

DOCUMENTAÇÃO TÉCNICA DO PROJETO – API ACADEMIA

Título do Projeto: Sistema de Gerenciamento de Equipamentos de Academia

Grupo: Taynara Mafra

Emerson Junior

Kayo Henrique

1. Objetivo do Projeto

O objetivo deste projeto é desenvolver uma API voltada à gestão de equipamentos de academia, utilizando tecnologias modernas de desenvolvimento web. A aplicação visa atender as necessidades de controle de inventário dos aparelhos, permitindo o registro, consulta e remoção de equipamentos. Este modelo é aplicável em academias ou centros esportivos que demandam um controle eficiente e preciso do seu acervo de equipamentos.

2. Estrutura da Solução

2.1 Modelagem de Dados

A entidade central da aplicação é o “Equipamento”. Os atributos definidos foram cuidadosamente selecionados para refletir os principais aspectos de identificação e controle dos aparelhos utilizados em academias, como nome, tipo, marca, modelo, data de aquisição, status de uso e descrição. Essa estrutura foi implementada na camada de domínio e reflete diretamente o esquema da base de dados utilizada, bem como as regras de negócio.

2.2 Integração da API com a Solução

A API é responsável por mediar a comunicação entre o sistema e a base de dados, oferecendo rotas RESTful para as principais operações com equipamentos. Os endpoints foram criados com o intuito de garantir clareza, eficiência e facilidade de integração com clientes diversos, como interfaces web e ferramentas de teste (Postman, Insomnia, etc.).

3. Endpoints da API

Método	Rota	Descrição
GET	/api/equipamentos	Retorna todos os equipamentos
GET	/api/equipamentos/{id}	Retorna um equipamento por ID
POST	/api/equipamentos	Adiciona um novo equipamento
DELETE	/api/equipamentos/{id}	Remove um equipamento por ID

4. Organização do Código

A estrutura do projeto foi organizada por responsabilidade, promovendo legibilidade e manutenção facilitada:•
Models/Equipamento.cs – Define a entidade principal Equipamento, com todos os seus atributos. •
Data/AppDbContext.cs – Responsável pela configuração da base de dados e integração com o Entity Framework. •
Data/DbInitializer.cs – Inicializa e migra o banco de dados durante o startup da aplicação. •
Services/EquipamentoService.cs – Implementa a lógica de negócio e as operações sobre os dados dos equipamentos. •
Controllers/EquipamentosController.cs – Define os endpoints da API e realiza o controle das requisições. •
Program.cs – Ponto de entrada da aplicação, onde os serviços são registrados e a aplicação é configurada.

5. Justificativa Técnica

A modelagem da entidade Equipamento foi baseada em requisitos reais de controle de inventário de academias, contemplando atributos fundamentais como nome, tipo, marca, status e data de aquisição. A separação entre camadas (modelo, serviço, controle e dados) segue os princípios da arquitetura limpa, facilitando a manutenção e escalabilidade do sistema. A adoção do Entity Framework Core com SQLite permite simplicidade na persistência de dados e flexibilidade no ambiente de desenvolvimento. A organização modular do código torna possível a futura adição de funcionalidades, como atualização de registros (PUT) ou autenticação de usuários, com impacto mínimo na estrutura atual.