



Lembre Bem: Sistema de Gestão de Medicamentos para Cuidadores e Familiares

Euritales Silva dos Santos

Flávia Maristela S. Nascimento
Orientadora

Instituto Federal da Bahia – IFBA
Curso de Análise e Desenvolvimento de Sistemas
Campus Salvador

Salvador, Bahia, Brasil
Agosto 2024

SUMÁRIO

1.1 Declaração do Problema	4
1.2 Proposta de Solução de Software	5
1.3 Tecnologias Adotadas	6
1.3.1 Back-end	6
1.3.1.1 Kotlin	6
1.3.1.2 Spring Boot	6
1.3.2 Front-end	7
1.3.2.1 NextJS	7
1.3.2.2 TypeScript	7
1.3.2.3 ChakraUI	7
1.3.3 Banco de Dados	7
1.3.3.1 MySQL	7
1.3.4 Versionamento	8
1.3.4.1 Git	8
1.3.5 Testes	8
1.3.5.1 Jest	8
1.3.5.2 Insomnia	8
1.4 Trabalhos Relacionados	8
1.4.1 Medisafe	8
1.4.2 Care Zone	9
1.4.3 MyTherapy	10
1.4.4 Tabela comparativa	10
2. Requisitos	11
2.1 Requisitos Funcionais	11
2.2 Requisitos Não-Funcionais	12
3. Design	13
3.1 Projeto UML	13
3.1.1 Diagrama de Classe	13
3.1.2 Diagrama de Caso de Uso	14
3.1.2.1 Diagrama de Caso de Uso - Gerenciamento de Dependentes	14
3.1.2.2 Diagrama de Caso de Uso - Gerenciamento de Tratamentos	15
3.1.2.3 Diagrama de Caso de Uso - Gerenciamento de Lembretes	15
3.2 Visão Arquitetural	16
3.3 Modelo de Banco de Dados	18
3.3.1 Modelo de Dados Lógico	18
3.3.2 Modelo de Dados Físico	19
4. Testes de Software	20
4.1 Projeto de Testes	20

Caso de Teste 1	20
Caso de Teste 2	21
Caso de Teste 3	22
Caso de Teste 4	22
Caso de Teste 5	23
Caso de Teste 6	24
Caso de Teste 7	25
Caso de Teste 8	25
Caso de Teste 9	26
Caso de Teste 10	27
5. Implantação	29
5.1 Projeto de Implantação	29
6. Manual do Usuário	30
Agradecimentos	36
Referências	37

1. Visão Geral

1.1 Declaração do Problema

As doenças incapacitantes, como Alzheimer e Parkinson, têm um impacto significativo na população, não apenas sobre os portadores, mas também sobre os familiares que em alguns casos se tornam cuidadores. A natureza dessas condições exige cuidados intensivos contínuos, resultando em desafios emocionais e físicos tanto para o paciente quanto para seu cuidador.[1] Segundo a *Alzheimer's Disease International* (ADI), uma organização sem fins lucrativos registrada nos EUA, cerca de 50 milhões de pessoas são afetadas globalmente, com estimativas alarmantes de 74,7 milhões em 2030 e 131,5 milhões em 2050. No Brasil, aproximadamente 1,2 milhões de pessoas vivem com alguma forma de demência, e são diagnosticados cerca de 100 mil novos casos anualmente [2].

Quando uma doença incapacitante se manifesta em um membro da família, o impacto sobre a dinâmica familiar é imediato. A rotina precisa ser reconfigurada para atender às novas necessidades da pessoa portadora da doença, demandando um ajuste emocional e de atividades. À medida que a doença progride, aumenta-se a exigência sobre os cuidados necessários, uma dedicação muitas vezes além da capacidade ou preparo dos familiares, que enfrentam a difícil decisão de abdicar de suas vidas pessoais para cuidar do ente querido.

No Brasil, a maioria dos cuidadores é composta por familiares que, em sua grande parte, não possuem formação específica para lidar com as complexidades dessas doenças. Essa falta de preparo agrava a situação, pois muitas vezes o cuidador não está apto para gerenciar de forma eficaz as necessidades diárias do paciente, resultando em um desgaste físico e emocional significativo. De acordo com uma reportagem da BBC Brasil, além do impacto emocional, há também um custo econômico, pois muitos cuidadores precisam reduzir sua carga de trabalho ou até mesmo abandonar seus empregos para se dedicar integralmente ao cuidado do ente querido [3].

Além dos desafios emocionais e financeiros, os cuidadores frequentemente enfrentam isolamento social. O tempo necessário para cuidar de um familiar com

Alzheimer ou outra condição incapacitante pode restringir a capacidade do cuidador de manter relações sociais e realizar atividades fora de casa, o que pode levar ao aumento dos níveis de estresse e depressão. Um estudo discutido pelo blog de saúde mental da Vittude, ressalta que a maioria dos cuidadores acabam sofrendo com a "síndrome do cuidador", caracterizada por cansaço extremo, problemas de saúde mental, e um alto nível de estresse [4].

A rotatividade de profissionais de cuidado, quando contratados, também é um problema recorrente. Segundo o estudo publicado na SciELO, essa rotatividade pode se dar pela sobrecarga de trabalho ou pela falta de preparo para lidar com a complexidade das demandas. Cada novo cuidador precisa de um período de adaptação para entender as rotinas e particularidades do paciente, o que pode prolongar o tempo necessário para oferecer um cuidado eficaz [5].

1.2 Proposta de Solução de Software

Diante das diversas complexidades e desafios no que se diz respeito ao cuidado de pessoas com condições incapacitantes, o "Lembre Bem" se apresenta como uma solução tecnológica voltada a oferecer apoio no gerenciamento de medicamentos e atividades diárias. O objetivo do sistema é complementar os cuidados já existentes, proporcionando uma ferramenta prática e acessível que facilite a organização e execução das tarefas diárias de cuidado.

O "Lembre Bem" reúne as informações essenciais de dependentes, que são pessoas que necessitam dos cuidados, e de seus tratamentos, de maneira clara e intuitiva, o que ajuda a reduzir o tempo de adaptação dos cuidadores em uma rotina de cuidados, especialmente em cenários onde há alta rotatividade de profissionais e também quando familiares responsáveis pelos cuidados precisam delegar temporariamente suas atividades a outra pessoa. Isso ajuda na realização de uma transição mais eficiente entre cuidadores, minimizando o risco de erros e contribuindo para que os pacientes recebam seus cuidados de maneira adequada.

A ferramenta busca melhorar a qualidade de vida dos pacientes e facilitar o trabalho dos cuidadores através do suporte contínuo, tornando a rotina de cuidados mais

mais gerenciáveis e seguras. Com isso, o "Lembre Bem" visa ser um aliado no dia a dia dos cuidadores, proporcionando mais organização e tranquilidade no cumprimento das atividades de cuidado.

1.3 Tecnologias Adotadas

Para o desenvolvimento da solução de software, foram adotadas tecnologias que melhor se adequam à proposta do "Lembre bem". Essas tecnologias foram escolhidas pela suas capacidades em atender às necessidades específicas da ferramenta "Lembre Bem":

1.3.1 Back-end

1.3.1.1 Kotlin

Kotlin [6] é uma linguagem de programação moderna, expressiva e concisa, que proporciona segurança e interoperabilidade total com o Java. Sua escolha se deve à sua sintaxe clara e à capacidade de reduzir erros comuns de *NullPointerException* (NPE), que ocorre quando se tenta acessar um objeto nulo, e erros de tipo, que são minimizados devido ao forte sistema de tipos do Kotlin.

1.3.1.2 Spring Boot

Spring Boot [7] é um *framework* poderoso e flexível para o desenvolvimento de aplicações Java. Ele facilita a criação de serviços RESTful, que são fundamentais para a comunicação entre o *front-end* e o *back-end* dentro da aplicação.

A integração desse framework com Kotlin é bem interessante devido à sintaxe concisa do Kotlin, que reduz a quantidade de código necessário. O Spring Boot oferece uma configuração mínima e práticas recomendadas para o desenvolvimento, o que acelera o processo de criação e implementação de funcionalidades.

1.3.2 Front-end

1.3.2.1 NextJS

Next.js [8] é um framework para *React* que tem uma capacidade de melhorar o desempenho do sistema e a experiência do usuário, proporcionando uma navegação mais rápida e eficiente. Next.js simplifica o desenvolvimento de interfaces complexas e reativas, garantindo que a aplicação "Lembre Bem" seja intuitiva e fácil de usar para os cuidadores.

1.3.2.2 TypeScript

TypeScript [9] é uma extensão do JavaScript que adiciona tipagem estática ao código, facilitando a detecção de erros e melhorando a manutenção do código ao longo do tempo, contribuindo para um desenvolvimento mais robusto e previsível.

