

# Documentação do Sistema: Clinica.Medica

---

## 1. Objetivo

Este documento tem como objetivo descrever a arquitetura, funcionalidades e estrutura do projeto Clinica.Medica, que é uma aplicação ASP.NET Core baseada no padrão Clean Architecture. A aplicação contempla funcionalidades de cadastro de pacientes, triagem, geração de fila de atendimento e gerenciamento do fluxo de atendimentos.

## 2. Requisitos Funcionais

- i. Cadastro de Pacientes com os campos: nome, telefone, sexo, data de nascimento e e-mail.
- ii. Geração de um número sequencial de atendimento para a fila de espera.
- iii. Registro de triagem com sintomas, pressão arterial, peso, altura e especialidade.
- iv. Chamada do paciente de acordo com a ordem de chegada e finalização do atendimento.

## 3. Arquitetura Clean Architecture

O projeto segue os princípios da Clean Architecture, separando claramente as responsabilidades entre as camadas:

- Domain: contém as entidades e regras de negócio.
- Application: contém os casos de uso (services) e DTOs.
- Infrastructure: implementa o acesso a dados via EF Core.
- API: ponto de entrada HTTP com controllers REST.
- WebUI: interface Razor para interação com usuários.

## 4. Diagramas

Os seguintes diagramas descrevem a estrutura e fluxo do sistema:

- Diagrama UML de Classes
- Diagrama de Fluxo de Dados (DFD)
- Modelo Relacional de Dados

## 5. Scripts SQL (DDL + DML)

-- Script DDL

```
CREATE TABLE Pacientes (  
    Id INT PRIMARY KEY IDENTITY,  
    Nome NVARCHAR(100) NOT NULL,  
    DtNascimento DATE NOT NULL,  
    Telefone NVARCHAR(15),  
    Sexo CHAR(1),  
    Email NVARCHAR(100),  
    DataCadastro DATETIME DEFAULT GETDATE()  
);
```

```
CREATE TABLE Atendimentos (  
    Id INT PRIMARY KEY IDENTITY,  
    NumeroSequencial INT NOT NULL,  
    PacienteId INT NOT NULL,  
    DataHoraChegada DATETIME NOT NULL,  
    Status NVARCHAR(50),  
    FOREIGN KEY (PacienteId) REFERENCES Pacientes(Id)  
);
```

```
CREATE TABLE Especialidades (  
    Id INT PRIMARY KEY IDENTITY,  
    Nome NVARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE Triagens (  
    Id INT PRIMARY KEY IDENTITY,  
    AtendimentoId INT NOT NULL,  
    Sintomas NVARCHAR(500),  
    PressaoArterial NVARCHAR(10),  
    Peso FLOAT,  
    Altura FLOAT,  
    EspecialidadeId INT,  
    Prioritario BIT DEFAULT 0,  
    FOREIGN KEY (AtendimentoId) REFERENCES Atendimentos(Id),  
    FOREIGN KEY (EspecialidadeId) REFERENCES Especialidades(Id)  
);
```

