FAI – CENTRO DE ENSINO SUPERIOR EM GESTÃO, TECNOLOGIA E EDUCAÇÃO

CURSO DE SISTEMAS DE INFORMAÇÃO

DILTON THALES MELO DA SILVA

LUCAS DOS REIS SEVERINI

MATEUS BOCHE DANIEL

DONATE: SISTEMA DE INFORMAÇÃO PARA DOAÇÃO DE LEITE MATERNO

SANTA RITA DO SAPUCAÍ - MG

2025

FAI – CENTRO DE ENSINO SUPERIOR EM GESTÃO, TECNOLOGIA E EDUCAÇÃO

CURSO DE SISTEMAS DE INFORMAÇÃO

DILTON THALES MELO DA SILVA

LUCAS DOS REIS SEVERINI

MATEUS BOCHE DANIEL

DONATE: SISTEMA DE INFORMAÇÃO PARA DOAÇÃO DE LEITE MATERNO

Projeto de final de curso apresentado a FAI – Centro de Ensino Superior em Gestão, Tecnologia e Educação, como requisito parcial para obtenção do Título de Bacharel em Sistemas de Informação, sob a orientação da profa. Eunice Gomes de Siqueira.

SANTA RITA DO SAPUCAÍ - MG

2025

FOLHA DE APROVAÇÃO

Na impressão final do documento, esta folha será substituída pela assinada pela Banca examinadora final.

HISTÓRICO DE REVISÕES

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versão | Autor (es) | Descrição |
| 25/10/2025 | 2.3 | Dilton Thales | Implementação da Fase 4 – Conclusão do Plano de Implantação, inclusão da Análise de Complexidade Algorítmica (Seção 7.3), Métodos Gerenciais e correções dos apontamentos realizados nas fases anteriores. |
| 26/09/25 | 2.2 | Dilton Silva, Lucas Severini e Mateus Boche. | Correções dos apontamentos realizados na entrega da Fase 3. |
| 17/09/25 | 2.1 | Eunice Gomes. | Correções parciais da Fase 3. |
| 01/09/2025 | 2.0 | Lucas Severini. | Adição da Seção 6.5 Sistemas distribuídos e seus implementos. |
| 20/06/25 | 1.9 | Banca de qualificação. | Apontamentos da banca de qualificação. |
| 08/06/25 | 1.7 | Dilton Silva, Lucas Severini e Mateus Boche. | Correções dos apontamentos realizados na entrega anterior. |
| 28/05/25 | 1.6 | Eunice Gomes. | Apontamentos de melhorias para as Fases 1 e 2. |
| 24/05/25 | 1.5 | Dilton Silva, Lucas Severini e Mateus Boche. | Correções dos apontamentos realizados na entrega anterior e na apresentação oral. |
| 16/05/25 | 1.4 | Eunice Gomes. | Apontamentos de melhorias para as Fases 1 e 2. |
| 13/05/25 | 1.2 | Dilton Silva, Lucas Severini e Mateus Boche. | Entrega da Fase 2. |
| 03/04/25 | 1.1 | Eunice Gomes. | Apontamentos parciais para melhorias da Fase 1. |
| 29/03/25 | 1.0 | Dilton Silva, Lucas Severini e Mateus Boche. | Entrega da Fase 1. |

AGRADECIMENTOS

Primeiramente, agradecemos a Deus por nos conceder força, sabedoria e perseverança para concluir este trabalho. À nossas famílias, em especial aos nossos pais, pelo amor incondicional, apoio e compreensão em todos os momentos, especialmente durante as noites em claro dedicadas a este projeto. Vocês são a nossa inspiração! À nossa orientadora, profa. Eunice Gomes de Siqueira, pela paciência, dedicação e orientação precisa, que foram essenciais para a construção deste projeto de final de curso. Sua experiência e conhecimento me guiaram em cada etapa. Agradecemos também aos amigos, que nos incentivaram e ajudaram a manter o foco, compartilhando momentos de estudo e descontração. E, finalmente, a todos que direta ou indiretamente contribuíram para a realização deste trabalho, nosso muito obrigado!