1.3.2.3 ChakraUI

Chakra UI [10] é uma biblioteca de componentes modular para o React[16], que fornece blocos de construção necessários para criação de aplicativos responsivos de forma simples e rápida.

1.3.3 Banco de Dados

1.3.3.1 MySQL

MySQL [11] é um sistema de gerenciamento de banco de dados relacional muito utilizado e com grande confiabilidade, desempenho e facilidade de uso. Oferece suporte a grandes volumes de dados e consultas complexas, tratamentos e lembretes da aplicação. Além disso, o MySQL possui uma comunidade de desenvolvedores e uma ampla documentação, facilitando a resolução de eventuais problemas e a implementação de boas práticas.

1.3.4 Versionamento

1.3.4.1 Git

O Git [12] é um sistema de controle de versão distribuído que permite rastrear alterações em arquivos e colaborar com outros desenvolvedores de forma eficiente. Ele facilita desenvolvimento mais eficaz ao permitir a criação de *branches* (ramificações) para desenvolver funcionalidades de forma isolada, que podem ser mescladas ao projeto principal.

1.3.4.2 Github

O GitHub [13] é uma plataforma online que hospeda repositórios Git, permitindo o armazenamento, compartilhamento e colaboração em projetos de software.

O código da API pode ser encontrado no repositório:

<https://github.com/euritales/Projeto-TCC-Lembre-Bem>

1.3.5 Testes

1.3.5.1 Jest

O Jest [14] é um *framework* de teste em *JavaScript* amplamente utilizado para testar aplicações web construídas com bibliotecas e *frameworks* como React, Vue.js e Next.js. Ele permite realizar testes unitários, de integração e de ponta a ponta.

1.3.5.2 Insomnia

O Insomnia [15] é uma ferramenta que auxilia no desenvolvimento de APIs, bastante utilizada principalmente para testar as requisições HTTPs realizadas por um sistema. Oferece uma interface gráfica para criação, envio e visualização de requisições HTTP, permitindo que desenvolvedores interajam com APIs RESTful e GraphQL.

1.4 Trabalhos Relacionados

1.4.1 Medisafe

Medisafe é uma aplicação popular para gerenciamento de medicamentos. Ela permite aos usuários configurar lembretes para a administração de medicamentos, rastrear a adesão ao tratamento e compartilhar informações com cuidadores ou profissionais de saúde.

Vantagens:

- Gerenciamento de Medicamentos: Oferece funcionalidades robustas para o gerenciamento de medicamentos, com lembretes e rastreamento de adesão ao tratamento;
- Interface Amigável: Possui uma interface intuitiva e fácil de usar, adequada para cuidadores de diferentes níveis de experiência;
- Fácil Disponibilidade: Disponível em várias plataformas e fácil de acessar;

Desvantagens:

- Adição de Dependentes: Funcionalidade limitada na versão gratuita;

1.4.2 Care Zone

CareZone é uma aplicação que oferece uma plataforma para o gerenciamento de medicamentos e atividades diárias, além de permitir o compartilhamento de informações entre cuidadores e familiares.

Vantagens:

- Gerenciamento de Medicamentos e Atividades: Suporta tanto o gerenciamento de medicamentos quanto de atividades diárias, oferecendo uma solução mais completa;

Desvantagens:

- A interface complexa para usuários menos experientes;
- Adição de Dependentes: Não oferece suporte para a adição de múltiplos dependentes;
- Disponibilidade: Não possui mais disponibilidade nas lojas virtuais;

- Idioma em Português: Não está disponível em português, sendo uma barreira para os usuário;

1.4.3 MyTherapy

MyTherapy é uma aplicação focada no gerenciamento de medicamentos e acompanhamento da saúde. Ela oferece lembretes de medicamentos, rastreamento de sintomas e relatórios de saúde.

Vantagens:

- Gerenciamento de Medicamentos: Fornece lembretes para medicamentos e rastreamento de sintomas, ajudando na adesão ao tratamento;
- Interface Amigável: Fácil de usar, adequada para uma ampla gama de usuários;
- Fácil Disponibilidade: Disponível em várias plataformas e fácil de acessar;
- Idioma totalmente traduzido para o português;

Desvantagens:

- Adição de Dependentes: O aplicativo é voltado a um único usuário, realizando o autogerenciamento, não permitindo a adição de dependentes.

1.4.4 Tabela comparativa

Características	MedSafe	CareZone	My Therapy	Lembre Bem
Gerenciamento De Medicamentos	Sim	sim	sim	sim
Gerenciamento de Atividades	Não	Sim	Não	Sim
Interface Amigável	Sim	Não	Sim	Sim
Adição de Dependentes	versão paga	Não	Não	Sim
Fácil Disponibilidade	Sim	Não	Sim	Sim
Idioma em Português	Sim	Não	Sim	Sim

2. Requisitos

2.1 Requisitos Funcionais

[RF1] Como cuidador, eu preciso cadastrar dependentes no sistema com informações como nome, data de nascimento, sexo, comorbidade, peso e observações.

[RF2] Como cuidador, eu preciso atualizar as informações dos dependentes conforme necessário, para manter os dados sempre corretos e atualizados.

[RF3] Como cuidador, eu preciso excluir dependentes do sistema quando não forem mais necessários, para manter a base de dados limpa e organizada.

[RF4] Como cuidador, eu preciso cadastrar tratamentos para os dependentes, especificando nome, quantidade, unidade e tipo (atividade ou medicamento).

[RF5] Como cuidador, eu preciso atualizar as informações dos tratamentos conforme necessário, para refletir mudanças nas prescrições ou nas necessidades dos dependentes.

[RF6] Como cuidador, eu preciso excluir tratamentos que não são mais necessários, para manter a lista de tratamentos atualizada e relevante.

[RF7] Como cuidador, eu preciso criar lembretes para os dependentes, definindo horários, dias, dosagens e status para garantir a administração correta de medicamentos e realização de atividades.

[RF8] Como cuidador, eu preciso atualizar os lembretes conforme necessário, para ajustar os horários e doses conforme mudanças nas prescrições ou nas necessidades dos dependentes.

[RF9] Como cuidador, eu preciso excluir lembretes que não são mais necessários, para manter a agenda de lembretes limpa e organizada.

[RF10] Como cuidador, eu preciso gerenciar o estoque de medicamentos, recebendo alertas de baixa quantidade para evitar a interrupção dos cuidados.

2.2 Requisitos Não-Funcionais

[RNF1] O sistema deve estar disponível a todo tempo para garantir acesso constante às informações dos dependentes;

[RNF2] O sistema deve ser intuitivo e fácil de usar, com uma interface amigável para que cuidadores de diferentes níveis de habilidade possam utilizá-lo sem dificuldades;

[RNF3] O sistema deve ser acessível nas plataformas web;

[RNF4] O sistema deve proteger as informações pessoais e de saúde dos dependentes para garantir a privacidade e a segurança dos dados.

[RNF5] O sistema deve ter uma tela responsiva e se adaptar a dispositivos móveis.

3. Design

3.1 Projeto UML

3.1.1 Diagrama de Classe

O diagrama de classe é uma representação visual da estrutura do sistema, mostrando as classes que o compõem, seus atributos, métodos e as relações entre as classes. Através dele conseguimos entender como os elementos do sistema se conectam e interagem entre si.

