

Documentação do Projeto - Sistema de Cadastro de Produtos

Integrantes: Gabriel Carlos Rezende Nazario, José Eliomax Pereira Mariano, João Marcos Da Silva Braz, Luana Alexandre da Silva e Rodrigo Vieira Fagundes

• Descrição Geral

O Sistema de Cadastro de Produtos é uma aplicação desenvolvida em Java com Spring Boot e MySQL. Seu objetivo é gerenciar categorias de produtos, produtos e usuários com controle de acesso, permitindo o cadastro, listagem, atualização e exclusão de registros.

O sistema foi projetado com autenticação HTTP Basic e autorização baseada em perfis (COMUM e ADMIN), garantindo segurança e controle de acesso às operações.

• Dependências

- Spring Data JPA
 - Spring Boot Web
 - H2
 - MySQL Connector
 - DevTools
 - Lombok
 - Spring Boot Test
-

• Arquitetura

```
com.cadastro.produtos/
  └── configs/
      └── SecurityConfigurations.java → Configuração de segurança
          (autenticação/autorização)
      └── AuthService.java → Serviço de autenticação (carrega usuário)
  └── controller/ → Camada de apresentação (API REST)
      └── ProdutoController.java → Endpoints de produtos
      └── CategoriaController.java → Endpoints de categorias
```

```
|   └── UsuarioController.java → Endpoints de usuários  
|  
|   ├── entity/ → Entidades do banco  
|   |   ├── Produto.java → Entidade Produto  
|   |   ├── Categoria.java → Entidade Categoria  
|   |   ├── Usuario.java → Entidade Usuario (implementa UserDetails)  
|   |   └── Perfil.java → Entidade Perfil (implementa GrantedAuthority)  
|  
|   ├── repository/ → Camada de persistência (acesso ao banco)  
|   |   ├── ProdutoRepository.java → Interface JPA para Produto  
|   |   ├── CategoriaRepository.java → Interface JPA para Categoria  
|   |   ├── UsuarioRepository.java → Interface JPA para Usuario  
|   |   └── PerfilRepository.java → Interface JPA para Perfil  
|  
|   ├── service/ → Camada de negócio (lógica da aplicação)  
|   |   ├── ProdutoService.java → Regras de negócio para Produto  
|   |   ├── CategoriaService.java → Regras de negócio para Categoria  
|   |   └── UsuarioService.java → Regras de negócio para Usuario  
|  
|   └── exception/ → Camada de tratamento de exceções  
|       ├── ResourceNotFoundException.java → Exceção para recurso não encontrado  
|       ├── ValidationException.java → Exceção para validações  
|       └── GlobalExceptionHandler.java → Tratamento global de exceções  
  
└── Cadastroprodutos5Application.java → Classe principal do Spring Boot
```

- **Funcionalidades**

- **Categoria**
 - Cadastrar nova categoria (Requer ADMIN)
 - Listar todas as categorias (COMUM e ADMIN)
 - Buscar categoria por ID (COMUM e ADMIN)
 - Atualizar categoria existente (Requer ADMIN)
 - Excluir categoria (Requer ADMIN)

- **Produto**

- Cadastrar novo produto (associando a uma categoria existente) **(Requer ADMIN)**
- Listar todos os produtos **(COMUM e ADMIN)**
- Buscar produto por ID **(COMUM e ADMIN)**
- Atualizar produto existente **(Requer ADMIN)**
- Excluir produto **(Requer ADMIN)**

- **Usuário**

- Cadastrar novo usuário com senha criptografada **(Requer ADMIN)**
- Listar todos os usuários **(Requer ADMIN)**
- Buscar usuário por ID **(Requer ADMIN)**
- Buscar dados do próprio usuário **(COMUM e ADMIN)**
- Atualizar usuário existente **(Requer ADMIN)**
- Excluir usuário **(Requer ADMIN)**

- **Rotas Principais (Endpoints)**

- **Categoria**

Método	Endpoint	Descrição	Permissão
GET	/categorias/	Retorna categorias	COMUM ADMIN
GET	/categorias/{id}	Retorna categoria específica	COMUM ADMIN
POST	/categorias	Cadastra categoria	ADMIN
PUT	/categorias/{id}	Atualiza categoria	ADMIN
DELETE	/categorias/{id}	Exclui categoria	ADMIN

