

# Sistema de Cadastro de Produtos

Documentação do Projeto (Java / Spring Boot / MySQL)

# A Equipe

Gabriel Carlos Rezende Nazario

---

Rodrigo Vieira Fagundes

---

João Marcos Da Silva Braz

---




Luana Alexandre da Silva

---







José Eliomax Pereira Mariano

# Visão Geral e **Arquitetura**

## Descrição Geral

-  Aplicação em Java/Spring Boot para gerenciar produtos, categorias e usuários.
-  Objetivo: CRUD completo com regras de negócio.
-  Foco na separação de entidades, camadas e responsabilidades.

## Arquitetura em Camadas

-  **controller** → Trata as requisições HTTP (API REST).
-  **service** → Contém as regras de negócio (lógica).
-  **repository** → Comunicação com o banco de dados.
-  **entity** → Classes de modelo (entidades).
-  **exception** → Tratamento de erros personalizados.
-  **configs** → Configuração de segurança.

# Tecnologias Principais



**Java (21)** (Linguagem de Backend)



**Spring Boot** (Framework)



**Spring Data JPA** (Persistência)



**MySQL** (Banco de Dados de Produção)



**H2 Database** (Banco de Testes)

# Tecnologias Dependências

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

# Principais **Funcionalidades** (CRUD)

## **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 (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)

# Principais **Funcionalidades** (CRUD)

## **Usuário**

Cadastrar novo usuário com senha criptografada (**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**)