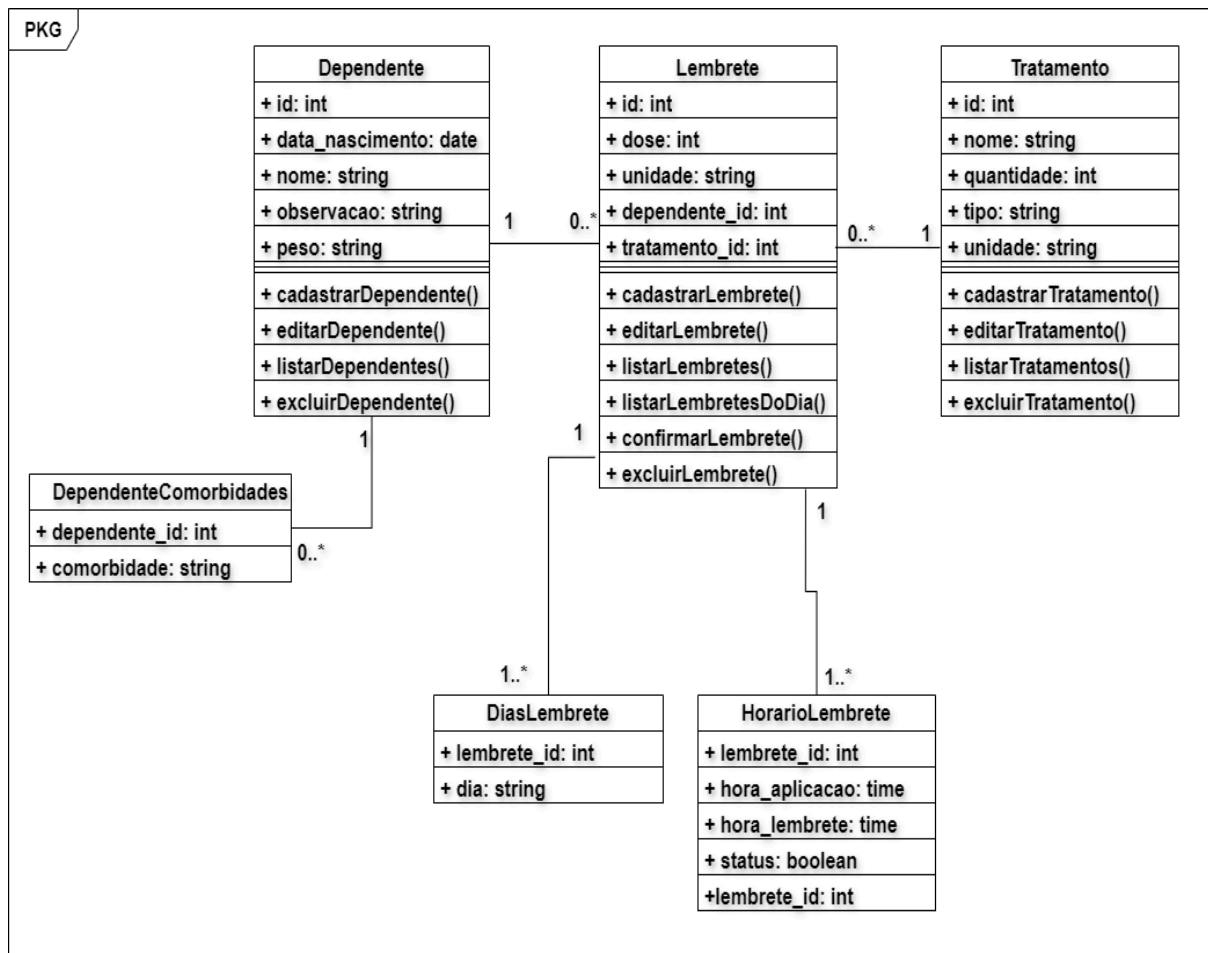


Figura 1 - Diagrama de Classe

3.1.2 Diagrama de Caso de Uso

O diagrama de caso de uso é uma representação visual das principais funcionalidades do sistema, e das interações que o usuário terá com as funcionalidades dele. Nele podemos identificar diferentes que o sistema oferece, permitindo uma melhor compreensão de como o sistema atende às necessidades do usuário.

