
Documentação de Projeto

para o sistema

Sistema de Lembretes

Versão 1.0

Projeto de sistema elaborado pelo aluno Rafael de Oliveira Caldeira Lopes
como parte da disciplina **Projeto de Software**.

16/11/2025

Tabela de Conteúdo

| | |
|--|-------------|
| 1. Introdução | 2 |
| 2. Modelos de Usuário e Requisitos | 3-6 |
| 2.1 Descrição de Atores | 3 |
| 2.2 Modelo de Casos de Uso e Histórias de Usuários | 4 |
| 2.3 Diagrama de Sequência do Sistema e Contrato de Operações | 5-6 |
| 3. Modelos de Projeto | 7-12 |
| 3.1 Arquitetura | 7-8 |
| 3.2 Diagrama de Componentes e Implantação. | 9-10 |
| 3.3 Diagrama de Classes | 11 |
| 3.4 Diagramas de Comunicação | 11 |
| 3.5 Diagramas de Estados | 12 |
| 4. Modelos de Dados | 13 |

1. Introdução

Este documento agrega: 1) a elaboração e revisão de modelos de domínio e 2) modelos de projeto para o sistema de lembretes. A referência principal para a descrição geral do problema, domínio e requisitos do sistema é o documento de especificação que descreve a visão de domínio do sistema.

2. Modelos de Usuário e Requisitos

2.1 Descrição de Atores

Nesta subseção é apresentado descrição de cada um dos atores que interagem com o sistema.

- **Usuário**

Descrição: Pessoa que utiliza o sistema para organizar tarefas e compromissos criando lembretes. É responsável por inserir as informações básicas como título, descrição e data e capaz de gerenciar seus próprios lembretes.

Responsabilidades: criar novos lembretes, visualizar a lista de lembretes futuros, excluir lembretes caso não forem mais necessários.

Interesses: manter seus lembretes organizados e atualizados, receber uma listagem ordenada que facilite visualizar o que está mais próximo e evitar a exibição de lembretes já vencidos, mantendo clareza no uso diário.

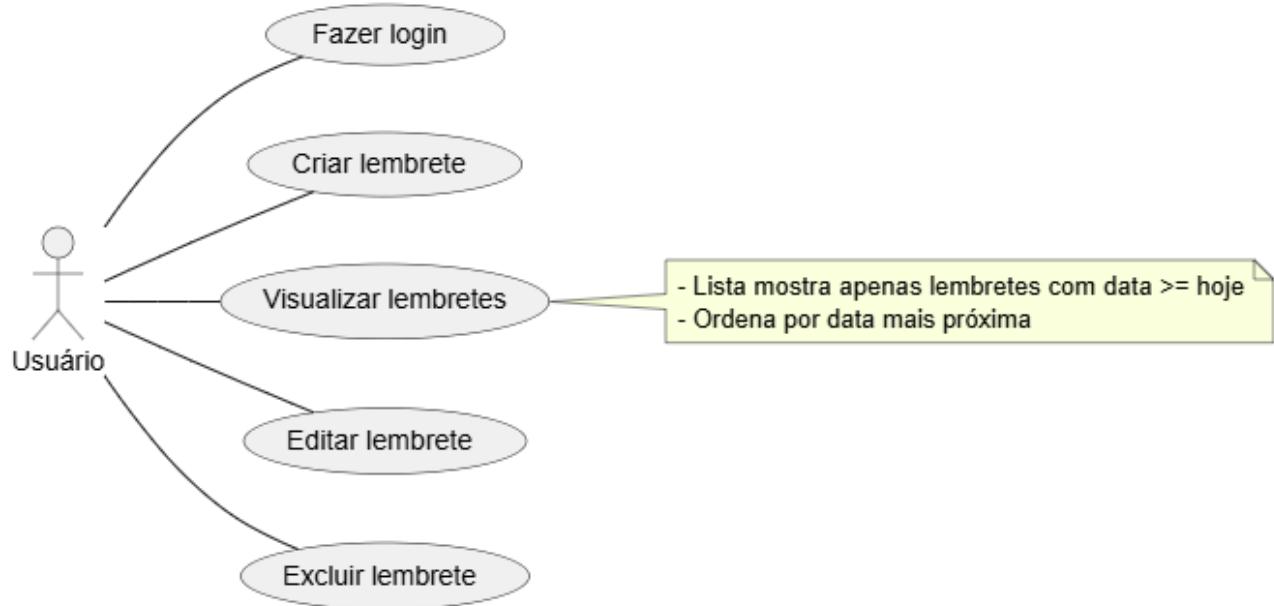
- **Sistema**

Descrição: O sistema representa o conjunto de funcionalidades que processa, armazena e oferece ao usuário os serviços relacionados aos lembretes.

