

Agro Família Pesca

Sistema de Gerenciamento da Agricultura Familiar e Pesca Artesanal

1. Introdução

A agricultura familiar e a pesca artesanal representam setores fundamentais para a segurança alimentar e o desenvolvimento socioeconômico local. Entretanto, a gestão das informações relacionadas a produtores, associações, programas governamentais e movimentações produtivas ainda enfrenta desafios decorrentes do uso de processos manuais ou sistemas fragmentados.

Neste contexto, o projeto **Agro Família Pesca** foi desenvolvido com o objetivo de criar uma **API RESTful robusta, segura e modular**, capaz de centralizar e organizar dados relacionados à agricultura familiar e pesca artesanal, atendendo às necessidades administrativas de Secretarias Municipais e Associações.

O sistema foi desenvolvido como parte de um **curso de especialização em Back-end com Node.js**, promovido pelo IFMA, servindo tanto como solução prática quanto como instrumento de aplicação de conceitos modernos de engenharia de software, arquitetura de APIs e segurança da informação.

2. Objetivos

2.1 Objetivo Geral

Desenvolver uma API REST para gerenciamento de dados da agricultura familiar e pesca artesanal, com controle de acesso por nível de usuário e escopo institucional.

2.2 Objetivos Específicos

- Implementar autenticação segura baseada em JWT.
- Controlar o acesso aos dados por níveis hierárquicos (Admin, Secretaria, Associação e Usuário).
- Aplicar arquitetura modular com separação clara de responsabilidades.
- Utilizar boas práticas de segurança e organização de código.
- Disponibilizar uma API reutilizável e extensível.

3. Visão Geral do Sistema

O projeto é composto por três partes principais:

- **API_agrofamilia_pesca**: API principal, responsável pelas regras de negócio e segurança.
- **API_Simplificada**: Versão reduzida da API, utilizada apenas para demonstração de consumo.
- **consumo**: Interface web estática para consumo da API simplificada.

O foco deste documento é a **API_agrofamilia_pesca**, que representa a implementação completa do sistema.

4. Arquitetura do Sistema

4.1 Estilo Arquitetural

O sistema adota uma **Arquitetura Monolítica Modular**, combinada com o padrão **Layered Architecture**, promovendo alta coesão, baixo acoplamento e facilidade de manutenção.

4.2 Camadas da Arquitetura

4.2.1 Camada de Entrada (Router & Controller)

Responsável por receber requisições HTTP, validar parâmetros básicos e encaminhar os dados para a camada de serviço.

- **Localização**: modules/*/*.controller.js
- **Responsabilidade**:
 - Controle de fluxo
 - Padronização das respostas HTTP
 - Tratamento inicial de erros

4.2.2 Camada de Serviço (Service Layer)

É o núcleo da aplicação, onde residem as regras de negócio, validações e controle de escopo.

- **Localização**: modules/*/*.service.js
- **Responsabilidade**:
 - Aplicar regras de negócio
 - Validar permissões
 - Orquestrar chamadas ao repositório

4.2.3 Camada de Persistência (Repository Layer)

Abstrai o acesso ao banco de dados, encapsulando consultas SQL e operações CRUD.

- **Localização:** modules/*/*.repository.js
- **Responsabilidade:**
 - Isolar o SQL da lógica de negócio
 - Utilizar tabelas e views conforme o contexto

4.2.4 Camada Compartilhada (Shared)

Contém recursos reutilizáveis por todo o sistema.

- **Exemplos:**
 - BaseService
 - Errors
 - validationsUtils
 - LevelPolicy

5. Controle de Acesso e Segurança

5.1 Autenticação

O sistema utiliza **JWT (JSON Web Tokens)** para autenticação stateless.

- Tokens com validade de **7 dias**
- Payload inclui informações essenciais para autorização
- Senhas armazenadas com **bcrypt (salt rounds = 10)**

5.1.1 Tratamento de Senhas Legadas

Durante o desenvolvimento do sistema **Agro Família Pesca**, foi identificada a existência de usuários com senhas armazenadas em formato **não criptografado**, oriundas de inserções manuais no banco de dados ou de versões iniciais do sistema, **anteriores à implementação completa dos mecanismos de segurança**.