RESUMO

A doação de leite materno é uma prática vital para garantir a nutrição e a sobrevivência de recém-nascidos prematuros e de baixo peso, especialmente quando não é possível a amamentação direta. Contudo, o processo ainda enfrenta desafios como a escassez de informação, desconhecimento dos benefícios e dificuldades de acesso aos postos de coleta. Diante desse cenário, este trabalho apresenta o projeto Donate, que consiste em um sistema de informação composto por três componente, sendo eles um aplicativo *Web*, um aplicativo móvel e uma *Aplication Programming Interface* de integração, cujo objetivo é ser um meio de comunicação entre doadoras, receptores e instituições que mantêm bancos de leite humano. O sistema permitirá o cadastro de usuários, a localização dos postos de coleta mais próximos, o acesso a orientações sobre armazenamento seguro do leite humano e a divulgação de campanhas educativas. Além disso, o sistema oferecerá um painel administrativo voltado às instituições de saúde, permitindo o acompanhamento das doações e divulgação de orientações ao público. A metodologia adotada neste trabalho seguiu uma abordagem aplicada e qualitativa, baseada no modelo Incremental, com foco na entrega contínua de artefatos e relatórios. Além disso, o projeto foi orientado pelas boas práticas da gestão de projetos e a construção do sistema foi dividida em fases curtas, permitindo ajustes rápidos a partir do *feedback* das partes interessadas. Esse processo contribuiu para que o sistema Donate estivesse mais alinhado às reais necessidades dos envolvidos, garantindo maior eficiência, usabilidade e confiabilidade na solução desenvolvida. Espera-se que este projeto amplie o alcance das ações de doação, como também promova a educação em saúde e contribua diretamente para a redução da mortalidade infantil, demonstrando como soluções tecnológicas podem fortalecer as políticas públicas de incentivo ao aleitamento materno e à solidariedade.

**Palavras-chave:** leite materno; bancos de leite humano; tecnologia em saúde; projeto final de curso; Sistemas de Informação.

LISTA DE FIGURAS

[FIGURA 1 – Aplicativo Doe Leite 21](#_Toc209020878)

[FIGURA 2 – Estrutura de pastas do projeto 31](#_Toc209020879)

[FIGURA 3 – Subdivisão das pastas da documentação do projeto. 32](#_Toc209020880)

[FIGURA 4 – Subdivisão das pastas de codificação do projeto. 32](#_Toc209020881)

[FIGURA 5 – Repositório do Github 32](#_Toc209020882)

[FIGURA 6 – Um diagrama de sistema distribuído do Donate. 62](#_Toc209020883)

LISTA DE QUADROS

[QUADRO 1 – Incrementos previstos 29](#_Toc209020806)

[QUADRO 1 – Documentos relevantes para testes 67](#_Toc209020807)

[QUADRO 2 – Equipamentos para a realização dos testes 67](#_Toc209020808)

[QUADRO 4 – Softwares para a realização dos testes 68](#_Toc209020809)

LISTA DE ABREVIATURAS E SIGLAS

API – Application Programming Interface

ARM – Advanced RISC Machines

BLH – Banco de Leite Humano

CD – Continuous Delivery

CI – Continuous Integration

FAI – FAI - Centro de Ensino Superior em Gestão, Tecnologia e Educação

FAQ – Perguntas Frequentes

HCSL – Hospital das Clínicas Samuel Libânio

HTTP – Hyper Text Transfer Protocol

HTTPS – Hyper Text Transfer Protocol Secure

IBGE – Instituto Brasileiro de Geografia e Estatística

IHC – Interação Humano-Computador

JWT – JSON Web Token

MVC – Model-View-Controller

ODS – Objetivos de Desenvolvimento Sustentável

OMS – Organização Mundial da Saúde

ONU – Optical Network Unit

RBLH-BR – Rede de Bancos de Leite Humano - Brasil

RF – Requisito Funcional

RISC – Reduced Instruction Set Computer

RN – Regras de Negocio

RNF – Requisito Não Funcional

RPO – Recovery Point Objective

RTO – Recovery Time Objective

SDK – Software Development Kit

TAP – Termo de Abertura de Projeto

UCIN – Unidade de Cuidados Intermediários Neonatais

UML – Unified Modeling Language

UTI – Unidade de Terapia Intensiva

SUMÁRIO

[1 INTRODUÇÃO 15](#_Toc209020949)