# Modelo **Relacional** (DER)

## Relacionamentos do Modelo



**Usuario (1,n)** possui **(1,n) Perfil**



**Perfil (1,n)** opera **(1,n) Categoria**

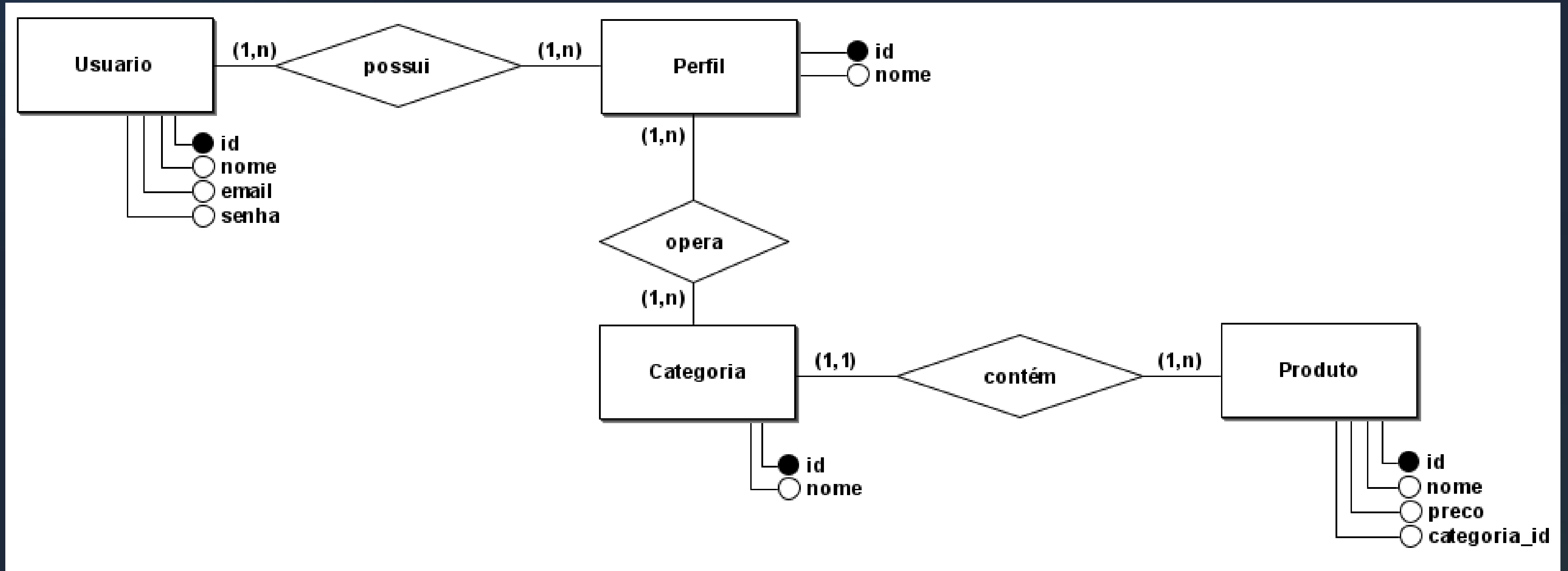


**Categoria (1,1)** contém **(1,n) Produto**

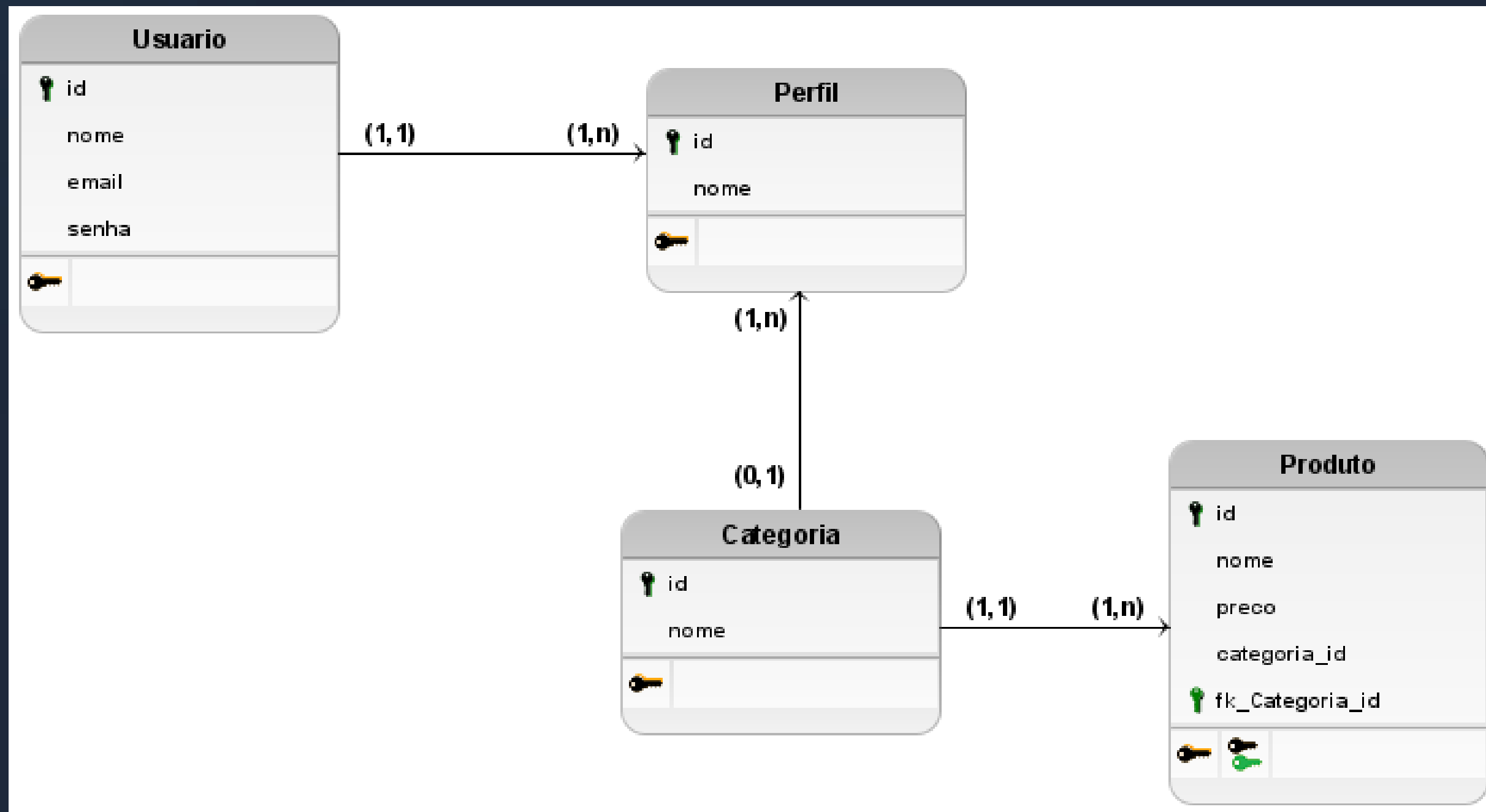
O modelo é composto pelas entidades Usuario, Perfil, Categoria e Produto. Usuario e Perfil usam um relacionamento N:N (muitos para muitos), enquanto Categoria e Produto usam um relacionamento 1:N (um para muitos).