Responsabilidades: validar os dados fornecidos pelo usuário ao criar um lembrete, armazenar lembretes com segurança, ordenar os lembretes pela data mais próxima, filtrar e não exibir os lembretes cuja data já tenha passado e excluir lembretes quando solicitado.

2.2 Modelo de Casos de Uso

Nesta subseção é apresentado o diagrama de casos de uso do sistema e as histórias de usuário.
UC_01 Casos de Uso do Usuário:



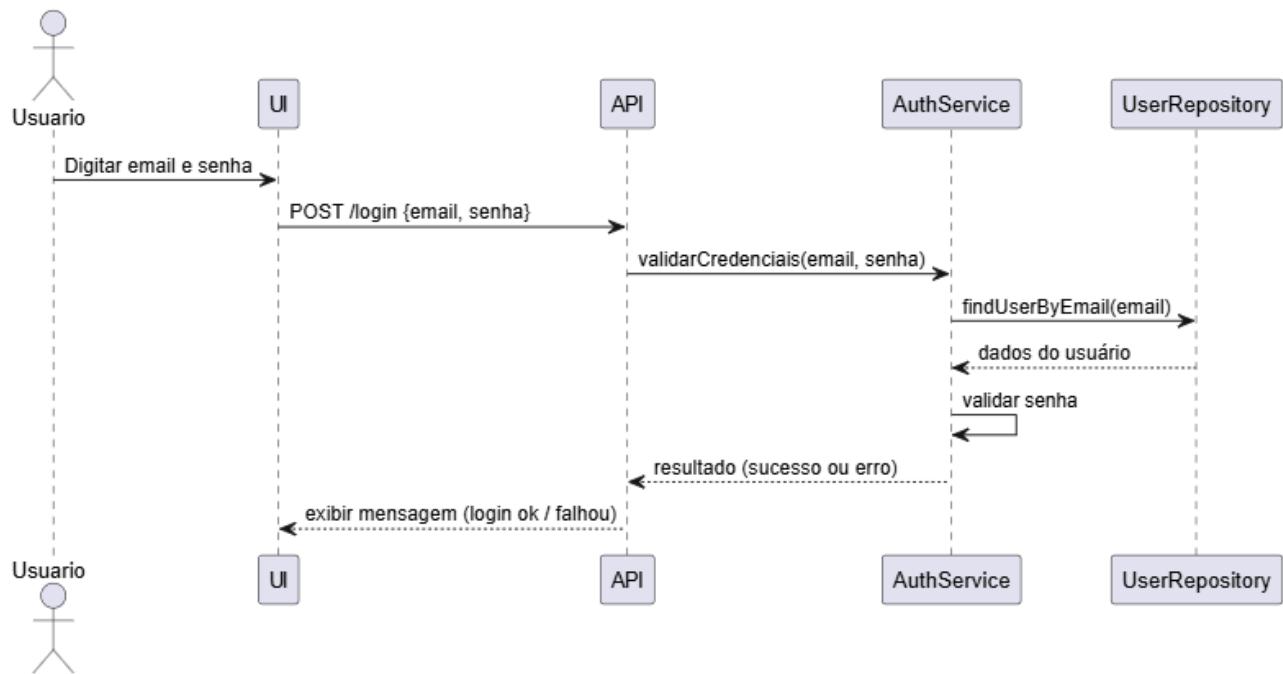
Histórias de usuário:

1. **Como** usuário registrado **eu quero** acessar minha conta no sistema **para que** eu possa gerenciar meus lembretes de forma segura.
2. **Como** usuário autenticado **eu quero** criar um novo lembrete informando título, descrição e data **para que** eu consiga organizar minhas tarefas futuras.
3. **Como** usuário autenticado **eu quero** visualizar meus lembretes em uma lista **para que** eu possa acompanhar as tarefas que ainda precisam ser realizadas.
4. **Como** usuário autenticado **eu quero** editar um lembrete existente **para que** eu possa atualizar informações como título, descrição ou data.
5. **Como** usuário autenticado **eu quero** excluir um lembrete **para que** eu remova tarefas que não são mais necessárias.