- **Produto**

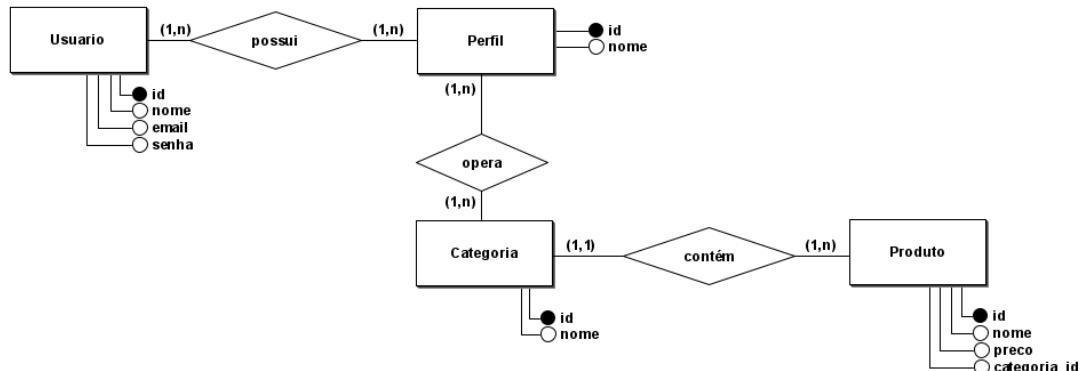
Método	Endpoint	Descrição	Permissão
GET	/produtos/	Retorna produtos	COMUM ADMIN
GET	/produtos/{id}	Retorna produto específico	COMUM ADMIN
POST	/produtos	Cadastra produto	ADMIN
PUT	/produtos/{id}	Atualiza produto	ADMIN
DELETE	/produtos/{id}	Exclui produto	ADMIN

- **Usuário**

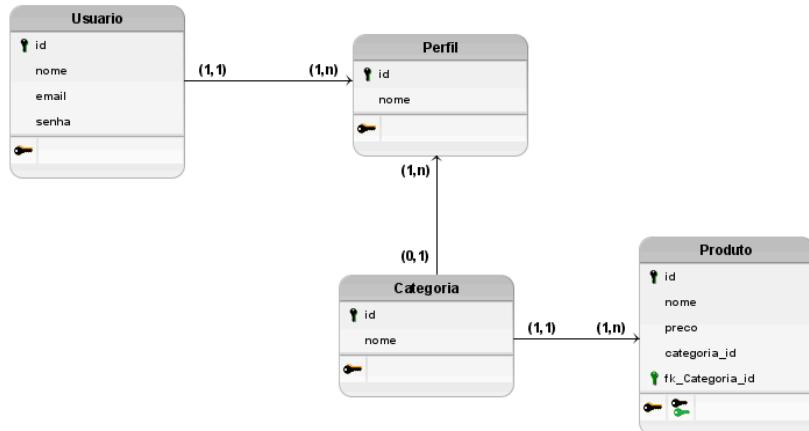
Método	Endpoint	Descrição	Permissão
GET	/usuarios/me	Retorna próprio usuário	COMUM ADMIN
GET	/usuarios	Retorna usuários	ADMIN
GET	/usuarios/{id}	Retorna usuário específico	ADMIN
POST	/usuarios/{id}	Cadastra usuário	ADMIN
PUT	/produtos/{id}	Atualiza usuário	ADMIN
DELETE	/produtos/{id}	Exclui usuário	ADMIN

- **Descrição Geral**

- **Modelo Entidade-Relacionamento (MER)**



- Diagrama Entidade-Relacionamento (DER)



- **Cardinalidades e Relacionamentos**

- Relacionamento 1: Usuario > Perfil
 - Usuário pode ter **vários perfis**.
 - Perfil pode estar associado a **vários usuários**.
 - Implementado através da tabela intermediária **usuário perfil**.
- Relacionamento 1: Categoria > Produto
 - Categoria pode ter vários produtos.
 - Um produto pertence a uma única categoria.
 - Implementado através da chave estrangeira categoria_id na tabela produto.

- **Estrutura de Dados**

- **Tabela: Usuario**

Campo	Tipo	Chave	Descrição
id	INTEGER	PRIMÁRIA	Identificador único do usuário
nome	VARCHAR	-	Nome completo do usuário
email	VARCHAR	ÚNICA	Email do usuário
senha	VARCHAR	-	Senha

- **Tabela: Perfil**

Campo	Tipo	Chave	Descrição
id	INTEGER	PRIMÁRIA	Identificador único do perfil
nome	VARCHAR	-	Nome do perfil

■ **Tabela: Usuario Perfil (Tabela de Junção)**

Campo	Tipo	Chave	Descrição
usuario_id	INTEGER	ESTRANGEIRA	Referencia usuario.id
perfil_id	INTEGER	ESTRANGEIRA	Referencia perfil.id

■ **Tabela: Categorias**

Campo	Tipo	Chave	Descrição
id	INTEGER	PRIMÁRIA	Identificador único da categoria
nome	VARCHAR	-	Nome da categoria

■ **Tabela: Produtos**

Campo	Tipo	Chave	Descrição
id	INTEGER	PRIMÁRIA	Identificador único do produto
nome	VARCHAR	-	Nome do produto
preco	DOUBLE	-	Preço do produto
categoria_id	INTEGER	ESTRANGEIRA	Referencia categorias.id

● **Considerações finais**

Este projeto exemplifica um CRUD completo e seguro, com:

- **Modelagem simples e eficiente**
- **Relacionamentos entre entidades**
- **Sistema de autenticação e autorização robusto**
- **Controle de acesso granular por perfis**
- **Persistência via Spring Data JPA**
- **Estrutura organizada e de fácil manutenção**
- **Segurança com criptografia de senhas**
- **Separação entre subsistema de autenticação e negócio**

O sistema está pronto para uso e pode ser expandido com novas funcionalidades, validações avançadas, paginação e filtros.