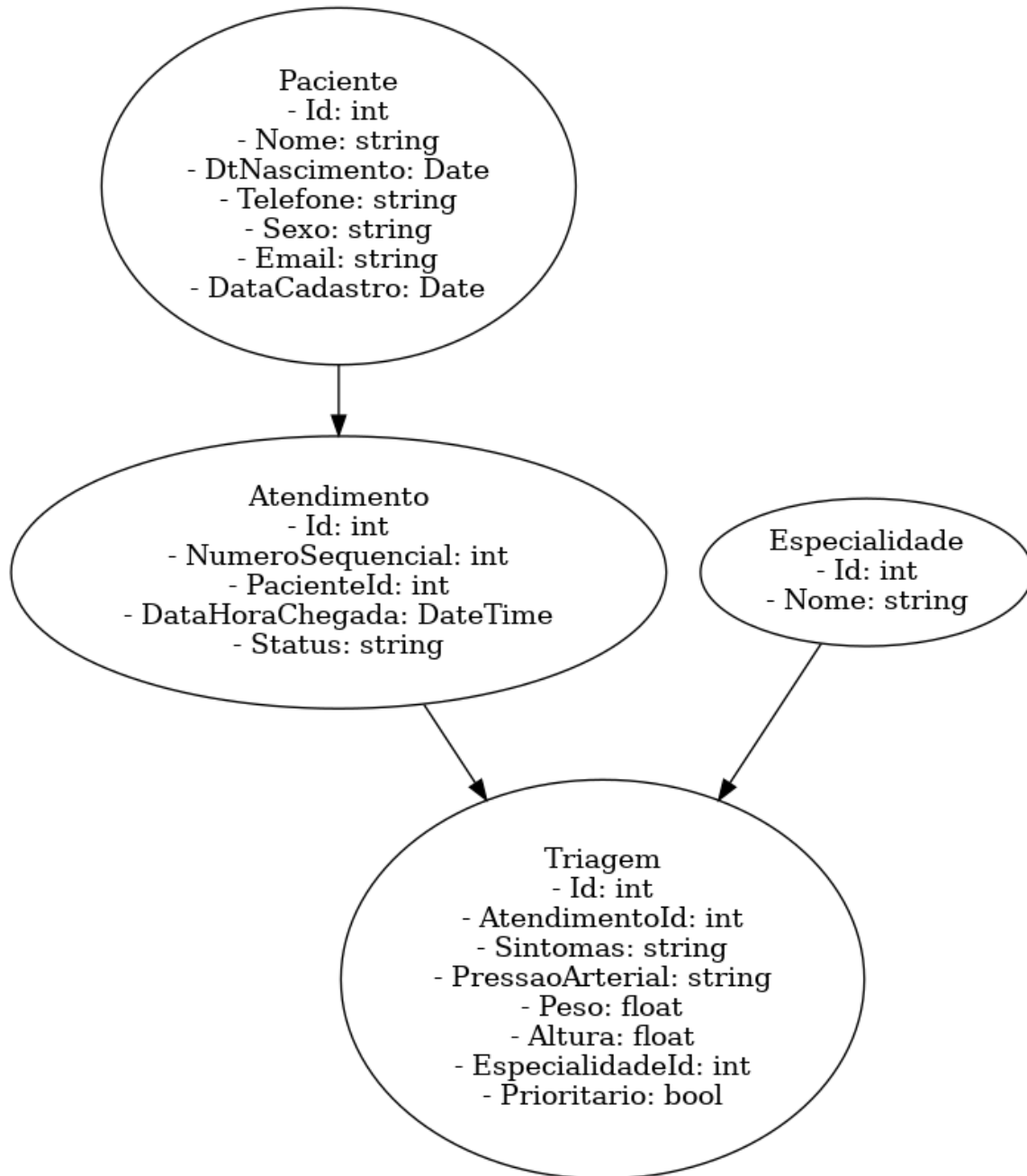
-- Script DML Exemplo

```
INSERT INTO Especialidades (Nome) VALUES ('Cardiologia'), ('Ortopedia'), ('Pediatria');
```

## 6. Diagramas Visuais

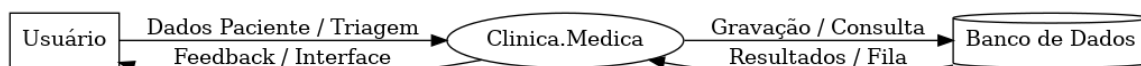
### 6.1 Diagrama UML de Classes

Este diagrama representa as principais entidades do domínio e suas relações.



### 6.2 Diagrama de Fluxo de Dados (DFD - Nível 1)

Este diagrama mostra o fluxo de informações entre o usuário, o sistema e o banco de dados.



### 6.3 Modelo Entidade-Relacionamento (ER)

Representa graficamente como as tabelas do banco de dados se relacionam entre si.