# Modelo Conceitual



# Modelo Lógico



# Entidade: **Usuario**

Campo	Tipo	Chave	Descrição
id	INTEGER	PK	Identificador único do usuário
nome	VARCHAR	-	Nome do usuário
email	VARCHAR	-	Email de login (único)
senha	VARCHAR	-	Senha criptografada

# Entidade: **Perfil**

Campo	Tipo	Chave	Descrição
id	INTEGER	PK	Identificador único do perfil
nome	VARCHAR	-	Nome do perfil (ADMIN ou COMUM)

# Entidade: **Categoria**

Campo	Tipo	Chave	Descrição
id	INTEGER	PK	Identificador único da categoria
nome	VARCHAR	-	Nome da categoria (ex: Eletrônicos, Roupas)

# Entidade: **Produto**

Campo	Tipo	Chave	Descrição
id	INTEGER	PK	Identificador único do produto
nome	VARCHAR	-	Nome do produto (ex: Celular, Camisa)
preco	DECIMAL	-	Preço do produto
categoria_id	INTEGER	FK → Categoria (id)	Identifica a categoria a que o produto pertence

# Segurança e **Autorização** (Spring Security)

## Framework e Entidades



Utiliza o Spring Security para autenticação e autorização.



A segurança é baseada nas entidades Usuario e Perfil.



A senha do Usuario deve ser criptografada (ex: com BCrypt).

## Regras de Acesso (Perfis)



**Perfil COMUM** Acesso de leitura: pode listar e buscar produtos e categorias (GET).



**Perfil ADMIN** Acesso total (CRUD): criar, atualizar e excluir produtos/categorias (POST, PUT, DELETE), além do acesso GET.

# Rotas da API **Categoria**

Método	Endpoint	Descrição	Requer Perfil
GET	/categorias	Lista todas as categorias	COMUM ou ADMIN
GET	/categorias/{id}	Retorna uma categoria específica	COMUM ou ADMIN
POST	/categorias	Cadastra uma nova categoria	ADMIN
PUT	/categorias/{id}	Atualiza uma categoria existente	ADMIN
DELETE	/categorias/{id}	Exclui uma categoria	ADMIN



# Rotas da API Produto

Método	Endpoint	Descrição	Requer Perfil
GET	/produtos	Lista todos os produtos	COMUM ou ADMIN
GET	/produtos/{id}	Retorna um produto específico	COMUM ou ADMIN
POST	/produtos	Cadastra um novo produto	ADMIN
PUT	/produtos/{id}	Atualiza um produto existente	ADMIN
DELETE	/produtos/{id}	Exclui um produto	ADMIN

# Rotas da API **Usuario**

Método	Endpoint	Descrição	Requer Perfil
GET	/usuarios/me	Retorna proprio usuario	COMUM ou ADMIN
GET	/usuarios	Retorna usuarios	ADMIN
GET	/usuarios/{id}	retorna usuario especifico	ADMIN
POST	/usuarios/{id}	Cadastra usuario	ADMIN
PUT	/usuarios/{id}	Atualiza usuario	ADMIN
Delete	/usuarios/{id}	Exclui um usuario	ADMIN

---

# Testes da API

Demonstração dos Endpoints no Insomnia / Isonomia

# Perguntas?

Obrigado.