Visando **manter a continuidade operacional** do sistema sem comprometer os princípios de segurança da informação, adotou-se uma estratégia de **migração automática de senhas** legadas. Nessa abordagem, o sistema identifica, no momento do login, se a senha armazenada no banco de dados encontra-se **criptografada no padrão bcrypt**.

Caso a senha esteja armazenada em **texto simples**, o sistema realiza a autenticação comparando diretamente o valor informado pelo usuário com o valor persistido. Após a autenticação bem-sucedida, **a senha é imediatamente convertida para o formato criptografado** utilizando o algoritmo bcrypt e atualizada no banco de dados.

Quando a senha já se encontra criptografada, o processo de autenticação **ocorre normalmente**. Dessa forma, garante-se **compatibilidade** com dados legados ao mesmo tempo em que **se eleva gradualmente o nível de segurança** do sistema.

Essa solução demonstra um equilíbrio entre usabilidade, segurança e manutenção de sistemas legados, sendo especialmente adequada para contextos institucionais e governamentais, nos quais a migração imediata de todas as credenciais pode não ser viável.

5.2 Autorização (RBAC)

O modelo de autorização é baseado em **RBAC (Role-Based Access Control)** combinado com **escopo contextual**.

Níveis de Acesso

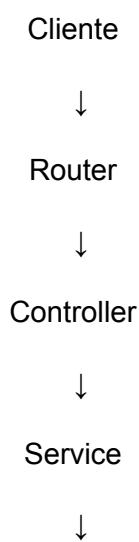
Nível	Perfil
1	Administrador
2	Secretaria
3	Associação
4	Usuário

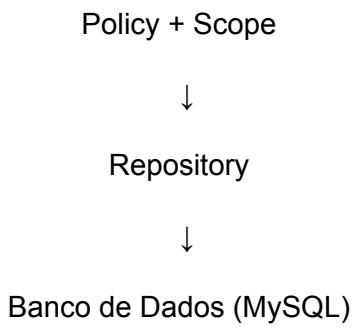
5.3 Escopo de Dados

O sistema utiliza o padrão **Scope**, implementado no BaseService.applyScope, garantindo que cada usuário visualize apenas dados compatíveis com sua Secretaria ou Associação.

6. Diagrama Conceitual (Descrição)

6.1 Diagrama de Arquitetura (Descrição Textual)





6.2 Diagrama de Entidades (Resumo)

- Pessoa
- Usuário
- Secretaria
- Associação
- Associado
- Programa
- Produto
- Movimentação

7. Documentação do Banco de Dados

7.1 Visão Geral do Banco de Dados

O sistema **Agro Família Pesca** utiliza um banco de dados relacional baseado em **MySQL**, projetado para garantir integridade referencial, consistência dos dados e eficiência nas consultas. A modelagem foi pensada para atender múltiplos perfis de acesso, com uso extensivo de **views** para isolamento de dados sensíveis e aplicação de escopo institucional.

O banco centraliza informações relacionadas a usuários, pessoas físicas, secretarias, associações, programas governamentais, produtos e movimentações produtivas.

7.2 Estratégia de Modelagem

A modelagem do banco segue os princípios da **normalização**, reduzindo redundâncias e garantindo integridade lógica:

- **1^a Forma Normal (1FN)**: Todos os atributos são atômicos.
- **2^a Forma Normal (2FN)**: Não há dependência parcial de chave.

- **3^a Forma Normal (3FN)**: Não há dependências transitivas.

As entidades foram separadas entre:

- **Entidades administrativas** (Usuário, Secretaria, Associação)
- **Entidades civis** (Pessoa)
- **Entidades operacionais** (Associado, Programa, Produto, Movimentação)

7.3 Entidades Principais

7.3.1 Pessoa

Representa o indivíduo físico, sendo a entidade base para usuários e associados.

Campos principais:

- ID (PK)
- NOME
- CPF
- DATA_NASCIMENTO
- SEXO
- ENDERECO

Observação:

A separação entre Pessoa e Usuário permite que uma pessoa exista no sistema sem, necessariamente, possuir credenciais de acesso.