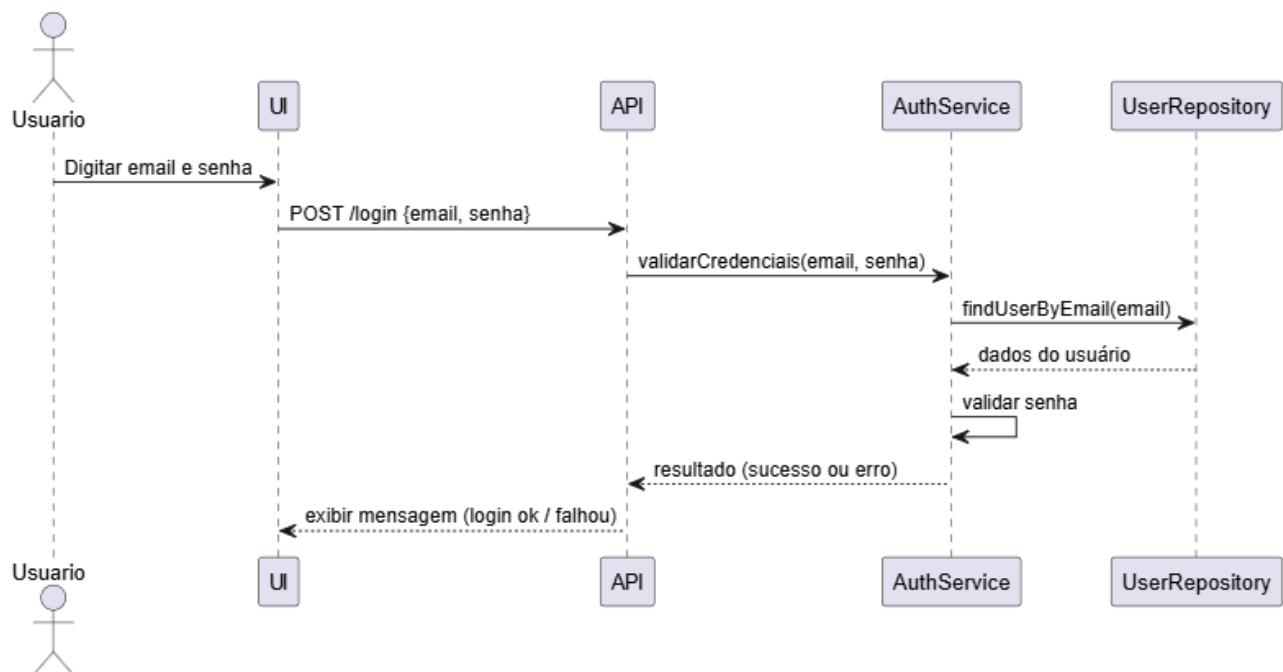
2.3 Diagrama de Sequência do Sistema

Nesta subseção é apresentado os diagramas de sequência do sistema.

