

UNIVERSIDADE POSITIVO
Curso de Engenharia de Software

API de Contratos – Documentação Técnica e Arquitetural

Matheus Pacheco Pires
Daniel Ramos
Douglas da Silva Mattos Junior
Brayan Cordeiro Rolim

Curitiba
2025

API de Contratos – Documentação Técnica e Arquitetural

Trabalho apresentado à disciplina de Desenvolvimento de Software Visual, ministrada pelo professor Tiago Pedra, como requisito parcial de avaliação.

Alunos:

- Matheus Pacheco Pires
- Daniel Ramos
- Douglas da Silva Mattos Junior
- Brayan Cordeiro Rolim

Curitiba
2025

Sumário

1. Introdução.....	4
2. Objetivo da API.....	5
3. Arquitetura da Solução.....	6
4. Modelagem da Entidade Contrato.....	7
5. Banco de Dados e Migrations.....	8
6. Endpoints da API.....	9
7. Organização do Código.....	10
8. Integração Front-End.....	11
9. Justificativa Técnica.....	12
10. Conclusão.....	13

1. Introdução

Este documento apresenta a documentação técnica e arquitetural da API de Contratos, desenvolvida na disciplina de Desenvolvimento de Software Visual. A API implementa as operações essenciais de um sistema CRUD, permitindo criar, consultar, atualizar e excluir contratos armazenados em um banco de dados SQLite utilizando o Entity Framework Core.

2. Objetivo da API

A API tem como objetivo fornecer uma interface padronizada para o gerenciamento de contratos, permitindo que aplicações externas consumam serviços REST para manipular dados contratuais. O sistema foi projetado para ser simples, eficiente e compatível com boas práticas de desenvolvimento.

3. Arquitetura da Solução

A solução segue uma arquitetura modular composta por:

- ****Modelo (Contrato.cs)**** – classe que representa os dados da aplicação;
- ****Rotas**** – arquivos responsáveis por GET, POST, PUT e DELETE;
- ****Program.cs**** – configurações do servidor e registro do contexto;
- ****Banco SQLite**** – armazenamento local persistente;
- ****Front-End (HTML, CSS e JS)**** – interface visual para uso da API.

A estrutura foi projetada conforme os slides do professor, obedecendo ao padrão MVC simplificado.

4. Modelagem da Entidade Contrato

A entidade utilizada pela API possui os seguintes atributos fundamentais:

- Id (int): identificador único;
- Numero (string): código numérico do contrato;
- Cliente (string): pessoa/empresa contratante;
- DataInicio (DateOnly): início da vigência;
- DataVencimento (DateOnly): fim da vigência;
- RenovacaoAutomatica (bool): indica renovação anual;
- Ativo (bool): calculado automaticamente dependendo do vencimento;
- Descricao (string): observações gerais.

5. Banco de Dados e Migrations

O banco de dados utilizado é o SQLite. As Migrations foram configuradas no Program.cs para que o EF Core crie e atualize automaticamente o schema do banco ao iniciar a aplicação. O arquivo `contratos.db` contém a tabela `Contratos`, gerada a partir da classe Contrato.

6. Endpoints da API

Os endpoints implementados são:

- **GET /api/contratos** → lista todos os contratos;
- **GET /api/contratos/{id}** → retorna um contrato específico;
- **POST /api/contratos** → cria novo contrato;
- **PUT /api/contratos/{id}** → atualiza um contrato existente;
- **DELETE /api/contratos/{id}** → remove um contrato.

7. Organização do Código

O código foi dividido em módulos para facilitar a manutenção:

- Contrato.cs → model principal;
- AppDbContext → conexão ao banco;
- ROTA_GET.cs, ROTA_POST.cs, ROTA_PUT.cs, ROTA_DELETE.cs → organização das rotas;
- Program.cs → configuração geral e inicialização.

8. Integração Front-End

O front-end oferece uma interface funcional construída em HTML, CSS e JavaScript. A interface permite:

- Cadastro de contratos;
- Edição via PUT;
- Exclusão via DELETE;
- Listagem dinâmica com cores por status;
- Busca por ID;
- Renderização com Bootstrap e ícones Bilcons.

9. Justificativa Técnica

A escolha pelo padrão REST e pelo Entity Framework Core foi motivada pela necessidade de simplicidade, escalabilidade e clareza arquitetural. SQLite fornece persistência leve e confiável, ideal para ambientes acadêmicos e

pequenos sistemas. A divisão modular do código facilita futuras expansões e organiza de forma limpa o fluxo da aplicação.

10. Conclusão

A API de Contratos cumpre integralmente todas as exigências descritas nos slides da disciplina. O projeto foi desenvolvido seguindo boas práticas de organização, arquitetura e documentação. O uso de rotas separadas, banco de dados real e integração visual torna a solução completa e pronta para manutenção e evolução.