7.3.2 Usuário

Representa a entidade responsável pela autenticação e autorização no sistema.

Campos principais:

- ID (PK)
- ID_PESSOA (FK → Pessoa)
- ID_SECRETARIA (FK → Secretaria)
- LOGIN
- SENHA
- NIVEL

Regras:

- A senha é armazenada de forma criptografada.
- O nível define o escopo de acesso do usuário.

7.3.3 Secretaria

Representa um órgão governamental responsável por associações e programas.

Campos principais:

- ID (PK)
- NOME
- CIDADE
- ESTADO
- ENDERECO

7.3.4 Associação

Entidade que agrupa produtores rurais ou pescadores.

Campos principais:

- ID (PK)
- NOME
- ENDERECO
- ID_SECRETARIA (FK → Secretaria)
- ID_CATEGORIA

7.3.5 Associado

Representa o vínculo formal entre uma Pessoa e uma Associação.

Campos principais:

- ID (PK)
- ID_PESSOA (FK → Pessoa)
- ID_ASSOCIACAO (FK → Associação)
- CAF
- VALIDADE_CAF

Regras de Negócio:

- O CAF deve ser único.
- Um associado pertence a apenas uma associação por vínculo ativo.

7.3.6 Programa

Representa programas governamentais de apoio à agricultura familiar e pesca.

Campos principais:

- ID (PK)
- NOME
- DESCRICAO
- ID_SECRETARIA

7.3.7 Produto

Representa produtos agrícolas ou pesqueiros cadastrados no sistema.

Campos principais:

- ID (PK)
- NOME
- UNIDADE_MEDIDA

7.3.8 Movimentação

Registra a produção ou movimentação de um associado.

Campos principais:

- ID (PK)
- ID_ASSOCIADO
- ID_PRODUTO
- QUANTIDADE
- DATA_MOVIMENTACAO

7.4 Relacionamentos Entre Entidades

Os principais relacionamentos do banco de dados são:

- **Pessoa 1:N Usuário**
- **Pessoa 1:N Associado**
- **Secretaria 1:N Associação**
- **Associação 1:N Associado**
- **Associado 1:N Movimentação**
- **Produto 1:N Movimentação**

Esses relacionamentos são implementados através de **chaves estrangeiras**, garantindo integridade referencial.

7.5 Uso de Views

O sistema utiliza **views SQL** como camada de abstração para consultas.

Objetivos do uso de Views:

- Melhorar a legibilidade das consultas
- Centralizar joins complexos
- Facilitar aplicação de escopo por nível de acesso
- Evitar exposição direta das tabelas base

Exemplos de Views:

- view_usuarios
- view_associados
- view_pessoas
- view_movimentacoes

As views são utilizadas exclusivamente para operações de **leitura**, enquanto operações de escrita são realizadas diretamente nas tabelas.

7.6 Integridade e Consistência dos Dados

O banco de dados implementa diversas estratégias para garantir a consistência:

- **Chaves primárias e estrangeiras**
- **Restrições de unicidade** (ex.: CAF, LOGIN)
- **Validação de existência de registros** antes de operações de escrita
- **Validação lógica na camada de serviço**, evitando inconsistências transacionais

7.7 Considerações sobre Escalabilidade e Manutenção

A separação entre tabelas e views, aliada ao uso do Knex.js como camada de acesso, facilita:

- Evolução do esquema sem impacto direto na API
- Refatoração de consultas complexas
- Portabilidade para outros bancos relacionais

7.8 Síntese da Camada de Persistência

A camada de banco de dados do sistema **Agro Família Pesca** foi projetada para suportar crescimento, segurança e clareza semântica, alinhando-se às boas práticas de engenharia de software e modelagem relacional.

7.9 Vinculação entre Banco de Dados e Módulos da API

O banco de dados do sistema **Agro Família Pesca** está diretamente acoplado à arquitetura modular da API, onde cada **módulo funcional** corresponde a um conjunto bem definido de **tabelas e views**, respeitando o princípio de **responsabilidade única**.

