

# Plano de testes

## API Simulação de Crédito

---

# ÍNDICE

<b>1. INTRODUÇÃO</b>	<b>3</b>
1.1 Propósito do sistema	3
<b>2. ABORDAGEM DE TESTES</b>	<b>4</b>
2.1 Categorização dos requisitos funcionais	4
2.2 Detalhamento da abordagem de teste	5
<b>3. AMBIENTE DE TESTES</b>	<b>8</b>
3.1 Definição do ambiente de teste	8
3.2 Possíveis riscos	8
<b>4. PROGRAMAÇÃO DE TESTES</b>	<b>9</b>
4.1 Objetivos e prioridades de testes	9
<b>5. CASOS DE TESTES</b>	<b>9</b>
5.1 Tabelas de testes	9
5.2 Cobertura de testes	13

---

# 1. INTRODUÇÃO

## 1.1 Propósito do sistema

A API Book Store é uma API REST para armazenar e buscar livros, clientes e pedidos em um banco de dados.

---

## 2. ABORDAGEM DE TESTES

### 2.1 Categorização dos requisitos funcionais

Requisitos Funcionais
<b>RF001</b> - O sistema deve realizar o cadastro do usuário
<b>RF002</b> - O sistema deve retornar um token
<b>RF003</b> - O sistema deve efetuar o login desde que o usuário obtenha um token para verificar a autorização do usuário
<b>RF004</b> - O sistema deve consultar usuário pelo ID
<b>RF005</b> - O sistema deve adicionar livros ao carrinho do usuário
<b>RF006</b> - O sistema deve modificar o livro que está no carrinho do usuário
<b>RF007</b> - O sistema deve mostrar os livros que estão no carrinho
<b>RF008</b> - O sistema deve realizar a busca do livro no carrinho pelo seu ISBN
<b>RF009</b> - O sistema deve excluir um livro específico do carrinho
<b>RF010</b> - O sistema deve excluir todos os livros do carrinho
<b>RF011</b> - O sistema deve excluir o usuário

---

## 2.2 Detalhamento da abordagem de teste

<b>Tipo de teste:</b>	Funcional
<b>Objetivo do teste:</b>	Testar se o sistema cadastra o usuário corretamente
<b>Requisito que motivou o teste:</b>	<b>RF001</b> - O sistema deve realizar o cadastro do usuário

<b>Tipo de teste:</b>	Funcional
<b>Objetivo do teste:</b>	Testar se o sistema retorna um token
<b>Requisito que motivou o teste:</b>	<b>RF002</b> - O sistema deve retornar um token

<b>Tipo de teste:</b>	Funcional
<b>Objetivo do teste:</b>	Testar se o usuário consegue verificar sua autorização
<b>Requisito que motivou o teste:</b>	<b>RF003</b> - O sistema deve efetuar o login desde que o usuário obtenha um token para verificar a autorização do usuário

<b>Tipo de teste:</b>	Funcional
<b>Objetivo do teste:</b>	Testar se o sistema consegue efetuar a consulta do usuário pelo ID
<b>Requisito que motivou o teste:</b>	<b>RF004</b> - O sistema deve consultar usuário pelo ID

<b>Tipo de teste:</b>	Funcional
<b>Objetivo do teste:</b>	Testar se o sistema adiciona livros no carrinho
<b>Requisito que motivou o teste:</b>	<b>RF005</b> - O sistema deve permitir adicionar livros ao carrinho

<b>Tipo de teste:</b>	Funcional
<b>Objetivo do teste:</b>	Testar se o sistema faz a modificação dos livros no carrinho
<b>Requisito que motivou o teste:</b>	<b>RF006</b> - O sistema deve modificar o livro que está no carrinho do usuário

<b>Tipo de teste:</b>	Funcional
<b>Objetivo do teste:</b>	Testar se o sistema mostra os livros que estão no carrinho
<b>Requisito que motivou o teste:</b>	<b>RF007</b> - O sistema deve mostrar os livros que estão no carrinho

<b>Tipo de teste:</b>	Funcional
<b>Objetivo do teste:</b>	Testar se o sistema realiza a busca do livro que está no carrinho
<b>Requisito que motivou o teste:</b>	<b>RF008</b> - O sistema deve realizar a busca do livro pelo seu ISBN

<b>Tipo de teste:</b>	Funcional
<b>Objetivo do teste:</b>	Testar se o sistema exclui um livro específico do carrinho
<b>Requisito que motivou o teste:</b>	<b>RF009</b> - O sistema deve excluir um livro específico do carrinho

<b>Tipo de teste:</b>	Funcional
<b>Objetivo do teste:</b>	Testar se o sistema exclui todos os livros do carrinho
<b>Requisito que motivou o teste:</b>	<b>RF010</b> - O sistema deve excluir todos os livros do carrinho

<b>Tipo de teste:</b>	Funcional
<b>Objetivo do teste:</b>	Testar se o sistema exclui o usuário
<b>Requisito que motivou o teste:</b>	<b>RF011</b> - O sistema deve excluir o usuário

---

## 3. AMBIENTE DE TESTES

### 3.1 Definição do ambiente de teste

Para executar os testes automatizados os seguintes recursos devem ser considerados:

- Energia elétrica
- Computador que suporte rodar a API Book Store e testes os softwares utilizados para a automação de testes são eles: JUnit, java, Rest Assured, allure e maven.
- Internet estável e com boa conexão