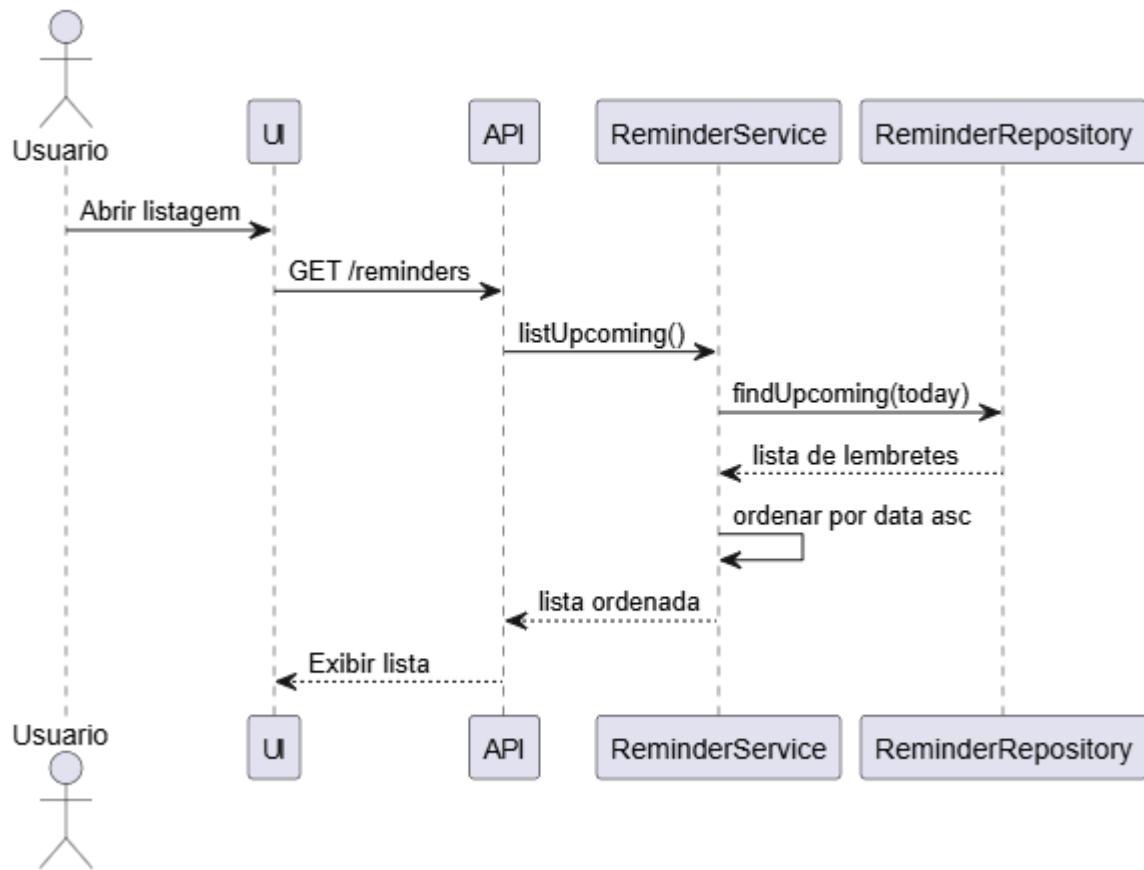
Login:



Criar Lembrete:



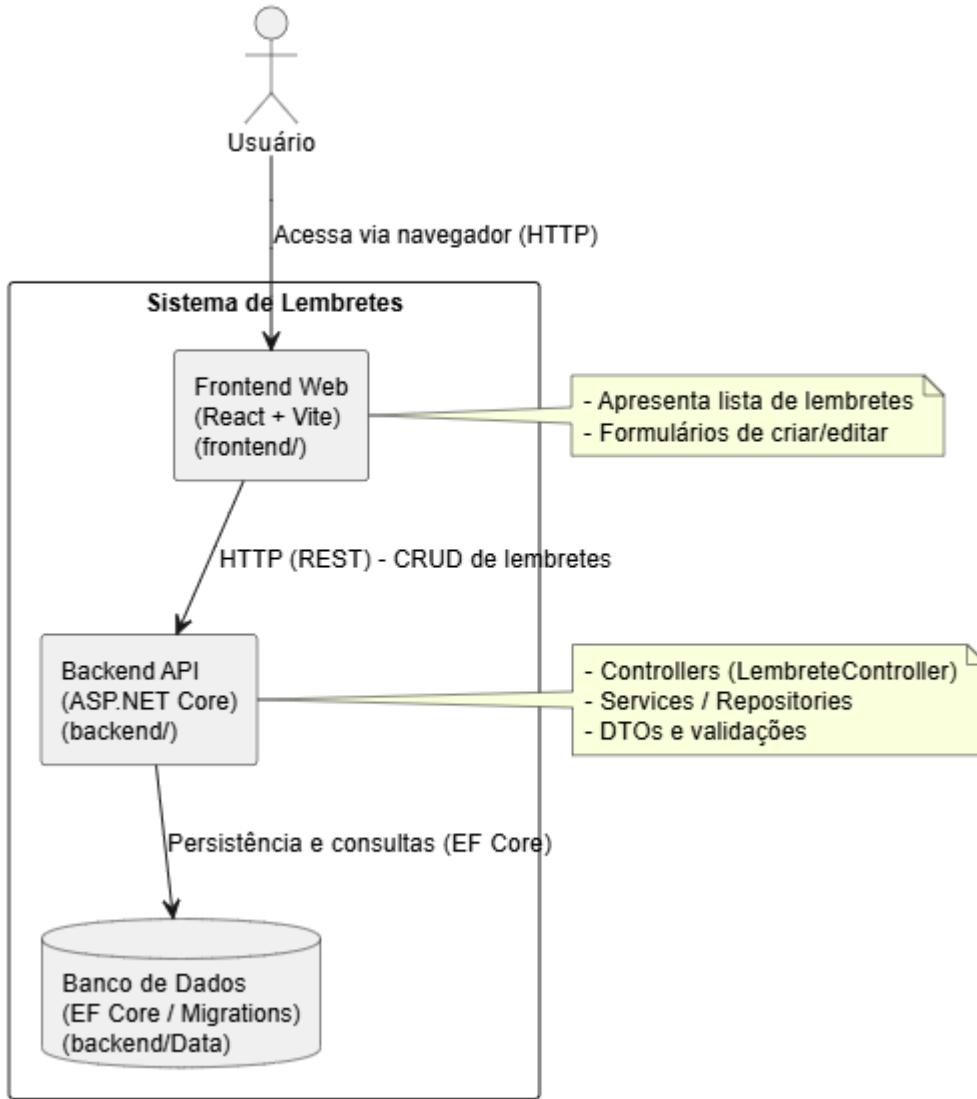
Visualizar Lembretes:



3. Modelos de Projeto

3.1 Arquitetura

Arquitetura descrita com um diagrama da UML.



A arquitetura do sistema de lembretes segue um modelo baseado em **três camadas principais**, organizadas em containers conforme o **C4 Model – Nível 2**:

1. Frontend (Aplicação Web)

- Local: pasta `frontend/`
- Tecnologias: React, Vite, Jest
- Responsável por:
 - Exibir interface gráfica
 - Coletar ações do usuário

- Consumir a API do backend via HTTP
- Renderizar lista de lembretes
- Enviar operações de CRUD

2. Backend (API REST)

- Local: pasta `backend/`
- Tecnologias: ASP.NET Core, Entity Framework Core
- Componentes principais:
 - `Controllers/` — endpoints (ex.: `LembreteController`)
 - `Models/` — entidades como `Lembrete.cs`
 - `DTOs/` — transferência de dados
 - `Repositories/` — acesso ao banco (`LembreteRepository`)
 - `Interfaces/` — abstrações de repositórios
 - `Data/` — `AppDbContext` e EF Core
- Responsável por:
 - Lógica de negócio
 - Validação (data, campos obrigatórios etc.)
 - Acesso ao banco via repositórios
- Exposição dos endpoints:
 - POST /lembretes
 - GET /lembretes
 - PUT /lembretes/{id}
 - DELETE /lembretes/{id}

3. Banco de Dados

- Gerenciado por: SQL Server
- EF Core Migrations (`backend/Migrations/`)
- `AppDbContext` (`backend/Data/AppDbContext.cs`)
- Função: Persistir lembretes, consultar lembretes futuros e excluir e atualizar dados

3.2 Diagrama de Componentes e Implantação.

Diagramas de componentes e implantação do sistema.

Diagrama de Componentes:

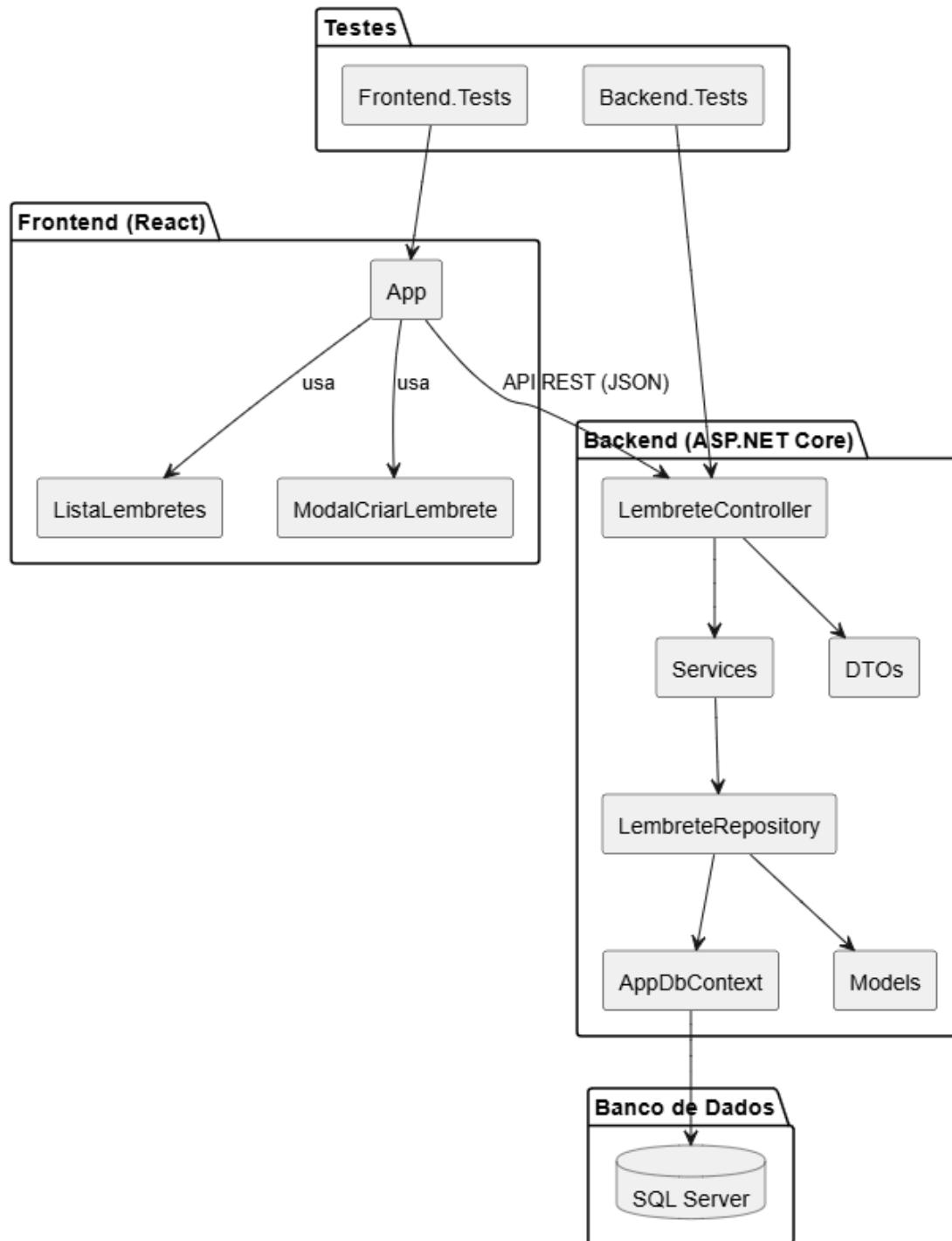
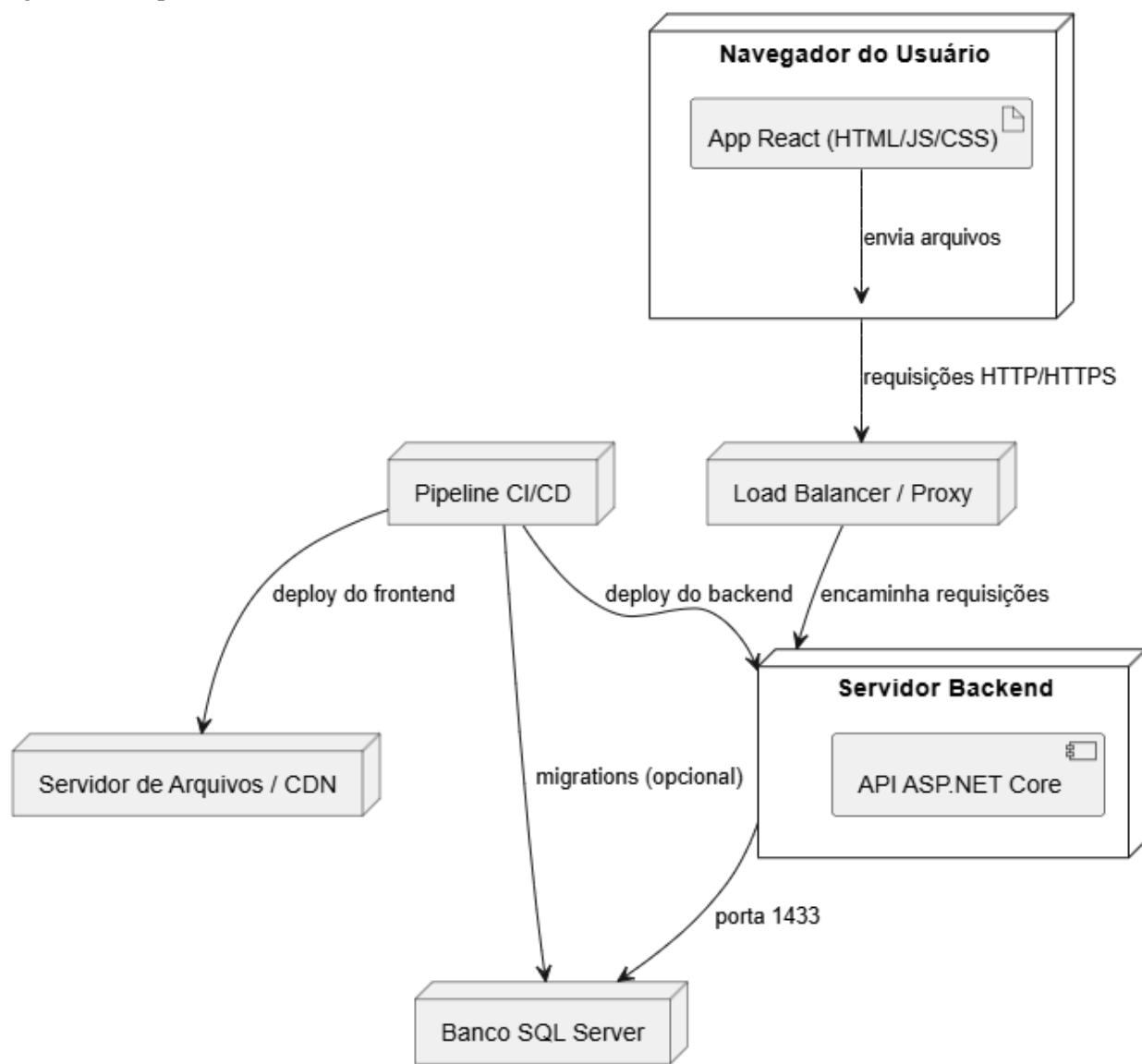
