

# **Desenvolvimento de um Sistema de Gerenciamento de Tutores e Pets com .NET, SQLite e Entity Framework Core**

---

## **1. Introdução**

Este artigo apresenta o desenvolvimento de uma aplicação para gerenciamento de Tutores e Pets, implementada na plataforma .NET com persistência de dados utilizando SQLite e Entity Framework Core. O sistema contempla requisitos funcionais essenciais, como cadastro, atualização, exclusão e consulta de dados, além de adotar padrões de projeto como Repository e injeção de dependência para garantir boas práticas de desenvolvimento. O código-fonte está disponível em repositório público no GitHub, permitindo transparência e reprodutibilidade.

Este possui como objetivo descrever o desenvolvimento de uma aplicação de gerenciamento de Tutores e Pets, aplicada ao contexto acadêmico da disciplina de Modelagem de Software.

## **2. Desenvolvimento**

O desenvolvimento da aplicação foi orientado pelos requisitos funcionais e não funcionais definidos previamente. A seguir, são descritos detalhadamente cada um desses requisitos e como foram atendidos no código.

### **2.1 Requisitos Funcionais**

RF01 - Cadastro, atualização, exclusão e consulta de Tutores: O sistema permite realizar todas as operações CRUD com a entidade Tutor. As operações estão implementadas na camada de repositório TutorRepository, utilizando o Entity Framework Core. Ao executar o programa em um ambiente de depuração, como o Insomnia, é permitido inserção de nome, telefone e e-mail, validando os campos obrigatórios antes do envio.

RF02 - Cadastro, atualização, exclusão e consulta de Pets: Assim como os tutores, os pets possuem seu próprio repositório (PetRepository). O

usuário pode cadastrar nome, espécie e raça do pet. Há associação obrigatória a um tutor, o que garante integridade referencial no banco de dados.

RF03 - Associação entre Pet e Tutor:

Na modelagem de dados, a entidade Pet possui uma propriedade TutorId que referencia a chave primária da entidade Tutor. O Entity Framework Core gerencia essa relação um-para-muitos, garantindo que cada pet pertença a um único tutor. O carregamento de dados utiliza Include() para exibir os pets com os dados dos respectivos tutores.

RF04 - Validação obrigatória de campos:

Foram implementadas validações no escopo do programa para os campos obrigatórios Nome e Telefone dos tutores. O campo telefone é validado quanto ao formato e obrigatoriedade. Caso a validação falhe, a interface bloqueia a operação e exibe mensagens de erro ao usuário.

## 2.2 Requisitos Não Funcionais

RNF01 - Persistência com Entity Framework Core e SQLite:

O projeto utiliza o DbContext do EF Core para realizar mapeamento objeto-relacional (ORM). A base de dados é persistida localmente com SQLite, garantindo portabilidade e simplicidade de implantação.

RNF02 - Repositório público no GitHub:

Todo o código-fonte está disponível no GitHub, no repositório: <https://github.com/MACHADO-Caetano/ModelagemSoftware/tree/main/ap2-modelagem-software/ap2>.

O repositório segue uma estrutura organizada, com separação clara entre camadas (Models, Repositories, Forms, Context).

RNF03 - Boas práticas com padrão Repository e Injeção de Dependências:

O código adota o padrão Repository para encapsular a lógica de acesso a dados. Isso permite maior modularidade e facilidade na manutenção. A injeção de dependência é utilizada na inicialização dos formulários, permitindo que os repositórios sejam passados por meio de construtores, promovendo desacoplamento entre as camadas.

### 3. Conclusão

O desenvolvimento da aplicação permitiu aplicar, na prática, conceitos importantes de modelagem de software e boas práticas de programação. Entre os principais desafios enfrentados estão a configuração correta do EF Core com SQLite e a modelagem das relações entre entidades de forma clara e robusta. A adoção de padrões como Repository e injeção de dependência facilitou a manutenção do código e possibilitou a expansão futura da aplicação. Além disso, a publicação do código em um repositório público no GitHub garante acessibilidade e possibilidade de colaboração.

O projeto demonstra como é possível construir uma aplicação funcional e bem estruturada utilizando tecnologias modernas, promovendo o aprendizado efetivo na área de desenvolvimento de software.

### Referências

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 6022: Informação e documentação – Artigo em publicação periódica científica impressa – Apresentação. Rio de Janeiro, 2018.

MICROSOFT DOCUMENTATION. Entity Framework Core Documentation. Disponível em: <https://learn.microsoft.com/en-us/ef/core/>. Acesso em: mai. 2025.

GITHUB. Repositório do Projeto. Disponível em: <https://github.com/MACHADO-Caetano/ModelagemSoftware>. Acesso em: 26 mai. 2025.