### 3.2 Possíveis riscos

- Energia elétrica instável

Possível solução: Obter um nobreak no local da aplicação

- Internet instável

Possível solução: Obter um modem 4G/5G

- Computador com problemas

Possível solução: Obter um notebook ou computador reserva



---

## 4.PROGRAMAÇÃO DE TESTES

### 4.1 Objetivos e prioridades de testes

As prioridades dos testes são:

- Possibilitar a o cadastro do usuário
- O sistema deve conseguir efetuar o login através de nome e senha para verificar autorização
- O sistema deve conseguir consultar o usuário cadastrado
- O sistema deve retornar um token
- O sistema deve conseguir deletar usuários cadastrados
- O sistema deve conseguir consultar os livros que estão no carrinho
- O sistema deve conseguir consultar os dados de um livro específico
- O sistema deve adicionar livros ao carrinho
- O sistema deve editar os livros do carrinho
- O sistema deve excluir um livro do carrinho
- O sistema deve excluir todos os livros do carrinho

## 5. CASOS DE TESTES

### 5.1 Tabelas de testes

Endpoint : Account

Verbo/Definição	ID	Passos	Resultado esperado
POST- Cadastrar usuário	01	Criar um usuário	Sistema retorna status 201
	02	Criar um usuário com senha inválida	Sistema retorna status code 406
	03	Criar um usuário sem senha	Sistema retorna status code 406
	04	Criar um usuário sem nome	Sistema retorna status code 406

Plano de testes

---

<b>Verbo/Definição</b>	<b>ID</b>	<b>Passos</b>	<b>Resultado esperado</b>
POST- Gerar token	05	Gerar token para usuário	Retornar status code 200
	06	Gerar token com campo senha em branco	Retornar status code 400
	07	Gerar token com campo nome em branco	Retornar status code 400
	08	Gerar token com todos os campos em branco	Retornar status code 400
	09	Gerar token para usuário que não existe	Retornar status code 400
	10	Gerar token com nome incorreto	Retornar status code 404
	11	Gerar token com senha incorreta	Retornar status code 404

<b>Verbo/Definição</b>	<b>ID</b>	<b>Passos</b>	<b>Resultado esperado</b>
POST- Autorização	12	Fazer login para verificar autorização	Retornar status code 200
	13	Fazer login com campo senha em branco para verificar autorização	Retornar status code 400
	14	Fazer login com campo nome em branco para verificar autorização	Retornar status code 400
	15	Fazer login com todos os campos em branco para verificar autorização	Retornar status code 400
	16	Verificar a autorização com	Retornar status code 404

		usuário que não existe	
	17	Verificar a autorização com usuário sem token	Retornar status code 404
	18	Verificar autorização com nome de usuário incorreto	Retornar status code 404
	19	Verificar autorização com senha de usuário incorreta	Retornar status code 404

Verbo/Definição	ID	Passos	Resultado esperado
GET/{ID}- Consultar usuário	20	Consultar usuário	Retornar status code 200
	21	Consultar usuário sem autorização	Retornar status code 401
	22	Consultar usuário com token errado	Retornar status code 401

Verbo/Definição	ID	Passos	Resultado esperado
DELETE/{Usuario} }- Deletar usuário	23	Deletar usuário	Retornar status code 200
	24	Deletar usuário com token incorreto	Retornar status code 204
	25	Deletar usuário sem token	Retornar status code 204
	26	Deletar usuário com ID incorreto	Retornar status code 404

Endpoint: BookStore

Verbo/Definição	ID	Passos	Resultado esperado
GET- Ver todos os livros	27	Ver todos os livros	Retornar status code 200

Verbo/Definição	ID	Passos	Resultado esperado
GET{ISBN}- Ver todos os livros do carrinho	28	Procurar por um livro pelo ISBN	Retornar status code 200
	29	Procurar um livro com ISBN incorreto	Retornar status code 400

Verbo/Definição	ID	Passos	Resultado esperado
POST- Adicionar livros no carrinho	30	Adicionar livros no carrinho	Retornar status code 201
	31	Adicionar livros no carrinho de um usuário sem token	Retornar status code 401
	32	Adicionar um livro que não existe no carrinho	Retornar status code 400
	33	Adicionar livro no carrinho com id incorreto	Retornar status code 401
	34	Adiciona dois livros iguais no carrinho separadamente	Retornar status code 400
	35	Adiciona dois livros iguais no carrinho ao mesmo tempo	Retornar status code 400

Verbo/Definição	ID	Passos	Resultado esperado
DELETE - Deletar todos os livros	36	Deletar todos os livros do carrinho	Retornar status code 204
	37	Deletar todos os livros do carrinho sem token	Retornar status code 401

	38	Deletar todos os livros do carrinho com id incorreto	Retornar status code 401
--	----	--	--------------------------

Verbo/Definição	ID	Passos	Resultado esperado
DELETE{ISBN}- Deletar um livro	39	Deletar um livro do carrinho	Retornar status code 204
	40	Deletar um livro do carrinho sem token	Retornar status code 401
	41	Deletar um livro que não está no carrinho	Retornar status code 400

Verbo/Definição	ID	Passos	Resultado esperado
PUT{ISBN}- Atualizar carrinho	42	Modificar o livro que está no carrinho	Retornar status code 200
	43	Modificar o livro que está no carrinho com um livro que não existe	Retornar status code 400
	44	Modificar um livro que está no carrinho sem token	Retornar status code 401

## 5.2 Cobertura de testes

- **Path Coverage(endpoints):** 100% dos endpoints cobertos
- **Operator Coverage(todos os métodos existentes):** 11 métodos = 100% dos métodos cobertos
- **Parameter Coverage(parâmetros existentes):** 11 parâmetros = 100% dos parâmetros cobertos

