinho de compras |
| Procedimento | <ol style="list-style-type: none">1) O ator deverá clicar em concluir compra2) O ator será direcionado para a tela de pagamento3) O ator deverá escolher pagamento com cartão de crédito4) O ator deverá informar os dados do cartão5) O ator deverá clicar em continuar e informar o código de segurança do cartão |
| Resultado esperado | Mensagem de compra aprovada |
| Dados de entrada | Informações do cartão |
| Prioridade | Alta |

3.2 Teste de Resposta da API

*Os testes foram feitos através de uma integração com o Swagger
(Framework para API's REST)*

3.2.1 @GetMapping

Nossa primeira requisição http irá buscar retornar uma lista de Produtos.

1. Primeiro faremos uma requisição sabendo que nosso banco de dados está vazio.
Feita uma requisição via GET em uma tabela sem dados:

The image shows the Swagger UI interface for a resource named 'produto-resource'. The endpoint is a GET request to '/produto list'. The parameters section is empty. The 'Execute' button has been clicked, and the response is displayed. The response status is 200, and the response body is an empty array []. The response headers include 'access-control-allow-methods: POST, GET, OPTIONS, PUT, DELETE, HEAD', 'access-control-allow-origins: *', and 'access-control-max-age: 1728000'.

| Code | Details |
|------|---|
| 200 | <p>Response body</p> <pre>[]</pre> <p>Response headers</p> <pre>access-control-allow-methods: POST, GET, OPTIONS, PUT, DELETE, HEAD access-control-allow-origins: * access-control-max-age: 1728000</pre> |

Podemos observar que o servidor nos devolve o código 200 OK, ou seja, a requisição foi realizada com sucesso; porém no Response body, ou seja, corpo da requisição/resposta ele [], ou seja não existem dados cadastrados na tabela Produto.

2. Agora faremos uma requisição com dados em nosso banco.

The screenshot shows the Swagger UI for a resource named 'produto-resource'. The endpoint is 'GET /produto' with a description 'list'. There is an 'Authorization' header field marked as required. The 'Execute' button is highlighted in blue. Below the 'Responses' section, the 'Response content type' is set to '*/*'. The 'Curl' section shows the command: `curl -X GET "http://localhost:8080/produto" -H "accept: */*" -H "Authorization: "`. The 'Request URL' is `http://localhost:8080/produto`. The 'Server response' section shows a status code of 200. The 'Response body' is a JSON array with one object:

```
{
  "id": 1,
  "nome": "Razer Naga",
  "preco": 249.99,
  "descricao": "Mouse Gamer",
  "qtdeEstoque": 71,
  "und": "pc",
  "grupo": {
    "id": 2,
    "nome": "Informatica"
  }
}
```

Na imagem acima vemos que desta vez a requisição veio com um retorno de um json. Observamos também que seu status foi um 200 OK, como deveria ser, devido a ter feito um retorno de requisição http get com sucesso.

3.2.2 @GetMapping("/{id}")

Esta requisição http irá retornar um Produto passando como parâmetro seu 'id'.

The screenshot shows the Swagger UI for the same resource, but for a specific ID. The endpoint is 'GET /produto/1'. The 'Request URL' is `http://localhost:8080/produto/1`. The 'Server response' section shows a status code of 200. The 'Response body' is a JSON object:

```
{
  "id": 1,
  "nome": "Teclado",
  "preco": 199,
  "descricao": "Gamer",
  "qtdeEstoque": 71,
  "und": "un",
  "grupo": {
    "id": 2,
    "nome": "Informatica"
  }
}
```

 There is a 'Download' button next to the response body. The 'Response headers' section shows: `access-control-allow-methods: POST, GET, OPTIONS, PUT, DELETE, HEAD`, `access-control-allow-origins: *`, `access-control-max-age: 1728000`, `connection: keep-alive`, `content-type: application/json;charset=UTF-8`, `date: Tue, 10 Dec 2019 13:53:36 GMT`, and `transfer-encoding: chunked`. At the bottom, the 'Responses' table shows a 200 status with the description 'OK'.

Ele executou a requisição e retornou um status 200 OK, ou seja, foi listado com sucesso o produto cujo 'id': 3, retornando assim um json com esse registro.

3.2.3 @DeleteMapping("/{id}")

Nesta requisição http irá deletar um Produto passando como parâmetro seu 'id'.

The screenshot shows a REST client interface with the following details:

- Method:** DELETE
- URL:** /produto/{id} delete
- Parameters:**
 - Name:** Authorization (required, string, header)
 - Description:** (empty text box)
 - Name:** id (required, integer(5int32), path)
 - Description:** id
 - Value:** 2
- Buttons:** Execute (blue), Clear
- Responses:**
 - Response content type:** */*
 - Curl:** curl -X DELETE "http://localhost:8080/produto/2" -H "accept: */*" -H "Authorization: "
 - Request URL:** http://localhost:8080/produto/2
 - Server response:**
 - Code:** 200
 - Details:**
 - Response headers:**

```
access-control-allow-methods: POST, GET, OPTIONS, PUT, DELETE, HEAD
access-control-allow-origins: *
access-control-max-age: 1728000
connection: keep-alive
content-length: 0
date: Tue, 10 Dec 2019 13:15:00 GMT
```

Ele executou a requisição e retornou um status 204, ou seja, foi deletado com sucesso o registro cujo id era 2.

3.3 Testes de Performance

Nos nossos projetos não foi preciso fazer testes de performance, pois o nosso software não exige muito processamento.

3.4 Testes de Configuração

No nosso projeto não foi preciso fazer testes de configuração, pois um deles é Web e o outro funciona localmente.

4. Referências

Esta parte faz uma relação dos documentos que foram utilizados como referência por este.

- Slide Aula 5 – Teste de Software, Professora Ana Flávia
- Modelo de Relatório de Testes, Professora Ana Flávia
- Standard for Software Test Documentation:
<https://www.devmedia.com.br/padrao-para-documentacao-de-teste-de-software/26534>