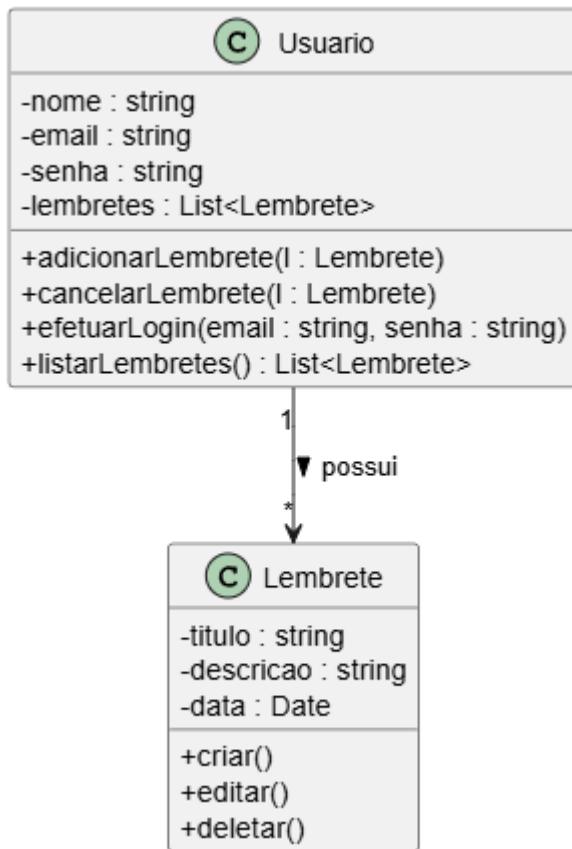