Essa vinculação garante:

- Clareza na manutenção
- Facilidade de rastreabilidade entre regra de negócio e persistência
- Segurança no controle de acesso por escopo

7.9.1 Mapeamento Módulo ↔ Entidades

7.9.1.1 Módulo Usuários

Responsável pela autenticação, autorização e controle de acesso.

Tabelas:

- usuario
- pessoa
- secretaria

Views:

- view_usuarios

Relacionamentos-chave:

- usuario.ID_PESSOA → pessoa.ID
- usuario.ID_SECRETARIA → secretaria.ID

7.9.1.2 Módulo Pessoas

Entidade base do sistema, utilizada por usuários e associados.

Tabela:

- pessoa

Relacionamentos:

- Pessoa ↔ Usuário
- Pessoa ↔ Associado

7.9.1.3 Módulo Secretarias

Representa órgãos governamentais responsáveis pela gestão institucional.

Tabela:

- secretaria

Relacionamentos:

- Secretaria ↔ Usuários
- Secretaria ↔ Associações
- Secretaria ↔ Programas

7.9.1.4 Módulo Associações

Agrupamento institucional de produtores e pescadores.

Tabelas:

- associacao
- associado

Views:

- view_associados

Relacionamentos:

- Associação ↔ Associados
- Associação ↔ Secretaria

7.9.1.5 Módulo Produtos / Tipos de Produto

Gerencia os produtos agrícolas e pesqueiros.

Tabelas:

- produto
- tipo_produto

Relacionamentos:

- Tipo de Produto ↔ Produto
- Produto ↔ Movimentação

7.9.1.6 Módulo Movimentações

Registra a produção dos associados.

Tabela:

- movimentacao

Views:

- view_movimentacoes

Relacionamentos:

- Associado ↔ Movimentação
- Produto ↔ Movimentação

7.9.1.7 Módulo Programas / Agricultura Familiar

Registra beneficiários de políticas públicas.

Tabelas:

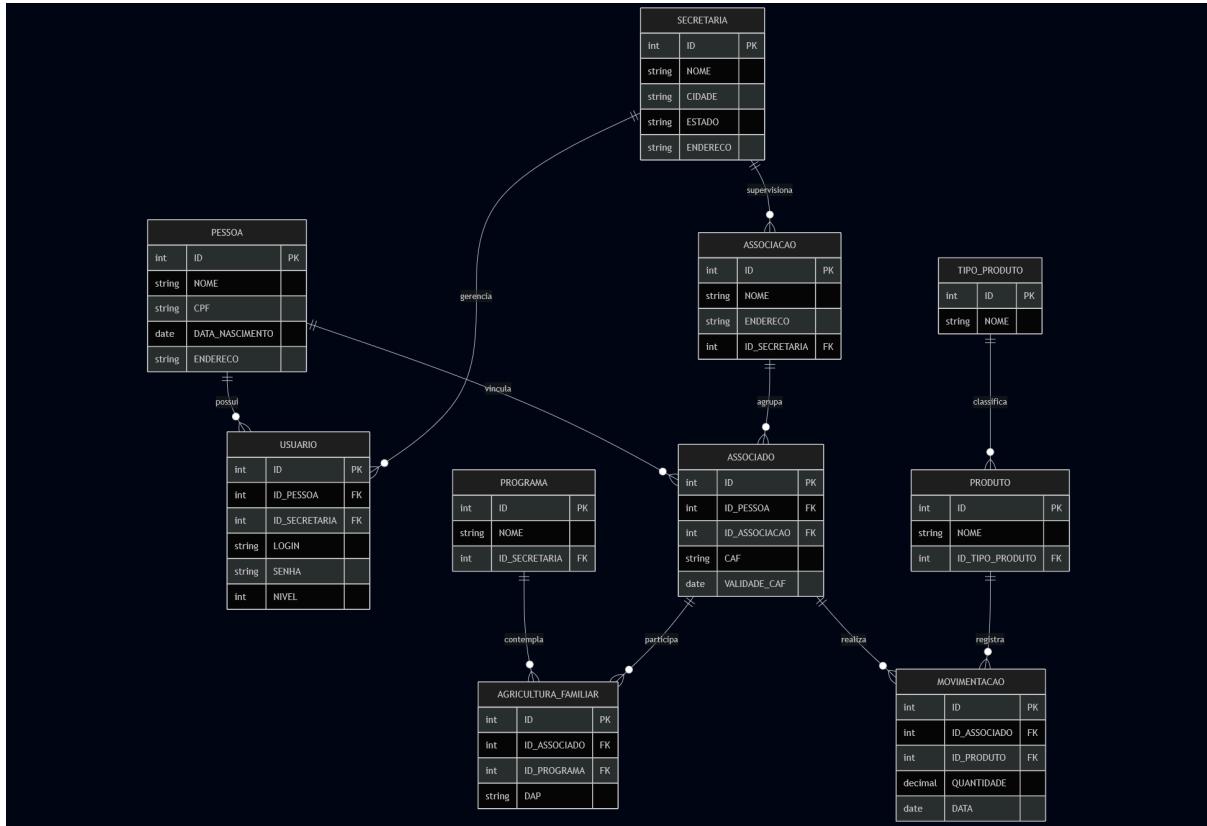
- programa
- agricultura_familiar

Relacionamentos:

- Programa ↔ Associado
- Secretaria ↔ Programa

7.10 Diagrama Entidade–Relacionamento (ER)

O diagrama abaixo representa a **estrutura lógica do banco de dados**, com suas entidades, atributos essenciais e relacionamentos principais.



8. Considerações Finais

O desenvolvimento do projeto **Agro Família Pesca** evidencia a aplicação prática e consistente de conceitos fundamentais e avançados de desenvolvimento **backend**, especialmente no que se refere à arquitetura em camadas, modelagem de banco de dados relacional, controle de acesso baseado em papéis (**RBAC**) e segurança em **APIs RESTful**.

A adoção de uma **estrutura modular**, com clara separação entre Controllers, Services, Repositories e Policies, contribui diretamente para a **manutenibilidade, extensibilidade e legibilidade do sistema**. Essa organização reduz o acoplamento entre componentes, facilita a identificação de responsabilidades e **promove boas práticas** alinhadas à engenharia de software moderna.

Além disso, o uso de escopo de dados baseado no contexto do usuário garante que as informações sejam acessadas de forma **segura e adequada** aos diferentes níveis hierárquicos do sistema (Admin, Secretaria, Associação e Usuário). Tal abordagem demonstra maturidade técnica no tratamento de requisitos de segurança e privacidade, especialmente relevantes em sistemas de natureza institucional e governamental.

Do ponto de vista acadêmico, o projeto atende plenamente aos objetivos propostos, integrando teoria e prática de forma coerente. Do ponto de vista técnico, o sistema apresenta-se como uma base sólida para futuras expansões, podendo evoluir para incluir funcionalidades como:

- Dashboards analíticos para apoio à tomada de decisão;
- Relatórios automatizados de produção e beneficiários;
- Integração com sistemas governamentais externos;
- Camadas adicionais de auditoria e monitoramento.

Dessa forma, o **Agro Família Pesca** não apenas resolve um problema real do domínio da agricultura familiar e pesca artesanal, como também se consolida como um projeto tecnicamente consistente, academicamente válido e **alinhado às boas práticas profissionais** de desenvolvimento de software.

8.1 Considerações Acadêmicas Finais

A vinculação direta entre os módulos da API e as entidades do banco de dados **evidencia uma arquitetura bem definida** e conceitualmente sólida. Essa relação clara entre domínio, persistência e regras de negócio demonstra aderência aos princípios de:

- **Alta coesão modular**, com cada módulo concentrando responsabilidades específicas;
- **Separação de responsabilidades**, evitando sobreposição de lógica entre camadas;
- **Segurança baseada em escopo**, assegurando acesso controlado às informações;
- **Clareza semântica do domínio**, facilitando o entendimento do sistema por desenvolvedores e stakeholders.

Essa estrutura arquitetural atende simultaneamente aos requisitos acadêmicos exigidos em projetos de formação técnica e superior, bem como às boas práticas profissionais adotadas em sistemas corporativos. Assim, o projeto mostra-se adequado para aplicações institucionais e governamentais, servindo como referência para soluções semelhantes no contexto do setor público.