3.1.2.1 Diagrama de Caso de Uso - Gerenciamento de Dependentes

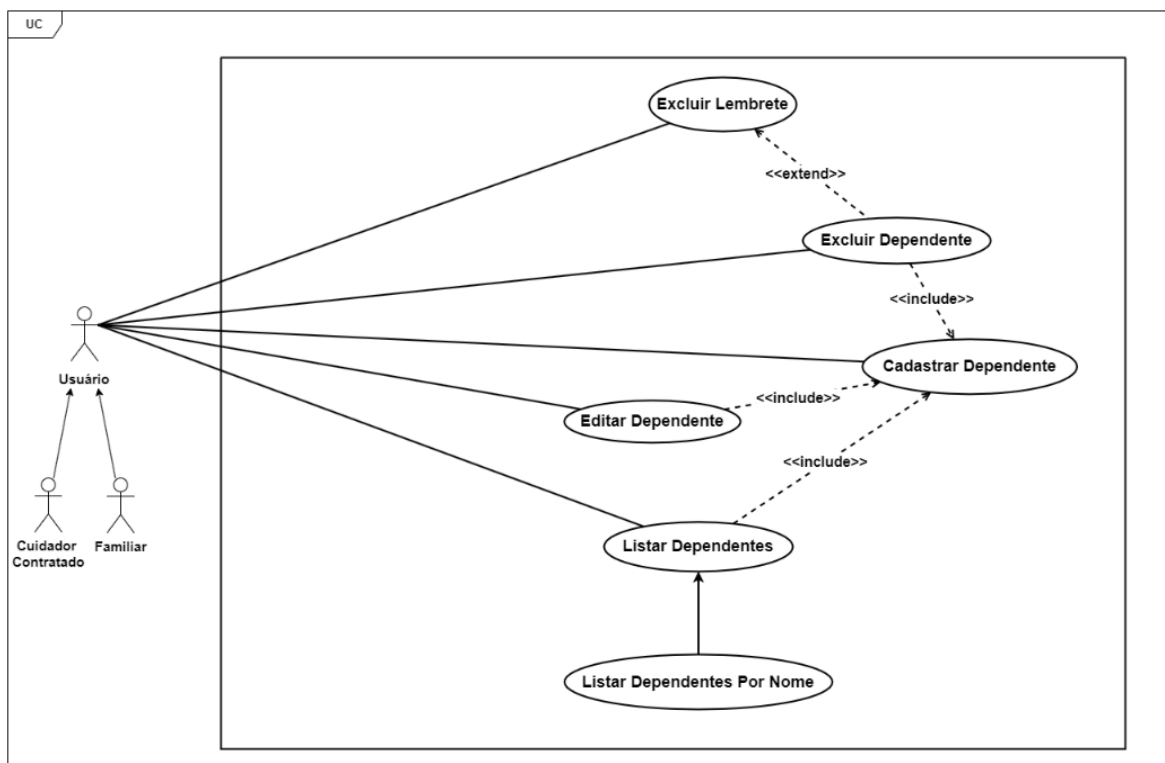


Figura 2 - Diagrama de Caso de Uso - Gerenciamento de Dependentes

3.1.2.2 Diagrama de Caso de Uso - Gerenciamento de Tratamentos

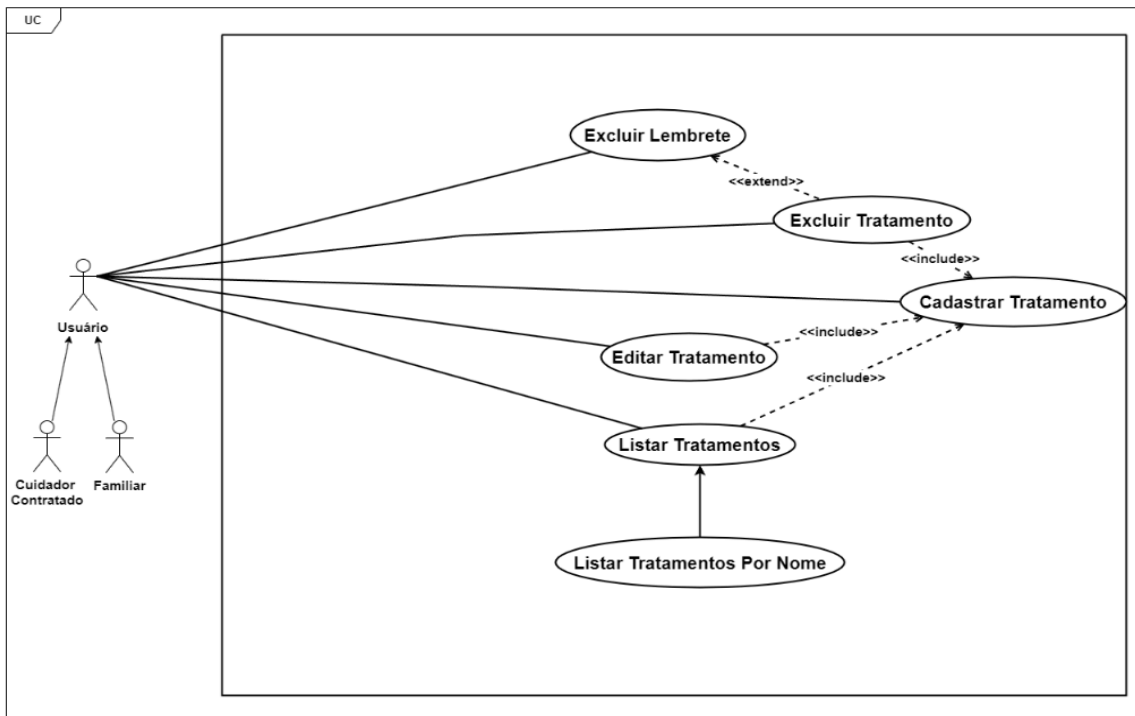


Figura 3 - Diagrama de Caso de Uso - Gerenciamento de tratamentos

3.1.2.3 Diagrama de Caso de Uso - Gerenciamento de Lembretes

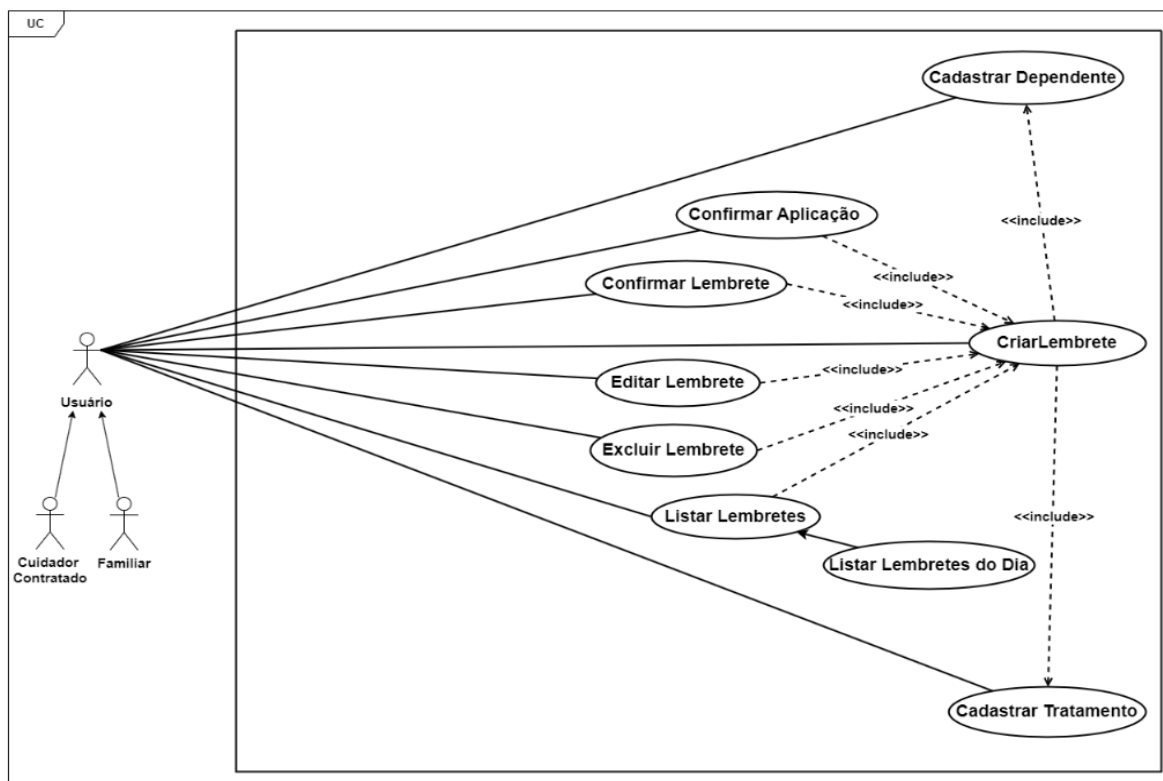


Figura 4 - Diagrama de Caso de Uso - Gerenciamento de Lembretes

3.2 Visão Arquitetural

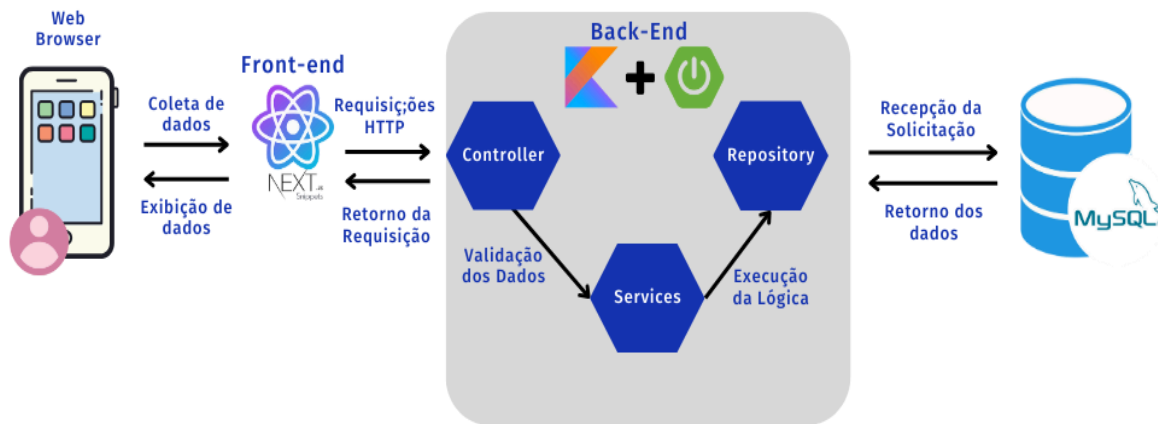


Figura 5 - Visão Geral do Sistema

O sistema “Lembre Bem” se divide em 3 partes principais:

- Interface da aplicação (Front-end) Desenvolvimento utilizando Next.js e TypeScript e biblioteca Chakra UI para componentes de interface. Oferece ao usuário uma interface onde ele conseguirá gerenciar dependentes, lembretes e tratamentos, interagindo diretamente com o back-end por meio de requisições HTTP.
- API (Back-End): Implementada em Kotlin com o framework Spring Boot. A API segue o padrão arquitetural RESTful, que faz a utilização de métodos HTTP e a manipulação de recursos identificados por URLs e a representação de dados em formatos JSON ou XML. Por meio dessa estrutura, se consegue expor endpoints que permitem a interação com os dados e a lógica de negócio do sistema.
- Banco de Dados: Utiliza o sistema de gerenciamento de banco de dados relacional MySQL, onde são armazenadas as informações relacionadas aos lembretes, tratamentos, dependentes e outras entidades complementares do sistema. O banco de dados é acessado pelo back-end através do ORM Spring Data JPA, que facilita o mapeamento entre as entidades Java e as tabelas do banco.

Conforme ilustrado na Figura 3, o fluxo de dados no sistema! inicia-se na interface da aplicação, onde o usuário realiza uma ação, como cadastrar um novo lembrete. Essa ação gera uma requisição HTTP que é enviada ao back-end. O *Controller* no back-end recebe a requisição e delega as responsabilidades de validação e processamento ao *Service*. O *Service* então interage com o *Repository*, que realiza as operações necessárias no banco de dados. Após a execução das operações, os dados são retornados, passando novamente pelo *Service* e *Controller*, que enviam a resposta ao front-end. Por fim, a interface exibe os resultados para o usuário, completando o ciclo de interação.

3.3 Modelo de Banco de Dados

O sistema "Lembre Bem" adota um modelo de banco de dados relacional para armazenar e gerenciar as informações dos dependentes, lembretes, tratamentos e outras entidades relacionadas. Este modelo relacional foi implementado utilizando o Sistema de Gerenciamento de Banco de Dados MySQL.

3.3.1 Modelo de Dados Lógico

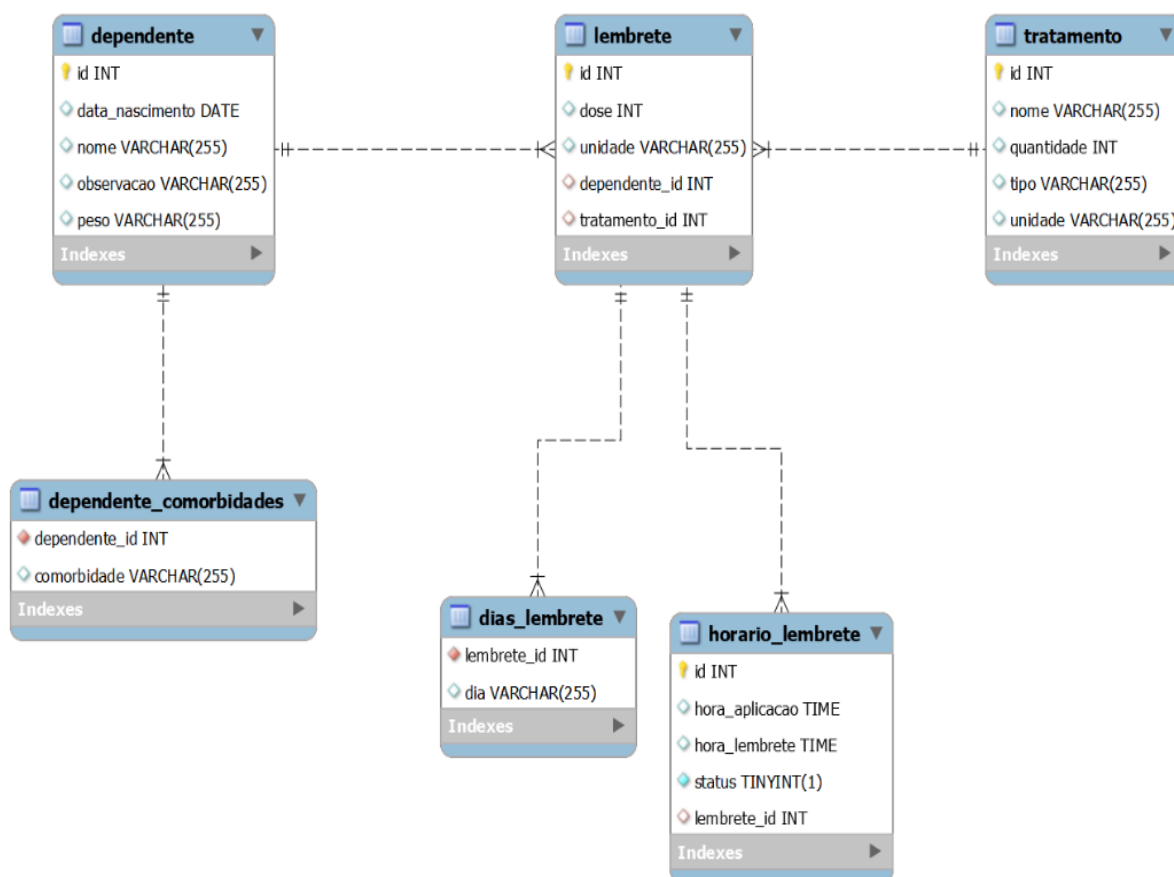


Figura 6 - Diagrama entidade-relacionamento

O modelo de dados lógico é representado pela Figura 4. As entidades principais no sistema "Lembre Bem" incluem "Dependente", "Lembrete", "Tratamento", "DiasLembrete", "HorarioLembrete", e "DependenteComorbidades".

As entidades são relacionadas entre si de forma que um dependente pode ter múltiplos lembretes, que, por sua vez, estão associados a um único tratamento. As entidades “DiasLembrete” e “HorarioLembrete” detalham os dias da semana e os horários específicos para cada lembrete. Além disso, a entidade “DependenteComorbidades” armazena as comorbidades associadas a cada dependente.

3.3.2 Modelo de Dados Físico

O modelo de dados físico mapeia cada entidade do modelo lógico para uma tabela no banco de dados. As principais tabelas e seus atributos são:

- **Dependente:** Armazena informações sobre os dependentes, como id, nome, data_nascimento, peso, observacao.
- **Lembrete:** Contém os lembretes associados aos dependentes, incluindo id, dose, unidade, dependente_id, tratamento_id.
- **Tratamento:** Armazena os tratamentos disponíveis, incluindo id, nome, quantidade, tipo, unidade.
- **DiasLembrete:** Registra os dias da semana em que cada lembrete deve ser executado, com atributos lembrete_id e dia.
- **HorarioLembrete:** Armazena os horários específicos para os lembretes, incluindo id, hora_aplicacao, hora_lembrete, status, lembrete_id.
- **DependenteComorbidades:** Armazena as comorbidades associadas a cada dependente, incluindo dependente_id e comorbidade.

4. Testes de Software

4.1 Projeto de Testes

Os testes de software adotados para o desenvolvimento do sistema foram realizados tanto de forma manual quanto automatizada, para avaliar as funcionalidades, usabilidades e desempenho do sistema.

- **Testes Funcionais:** Verifica se o software atende às especificações e requisitos funcionais. São voltados para testar se as funcionalidades implementadas estão operando conforme o esperado e que todas as funcionalidades essenciais estão disponíveis ao usuário.
- **Testes de Interface de Usuário (UI):** Validam a usabilidade e a experiência do usuário, verificando se a interface é intuitiva e se os elementos visuais estão dispostos de maneira clara e acessível. Esses testes foram conduzidos manualmente, por pessoas que utilizaram o sistema como usuário final.
- **Testes de API:** Validam a integração e a funcionalidade das interfaces de programação (APIs), garantindo que as comunicações entre o front-end e o back-end ocorrem de forma correta.
- **Testes de Performance:** Medem o desempenho do sistema em diferentes condições de carga, assegurando que o sistema é capaz de lidar eficientemente com múltiplos usuários e processos simultâneos.

Abaixo são as tabelas contendo as descrições dos principais cenários de testes da aplicação:

Caso de Teste 1

Campo	Descrição
Identificador	LEMBREBEM001
Caso de Uso	Cadastrar Tratamento
Requisito	Cadastrar Tratamento

Pré-condição de Teste	Estar dentro do sistema.
Entradas	Informar nome, unidade, dose e dias do tratamento.
Saídas Esperadas	O sistema deve criar um novo tratamento e exibir uma mensagem de sucesso da ação.
Procedimento de Teste	<ol style="list-style-type: none"> 1. O usuário deve acessar a tela de gerenciamento de tratamentos; 2. O usuário deve acessar a tela de criação de tratamento através do botão “Cadastrar Tratamento”; 3. Preencher os campos obrigatórios; 4. O sistema deve confirmar a criação com uma mensagem de sucesso;
Observações	Verificar se o tratamento foi salvo no banco de dados.

Caso de Teste 2

Campo	Descrição
Identificador	LEMBREBEM002
Caso de Uso	Editar Tratamento Existente
Requisito	Editar Tratamento
Pré-condição de Teste	Estar dentro do sistema e existir ao menos um tratamento cadastrado.
Entradas	Selecionar um tratamento e alterar os dados desejados.
Saídas Esperadas	O sistema deve salvar as alterações e exibir uma mensagem de sucesso.
Procedimento de Teste	<ol style="list-style-type: none"> 1. O usuário deve acessar a tela de gerenciamento de tratamentos; 2. O usuário deve acessar a tela de edição de tratamento através do botão de “Editar”; 3. O usuário deve alterar os dados desejados; 4. O sistema deve confirmar a edição com uma mensagem de sucesso;

Observações	Verificar se as alterações foram salvas corretamente.
--------------------	---

Caso de Teste 3

Campo	Descrição
Identificador	LEMBREBEM003
Caso de Uso	Excluir Tratamento Existente
Requisito	Remover Tratamento
Pré-condição de Teste	Estar dentro do sistema e existir ao menos um tratamento cadastrado.
Entradas	Selecionar um tratamento e confirmar a exclusão.
Saídas Esperadas	O sistema deve excluir o tratamento, exibir uma mensagem de sucesso e removê-lo da lista de tratamentos disponíveis.
Procedimento de Teste	<ol style="list-style-type: none"> 1. O usuário deve acessar a tela de gerenciamento de tratamentos. 2. O usuário deve selecionar um tratamento para exclusão. 3. O sistema deve exibir uma mensagem de confirmação. 4. O usuário deve confirmar a exclusão do tratamento. 5. O sistema deve exibir uma mensagem de sucesso e o tratamento deve ser removido da lista de tratamentos disponíveis.
Observações	Verificar se o tratamento foi realmente removido do banco de dados.

Caso de Teste 4

Campo	Descrição
Identificador	LEMBREBEM004

Caso de Uso	Cadastrar Dependente
Requisito	Cadastrar Dependente
Pré-condição de Teste	Estar dentro do sistema.
Entradas	Informar nome, data de nascimento e informações adicionais do dependente.
Saídas Esperadas	O sistema deve criar um novo dependente e exibir uma mensagem de sucesso.
Procedimento de Teste	<ol style="list-style-type: none"> 1. O usuário deve acessar a tela de gerenciamento de dependentes; 2. O usuário deve acessar a tela de criação de dependente através do botão de “cadastrar Dependente”; 3. O usuário deve preencher os campos obrigatórios; 4. O sistema deve confirmar a criação com uma mensagem de sucesso;
Observações	Verificar se o dependente foi salvo no banco de dados.

Caso de Teste 5

Campo	Descrição
Identificador	LEMBREBEM005
Caso de Uso	Editar Dependente Existente
Requisito	Editar Dependente
Pré-condição de Teste	Estar dentro do sistema e existir ao menos um Dependente cadastrado.
Entradas	Selecionar um dependente e alterar os dados desejados.
Saídas Esperadas	O sistema deve salvar as alterações e exibir uma mensagem de sucesso.

Procedimento de Teste	<ol style="list-style-type: none"> 1. O usuário deve acessar a tela de gerenciamento de dependentes; 2. O usuário deve acessar a tela de edição de Dependente através do botão de “Editar”; 3. O usuário deve alterar os dados desejados; 4. O sistema deve confirmar a edição com uma mensagem de sucesso;
Observações	Verificar se as alterações foram salvas corretamente.

Caso de Teste 6

Campo	Descrição
Identificador	LEMBREBEM006
Caso de Uso	Excluir Dependente Existente
Requisito	Remover Dependente
Pré-condição de Teste	Estar dentro do sistema e existir ao menos um Dependente cadastrado.
Entradas	Selecionar um dependente e confirmar a exclusão.
Saídas Esperadas	O sistema deve excluir o dependente, exibir uma mensagem de sucesso e removê-lo da lista de tratamentos disponíveis.
Procedimento de Teste	<ol style="list-style-type: none"> 1. O usuário deve acessar a tela de gerenciamento de dependentes; 2. O usuário deve selecionar um tratamento para exclusão; 3. O sistema deve exibir uma mensagem de confirmação; 4. O usuário deve confirmar a exclusão do dependente; 5. O sistema deve exibir uma mensagem de sucesso e o dependente deve ser removido da lista de dependentes disponíveis;

Observações	Verificar se o dependente foi realmente removido do banco de dados.
--------------------	---

Caso de Teste 7

Campo	Descrição
Identificador	LEMBREBEM007
Caso de Uso	Cadastrar Lembrete
Requisito	Cadastrar Lembrete
Pré-condição de Teste	Estar dentro do sistema, existência de Lembrete e Dependente.
Entradas	Informar dependente, tratamento, horários, dias, dose e unidade.
Saídas Esperadas	O sistema deve criar um novo lembrete e exibir uma mensagem de sucesso.
Procedimento de Teste	<ol style="list-style-type: none"> 1. O usuário deve acessar a tela de gerenciamento de Lembretes; 2. O usuário deve acessar a tela de criação de Lembrete através do botão de “Criar Lembrete”; 3. O usuário deve preencher os campos obrigatórios; 4. O sistema deve confirmar a criação com uma mensagem de sucesso;
Observações	Verificar se o lembrete foi salvo no banco de dados.

Caso de Teste 8

Campo	Descrição
Identificador	LEMBREBEM008
Caso de Uso	Editar Lembrete Existente

Requisito	Editar Dependente
Pré-condição de Teste	Estar dentro do sistema e existir ao menos um Dependente cadastrado.
Entradas	Selecionar um lembrete e alterar os dados desejados.
Saídas Esperadas	O sistema deve salvar as alterações e exibir uma mensagem de sucesso.
Procedimento de Teste	<ol style="list-style-type: none"> 1. O usuário deve acessar a tela de gerenciamento de lembretes; 2. O usuário deve acessar a tela de edição de lembrete através do botão de "Editar"; 3. O usuário deve alterar os dados desejados; 4. O sistema deve confirmar a edição com uma mensagem de sucesso;
Observações	Verificar se as alterações foram salvas corretamente.

Caso de Teste 9

Campo	Descrição
Identificador	LEMBREBEM009
Caso de Uso	Excluir Lembrete Existente
Requisito	Remover Lembrete
Pré-condição de Teste	Estar dentro do sistema e existir ao menos um lembrete cadastrado.
Entradas	Selecionar um lembrete e confirmar a exclusão.
Saídas Esperadas	O sistema deve marcar o lembrete como "Confirmado" e exibir uma mensagem de confirmação para o usuário.

Procedimento de Teste	<ol style="list-style-type: none"> 1. O usuário deve acessar a tela de gerenciamento de lembretes; 2. O usuário deve selecionar um lembrete para exclusão; 3. O sistema deve exibir uma mensagem de confirmação; 4. O usuário deve confirmar a exclusão do lembrete; 5. O sistema deve exibir uma mensagem de sucesso e o lembrete deve ser removido da lista de dependentes disponíveis;
Observações	Verificar se o dependente foi realmente removido do banco de dados.

Caso de Teste 10

Campo	Descrição
Identificador	LEMBREBEM010
Caso de Uso	Confirmar Lembrete
Requisito	Confirmar Lembrete
Pré-condição de Teste	Estar dentro do sistema e existir ao menos um lembrete cadastrado.
Entradas	Selecionar um lembrete disponível na lista de lembretes
Saídas Esperadas	O sistema deve excluir o lembrete, exibir uma mensagem de sucesso e removê-lo da lista de tratamentos disponíveis.

Procedimento de Teste	<ol style="list-style-type: none">1. O usuário deve acessar a tela de exibição dos lembretes do dia;2. O usuário deve selecionar um lembrete disponível na lista;3. O sistema deve exibir a opção de confirmar o lembrete;4. O usuário deve clicar no botão de confirmação do lembrete;5. O sistema deve marcar o lembrete como "Confirmado" e exibir uma mensagem de sucesso;
Observações	O sistema deve garantir que, após a confirmação, o lembrete não seja mais exibido como pendente.

5. Implantação

5.1 Projeto de Implantação

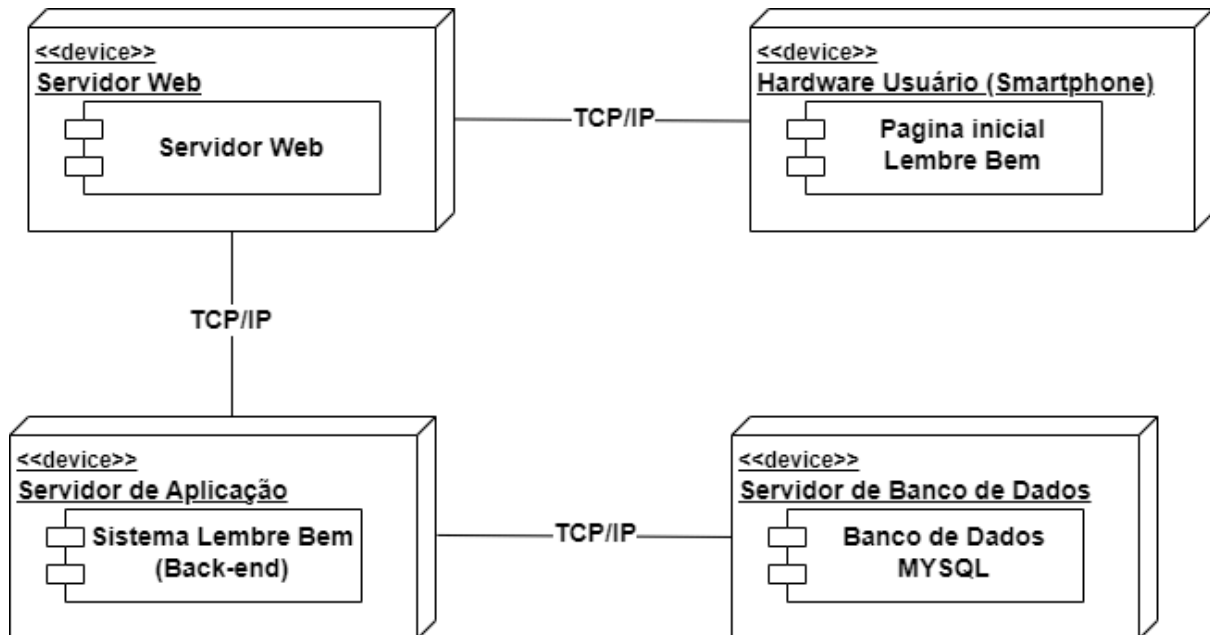


Figura 7 - Diagrama de Implantação

A aplicação "Lembre Bem" foi desenvolvida para ser acessada por meio de navegadores web (browsers) em diversos dispositivos, como computadores, tablets e smartphones. O design responsivo da aplicação garante que ela se adapte de forma otimizada a diferentes tamanhos de tela, proporcionando uma experiência de uso consistente e intuitiva.

A aplicação segue o fluxo apresentado na Figura 5, descrito pelas seguintes etapas:

1. Acesso à Aplicação: O usuário, utilizando um dispositivo com navegador web, como um smartphone, tablet ou computador;
2. Carregamento da interface permitindo que o usuário tenha acesso ao sistema;
3. Interação com o Back-End: Quando o usuário realiza ações como cadastrar ou editar lembretes, uma requisição HTTP é enviada ao backend da aplicação, ele valida os dados recebidos e processa a lógica de negócios correspondente;
4. Após validação é feita a consulta no banco e o retorno para o usuário através da interface;

6. Manual do Usuário



Figura 8 - Tela inicial Lembre Bem

- Na tela Inicial da aplicação onde há um botão "Entrar" que leva o usuário ao menu principal do aplicativo;

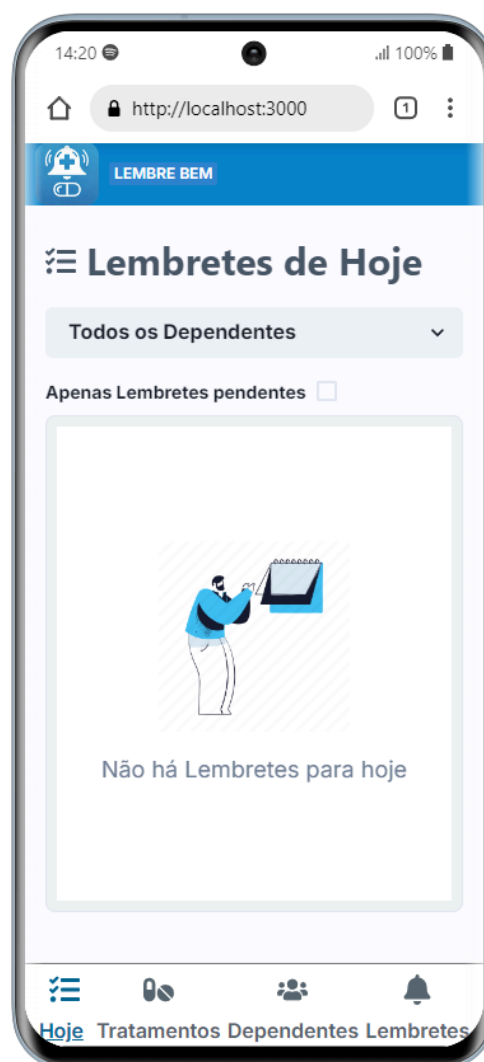


Figura 9 - Tela inicial Lembre Bem

- Na tela são exibidos os lembretes do dia.
- Há a opção de filtrar os lembretes por dependente ou visualizar todos;
- A opção "Apenas Lembretes Pendentes" permite visualizar somente os lembretes que ainda não foram confirmados;
- Se não houver lembretes para o dia, a mensagem "Não há Lembretes para hoje" será exibida;



Figura 10 - Tela Tratamentos

- Na tela exibe a lista de tratamentos cadastrados, acessada pelo menu inferior;
- Caso não existam tratamentos cadastrados, será exibida a mensagem "Não há Tratamentos cadastrados" e uma imagem ilustrativa;
- O botão "Cadastrar Tratamento" na parte inferior permite ao usuário adicionar um novo tratamento;

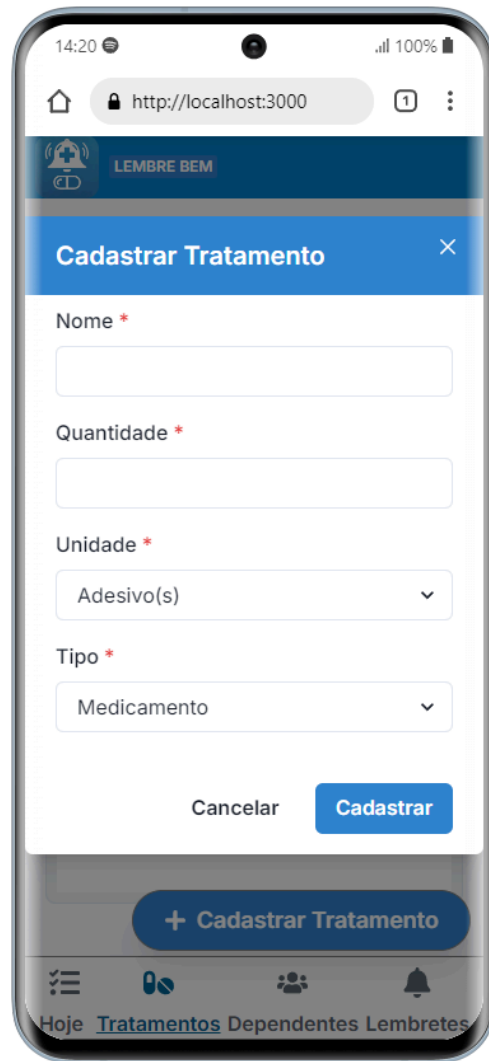


Figura 11 - Cadastro de Tratamento

- O usuário pode acessar esta tela para cadastrar um novo tratamento;
- Um formulário permite a inserção de dados como Nome, Quantidade, Unidade e Tipo (Medicamento ou Atividade);
- Dois botões estão disponíveis: "Cancelar" para descartar as alterações e "Cadastrar" para confirmar e salvar o novo tratamento;

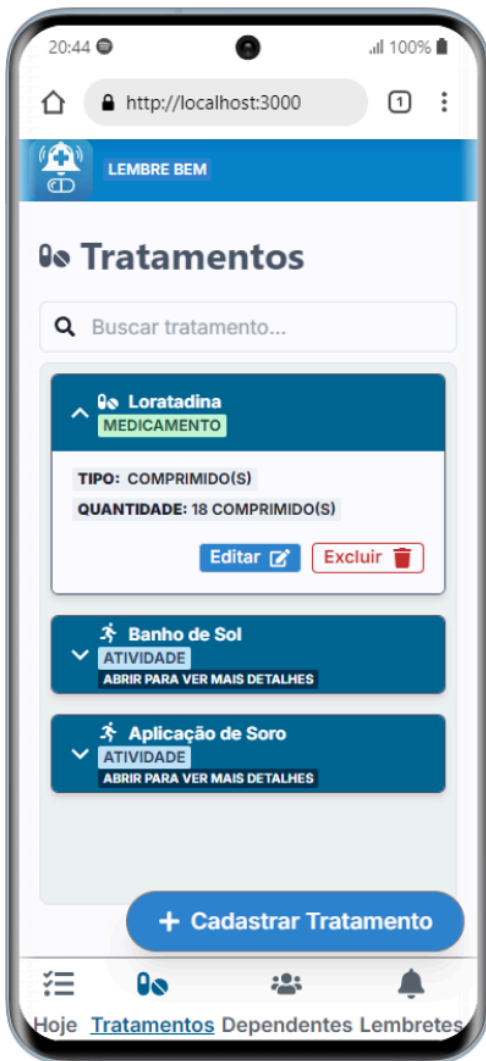


Figura 12 - Tela Tratamentos com dados

- Na tela, os tratamentos cadastrados são exibidos em cartões individuais;
- Cada cartão exibe informações detalhadas sobre o tratamento, incluindo Nome, Tipo, Quantidade e Unidade;
- Há botões de "Editar" e "Excluir" em cada cartão, permitindo a modificação ou exclusão do tratamento respectivo.
- O botão "Cadastrar Tratamento" permanece disponível para adicionar novos tratamentos;



Figura 13 - Tela de Dependentes

- Na Tela "Dependentes", você pode gerenciar os dependentes do sistema;
- Se não houver dependentes cadastrados, a tela exibirá uma mensagem de "Não há Dependentes Cadastrados";
- O botão "Cadastrar Dependentes" está disponível para o cadastro de Dependentes;

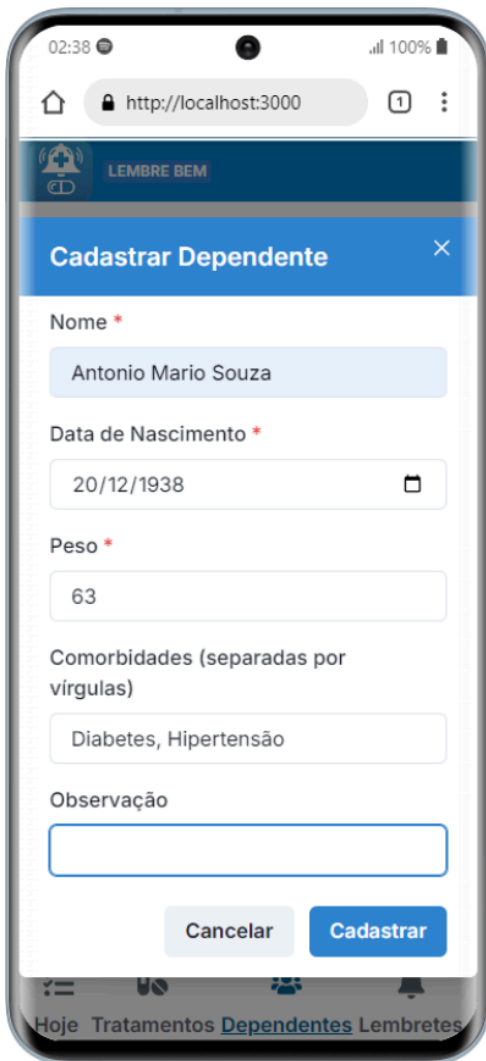


Figura 14 - Cadastro de Dependentes

- Na tela, você pode cadastrar um novo dependente;
- O formulário inclui campos para Nome, Data de Nascimento, Peso, Comorbidades (separadas por vírgulas) e Observações;
- Dois botões estão disponíveis: "Cancelar" para descartar as alterações e "Cadastrar" para confirmar e salvar o novo dependente;



Figura 15 - Tela de Dependentes com dados

- A tela exibe a lista de dependentes cadastrados, cada um em seu cartão individual, mostrando o nome e outras informações do dependente;
- Cada cartão tem botões de "Editar" e "Excluir" para modificar ou remover o dependente;
- O botão "Cadastrar Dependentes" continua disponível para adicionar novos dependentes;



Figura 16 - Tela de Lembretes

- Na tela, você pode cadastrar um novo lembrete;
- O formulário inclui campos para Nome, Data de Nascimento, Peso, Comorbidades (separadas por vírgulas) e Observações;
- Dois botões estão disponíveis: "Cancelar" para descartar as alterações e "Cadastrar" para confirmar e salvar o novo lembrete;

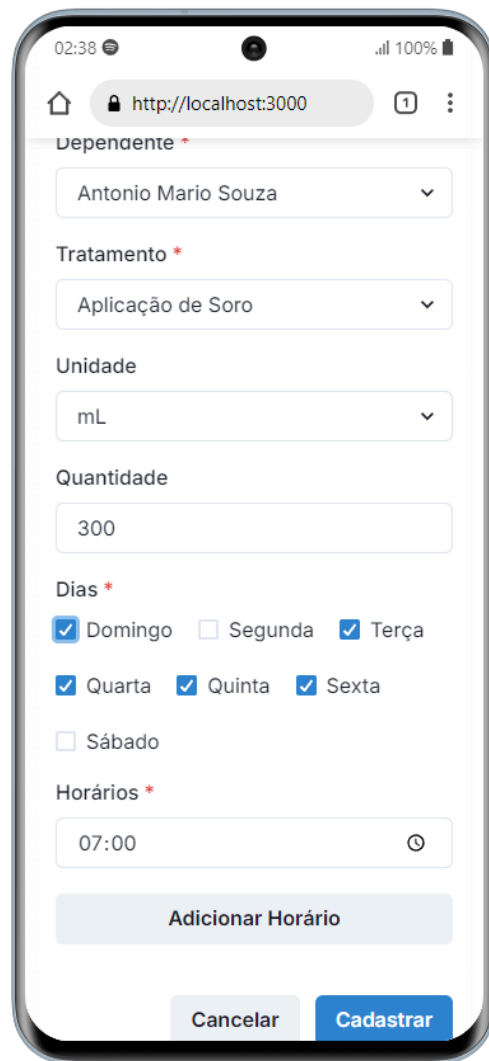


Figura 17 - Cadastro de Lembretes

- Na tela, o usuário pode criar um novo lembrete através de um formulário faz a inserção de informações do dependente, tratamento, unidade, dose, dias da semana e horários;
- Dois botões estão disponíveis: "Cancelar" para descartar as alterações e "Cadastrar" para confirmar e salvar o novo Lembrete;

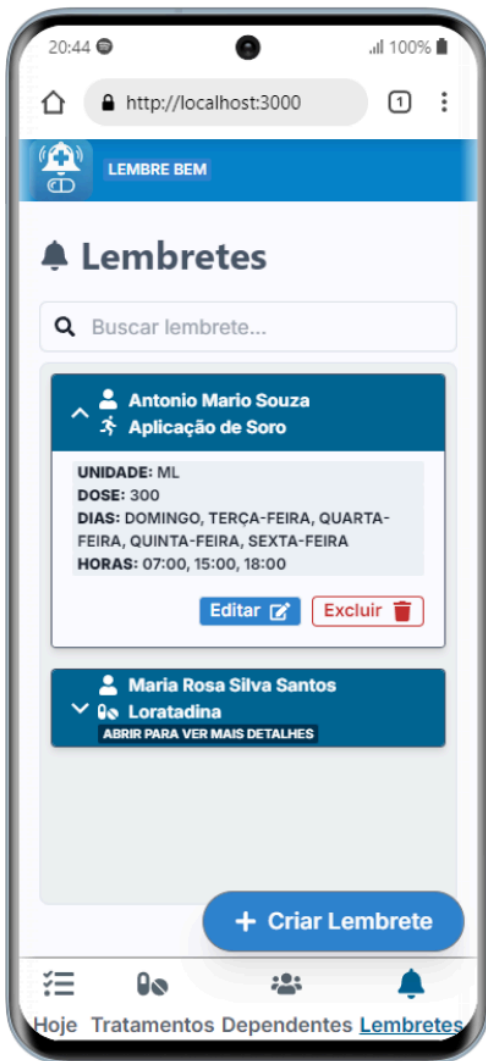


Figura 18 - Tela de Lembretes com dados

- Na tela são exibidos os lembretes cadastrados, organizados em cartões individuais, onde mostram detalhes como nome do dependente, tratamento, horários e status do lembrete;
- Botões de "Editar" e "Excluir" permitem ao usuário modificar ou remover o lembrete;
- O botão "Criar Lembrete" continua disponível para adicionar novos lembretes;



Figura 19 - Tela inicial Lembre Bem com dados

- Os lembretes são exibidos em ordem cronológica, cada um em seu próprio cartão, com detalhes como hora, tratamento, e status (Feito, Próximo, Agora, Atrasado);
- O usuário pode confirmar o lembrete clicando no botão correspondente;
- Selecione um dependente específico ou veja lembretes para todos os dependentes;

Agradecimentos

Primeiramente, gostaria de agradecer a Deus, por ter me dado forças e saúde para conseguir concluir essa etapa e, principalmente, por me permitir realizá-la. Agradeço também a todas as pessoas que estiveram comigo durante todo esse tempo: meus amigos, minha família, minha namorada e meu cachorro. A todas as mensagens preocupadas, todo pedido de que eu devia descansar, Um agradecimento especial ao meu barbeiro, que sempre me motiva a seguir em frente a cada vez que vou ao salão.

Quero deixar aqui minha gratidão a meu amigo Fernando e à sua família por me acolherem. Sem eles, eu realmente não teria conseguido chegar até aqui e nada disso seria possível, inclusive não teria conseguido continuar na faculdade.

Sou grato aos meus professores, que me auxiliaram nesta jornada com boas instruções e conselhos, os que hoje considero como além de professores, amigos.

Agradeço também pelos dias difíceis que superei nessa jornada, foram essenciais para que eu percebesse o quão desafiador é conquistar algo e, assim, aprender a valorizar ainda mais as coisas.

Por fim, agradeço à turma de Radiologia de 2022, à turma de ADS de 2019.1 e 2019.2, e aos amigos incríveis que fiz durante esse tempo aqui no IFBA. Eles me motivaram e motivam, tanto na vida acadêmica quanto fora dela, servindo de espelho para ser uma pessoa melhor.

Obrigado, pessoal!

Referências

[1] "O impacto da doença de Alzheimer no familiar cuidador." Filadélfia. Disponível em:

<https://www.filadelfia.com.br/artigo-academico/o-impacto-da-doenca-de-alzheimer-n-o-familiar-cuidador/>. Acesso em: 22/04/2024

[2] MINISTÉRIO DA SAÚDE. Conhecer a demência, conhecer o Alzheimer: O poder do conhecimento. Setembro, Mês Mundial do Alzheimer. Disponível em: <https://bvsmis.saude.gov.br/conhecer-a-demencia-conhecer-o-alzheimer-o-poder-do-conhecimento-setembro-mes-mundial-do-alzheimer/>. Acesso em: 22/04/2024.

[3] BBC NEWS BRASIL. "'Não sabia o que fazer': os desafios de quem cuida de pessoas com Alzheimer". Disponível em: <https://www.bbc.com/portuguese/geral-43984413>. Acesso em: 22/04/2024.

[4] "Alzheimer: A doença que afeta não só o portador, mas também seus familiares." Vittude. Disponível em: <https://www.vittude.com/blog/alzheimer-afeta-pessoas-proximas-ao-portador/>. Acesso em: 22/04/2024.

[5] "Cuidadores de familiares com doenças crônicas: estratégias de enfrentamento utilizadas no cotidiano." SciELO. Disponível em: <https://www.scielo.br/j/cadbto/a/NBGPMpwyJCKKPPVnmHsSz7P/?format=pdf&lang=pt>. Acesso em: 22/04/2024.

[6] JETBRAINS. Kotlin Programming Language. Disponível em: <https://kotlinlang.org>. Acesso em: 22/07/2024.

[7] SPRING.IO. Spring Boot Framework. Disponível em: <https://spring.io/projects/spring-boot>. Acesso em: 22/07/2024.

[8] NEXT.JS. Next.js Framework Documentation. Disponível em: <https://nextjs.org>. Acesso em: 22/07/2024.

- [9] TYPESCRIPT. TypeScript Documentation. Disponível em: <https://www.typescriptlang.org/>. Acesso em: 22/07/2024.
- [10] CHAKRA UI. Chakra UI Documentation. Disponível em: <https://chakra-ui.com/>. Acesso em: 22/07/2024.
- [11] MYSQL. MySQL Database System. Disponível em: <https://www.mysql.com/>. Acesso em: 22/07/2024.
- [12] GIT-SCM. Git Version Control System. Disponível em: <https://git-scm.com/>. Acesso em: 22/07/2024.
- [13] GITHUB. GitHub Code Hosting Platform. Disponível em: <https://github.com/>. Acesso em: 22/07/2024.
- [14] JEST. Jest Testing Framework. Disponível em: <https://jestjs.io/>. Acesso em: 22/07/2024.
- [15] INSOMNIA. Insomnia API Client. Disponível em: <https://insomnia.rest/>. Acesso em: 22/07/2024.
- [16] REACT. React - A JavaScript library. Disponível em: <https://react.dev/>. Acesso em: 22/07/2024.