Diagrama de Implantação:



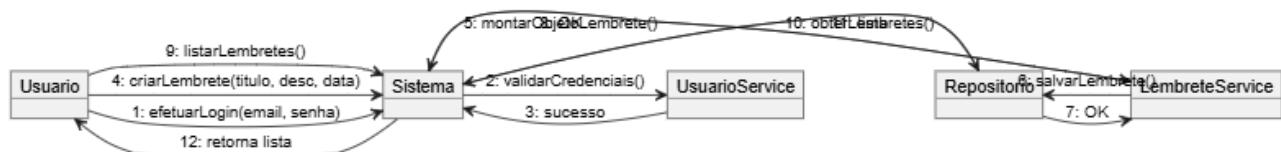
3.3 Diagrama de Classes

Diagrama de classes do sistema.



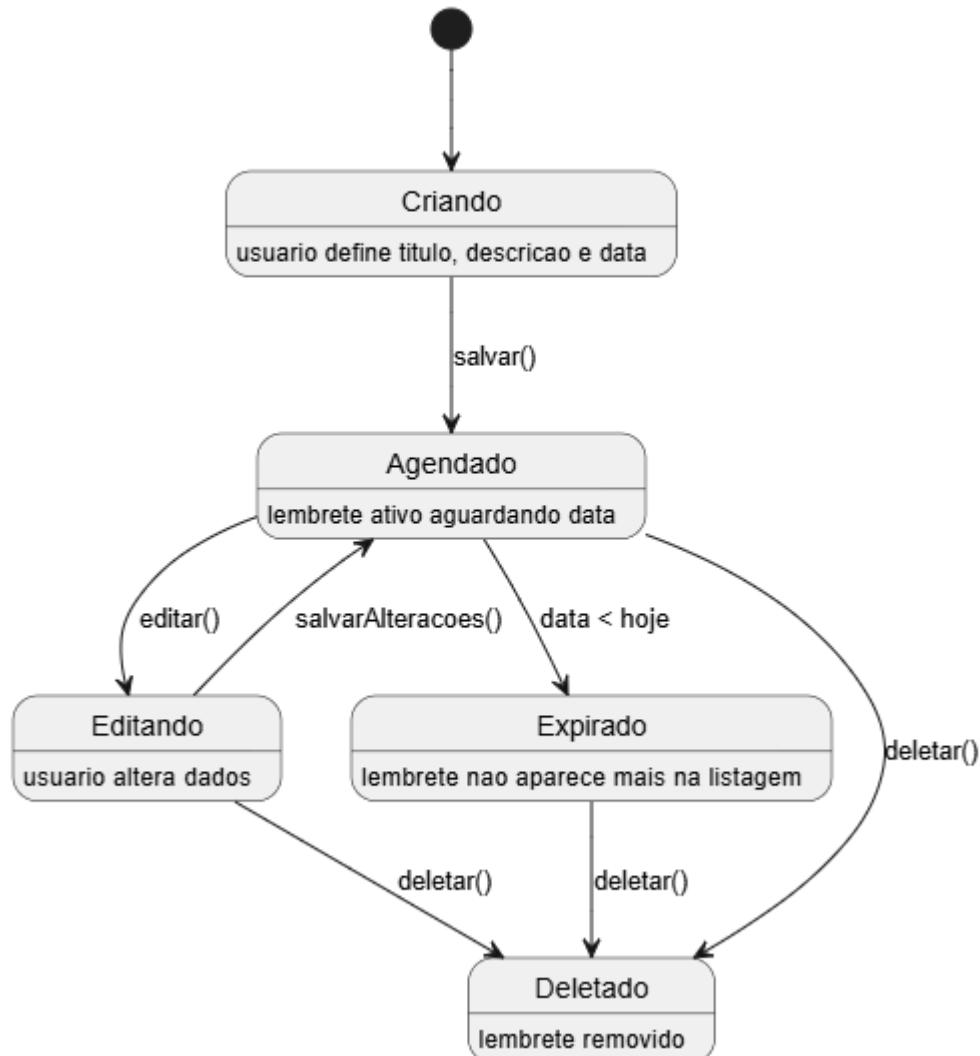
3.4 Diagramas de Comunicação

Diagramas de comunicação para realização de casos de uso.



3.5 Diagramas de Estados

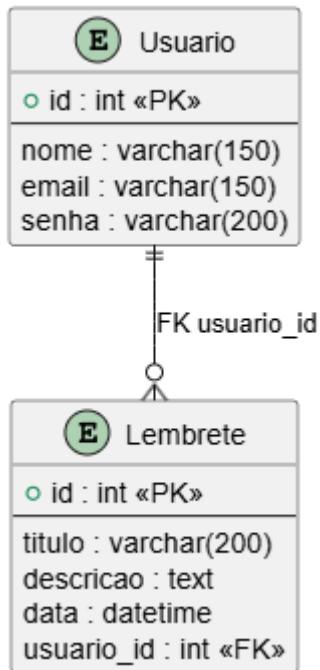
Diagramas de estados do sistema.



4. Modelos de Dados

Aqui existem os modelos de dados, contendo o Diagrama Relacional e Diagrama de Entidade e Relacionamento.

Diagrama Relacional:



DER:

