

DOCUMENTAÇÃO TÉCNICA DO PROJETO

Sistema de Cadastro de Clientes

Integrantes: Peterson Salles

1. Objetivo do Projeto

Este projeto tem como objetivo desenvolver uma API para o gerenciamento de clientes, permitindo operações básicas como cadastrar, consultar, excluir e listar informações. A proposta é criar uma estrutura simples, eficiente e de fácil manutenção, que possa ser utilizada em empresas ou sistemas que necessitem de um controle centralizado de dados de clientes. Todo o desenvolvimento foi realizado com C#, utilizando ASP.NET Core, Entity Framework Core e banco de dados SQLite.

2. Estrutura da Solução

2.1 Modelagem de Dados

A modelagem da solução gira em torno da entidade Cliente. A classe Cliente foi criada com os seguintes atributos:

- Id: identificador único gerado automaticamente;
- Nome: nome completo do cliente;
- Email: endereço eletrônico para contato;
- Telefone: telefone do cliente.

Esses dados foram considerados suficientes para uma estrutura inicial de cadastro. A classe foi usada como base para a criação da tabela no banco de dados, por meio da ferramenta Entity Framework Core.

2.2 Integração da API com a Aplicação

A API foi desenvolvida para seguir os princípios REST, com rotas específicas para cada funcionalidade. O Entity Framework Core foi utilizado para mapear a classe Cliente diretamente no banco de dados SQLite. Além disso, foi integrado o Swagger, permitindo testes rápidos diretamente pelo navegador e facilitando a documentação automática da API. O Swagger gera uma interface gráfica onde é possível executar e validar todas as rotas de forma visual e prática.

3. Endpoints da API

Método	Rota	Descrição
GET	/api/clientes	Lista todos os clientes cadastrados
GET	/api/clientes/{id}	Busca um cliente pelo ID
POST	/api/clientes	Adiciona um novo cliente
DELETE	/api/clientes/{id}	Remove um cliente a partir do ID

4. Organização do Código

O projeto está organizado de forma simples e lógica, respeitando o padrão de separação por responsabilidades. A seguir, explicamos os principais componentes:

- Models/Cliente.cs: Define os atributos da entidade Cliente.
- Data/AppDbContext.cs: Gerencia a conexão com o banco de dados e representa o contexto da aplicação.
- Controllers/CientesController.cs: Contém todas as rotas da API relacionadas à manipulação de clientes.
- Program.cs: Arquivo principal responsável pela configuração e execução da aplicação.
- Api.csproj: Arquivo de configuração do projeto, contendo as dependências necessárias.

5. Justificativa Técnica

A escolha das tecnologias se deu principalmente pela facilidade de uso e pelo fato de fazerem parte do ecossistema moderno de desenvolvimento .NET. O Entity Framework Core reduz a complexidade na comunicação com o banco de dados, enquanto o SQLite garante leveza e simplicidade para armazenamento local. O Swagger, por sua vez, facilitou os testes durante o desenvolvimento e oferece uma forma clara de documentação para outros desenvolvedores ou usuários da API.