

# DOCUMENTAÇÃO TÉCNICA DO PROJETO – API CADASTRO DE FUNCIONÁRIOS

**Título do Projeto:** Sistema de Cadastro e Gerenciamento de Funcionários

**Grupo:** Fernanda Caroline dos Santos

## 1. Objetivo do Projeto

O objetivo deste projeto é desenvolver uma API RESTful para o cadastro, consulta, atualização e exclusão de informações de funcionários. A aplicação foi concebida para facilitar a gestão de dados funcionais em empresas, permitindo operações simples e eficientes, garantindo a integridade e consistência das informações. A API visa atender demandas de sistemas corporativos que precisam manipular registros de colaboradores de forma rápida e segura.

## 2. Estrutura da Solução

### 2.1 Modelagem de Dados

A modelagem foi centrada na entidade “Funcionário”, contendo os atributos essenciais para identificação e classificação, como Id, Nome e Cargo. Essa modelagem foi implementada como uma classe na camada de domínio e serviu de base para a criação do banco de dados via Entity Framework Core.

### 2.2 Integração da API com a Solução

A API foi estruturada com rotas REST padrão, permitindo operações CRUD (Create, Read, Update, Delete). Cada rota é responsável por uma funcionalidade específica, e a comunicação entre o cliente (front-end ou ferramenta de testes) e o banco de dados é mediada por controladores no ASP.NET Core, garantindo segurança e consistência.

## 3. Organização do Código

O projeto está organizado em camadas e arquivos com responsabilidades definidas para facilitar manutenção e evolução:

- Models/Funcionario.cs – Define a estrutura da entidade Funcionário.
- Data/AppDbContext.cs – Configura a conexão com o banco de dados e integra o Entity Framework Core.
- Controllers/FuncionariosController.cs – Implementa os endpoints da API, lidando com as requisições HTTP.
- Program.cs – Arquivo de inicialização da aplicação, onde são configurados serviços e o pipeline HTTP.
- Migrations/ – Contém as migrações para criação e atualização do banco de dados.

## 4. Justificativa Técnica

A escolha do ASP.NET Core com Entity Framework Core para a construção da API proporciona um ambiente robusto, com suporte nativo a operações assíncronas, facilidade para realizar integrações futuras e escalabilidade.

A modelagem da entidade Funcionário foi feita com base em dados essenciais para cadastro e identificação, garantindo a integridade do sistema.

A separação das responsabilidades entre modelos, contextos e controladores assegura um código organizado, facilitando a manutenção, testes e futuras expansões, como autenticação e autorização.