[2 REVISÃO BIBLIOGRÁFICA 17](#_Toc209020950)

[2.1 O ALEITAMENTO MATERNO 17](#_Toc209020951)

[2.2 BANCOS DE LEITE HUMANO 18](#_Toc209020952)

[2.2.1 Tipos de Incentivos Oficiais à Doação de Leite Materno 19](#_Toc209020953)

[2.3 ATUAÇÃO DO POSTO DE COLETA DE LEITE HUMANO NO HOSPITAL DAS CLÍNICAS SAMUEL LIBÂNIO 19](#_Toc209020954)

[2.4 TRABALHOS RELACIONADOS 20](#_Toc209020955)

[3 OBJETIVO DO PROJETO 23](#_Toc209020956)

[3.1 FORMULAÇÃO DO PROBLEMA 23](#_Toc209020957)

[3.2 OBJETIVOS 24](#_Toc209020958)

[3.3 JUSTIFICATIVA 25](#_Toc209020959)

[3.4 NÍVEIS DE DECISÃO E GRUPOS FUNCIONAIS ATENDIDOS 25](#_Toc209020960)

[3.5 ADERÊNCIA AOS OBJETIVOS DE DESENVOLVIMENTO SUSTENTÁVEL (ODS) 26](#_Toc209020961)

[3.6 MODELO DE NEGÓCIO 26](#_Toc209020962)

[4 MÉTODOS GERENCIAIS 27](#_Toc209020963)

[4.1 GERENCIAMENTO DO PROJETO 27](#_Toc209020964)

[4.2 MODELO DE CICLO DE VIDA 27](#_Toc209020965)

[4.3 RECURSOS NECESSÁRIOS 29](#_Toc209020966)

[4.4 RELATÓRIO DE DESEMPENHO 30](#_Toc209020967)

[5 ESPECIFICAÇÃO E ANÁLISE DOS REQUISITOS 33](#_Toc209020968)

[5.1 REQUISITOS DO SISTEMA DE SOFTWARE 33](#_Toc209020969)

[5.1.1 Requisitos Funcionais 34](#_Toc209020970)

[5.1.2 Requisitos Não Funcionais 39](#_Toc209020971)

[5.1.3 Principais Regras de Negócio 44](#_Toc209020972)

[5.2 ANÁLISE DOS REQUISITOS 45](#_Toc209020973)

[5.2.1 Visão Funcional 45](#_Toc209020974)

[5.2.2 Modelo Conceitual dos Dados 47](#_Toc209020975)

[5.2.3 Modelo Inicial da Interface de Usuário 47](#_Toc209020976)

[6 ARQUITETURA E PROJETO DO SISTEMA DE SOFTWARE 48](#_Toc209020977)

[6.1 VISÃO ESTRUTURAL 48](#_Toc209020978)

[6.1.1 Diagrama de Pacotes 49](#_Toc209020979)

[6.1.2 Diagramas de Classes 50](#_Toc209020980)

[6.2 VISÃO COMPORTAMENTAL 51](#_Toc209020981)

[6.2.1 Projeto das Interações entre Objetos 52](#_Toc209020982)

[6.3 VISÃO DOS DADOS 53](#_Toc209020983)

[6.3.1 Modelo Lógico 54](#_Toc209020984)

[6.3.2 Dicionário de Dados do Modelo Lógico 54](#_Toc209020985)

[6.4 PROJETO DA INTERAÇÃO HUMANO-COMPUTADOR 55](#_Toc209020986)

[6.4.1 Perfil de Usuário 55](#_Toc209020987)

[6.4.2 Projeto da Interface de Usuário 56](#_Toc209020988)

[6.4.3 Heurísticas de Usabilidade 56](#_Toc209020989)

[6.4.4 Projeto da Acessibilidade 57](#_Toc209020990)

[6.5 PROJETO DO SISTEMA DISTRIBUÍDO 58](#_Toc209020991)

[6.5.1 Procedimentos para Tratamentos dos Desafios 58](#_Toc209020992)

[7 IMPLEMENTAÇÃO DO SISTEMA DE SOFTWARE 63](#_Toc209020993)

[7.1 COMPONETES DO SISTEMA DE SOFTWARE 63](#_Toc209020994)

[7.2 TECNOLOGIAS DE IMPLEMENTAÇÃO 64](#_Toc209020995)

[7.2.1 Linguagens de Programação e *Frameworks* Adotados 64](#_Toc209020996)

[7.2.2 *Design Patterns* Aplicados 64](#_Toc209020997)

[8 PLANO DE TESTES 66](#_Toc209020998)

[8.1 FINALIDADE 66](#_Toc209020999)

[8.2.1 Referências aos Documentos Relevantes 67](#_Toc209021000)

[8.2.2 Ambiente para a Realização dos Testes 67](#_Toc209021001)

[8.3 ESPECIFICAÇÃO DOS CASOS DE TESTES 68](#_Toc209021002)

[8.4 RESULTADOS DOS TESTES 69](#_Toc209021003)

[9 CONCLUSÃO 70](#_Toc209021004)

[REFERÊNCIAS 72](#_Toc209021005)

[APÊNDICE A – GERENCIAMENTO DO PROJETO 76](#_Toc209021006)

[APÊNDICE B – RELATÓRIO DE DESEMPENHO 77](#_Toc209021007)

[APÊNDICE C – VISÃO FUNCIONAL 78](#_Toc209021008)

[APÊNDICE D – VISÃO DOS DADOS 79](#_Toc209021009)

[APÊNDICE E – MODELO INICIAL DA INTERFACE DE USUÁRIO 80](#_Toc209021010)

[APÊNDICE F – VISÕES ESTRUTURAL E COMPORTAMENTAL 81](#_Toc209021011)

[APÊNDICE G – ENTREVISTAS COM USUÁRIOS 82](#_Toc209021012)

# 1 INTRODUÇÃO

O leite materno é o alimento padrão ouro para lactentes, sendo o alimento essencial para crianças prematuras, de baixo peso ou hospitalizadas que não podem ser amamentadas por suas mães. No entanto, a captação desse leite ainda enfrenta desafios significativos, como a falta de informação, dificuldades no acesso aos pontos de coleta e o desconhecimento sobre os benefícios da doação. Muitas mulheres que poderiam doar não sabem onde ou como realizar esse processo, e os bancos de leite humano, essenciais para a redistribuição desse alimento, frequentemente operam com estoques abaixo do necessário. Segundo a Rede Brasileira de Bancos de Leite Humano (rBLH-BR), uma maior conscientização e acessibilidade ao processo de doação poderia aumentar significativamente a quantidade de leite disponível, salvando mais vidas neonatais (RBLH-BR, 2023).

Diante desse cenário, o projeto Donate visa o desenvolvimento de um sistema de informação de forma a utilizar as tecnologias de informação e comunicação para conectar doadoras e bancos de leite humano. A projeto consiste de um sistema de software intuitivo e acessível, composto por um aplicativo móvel e outros componentes, que permitirá às pessoas localizar postos de coleta, obter informações sobre o armazenamento correto do leite e conhecer campanhas de incentivo à doação. Além disso, o sistema também beneficiará os bancos de leite humano ao facilitar o gerenciamento das coletas e ampliar o alcance das campanhas de conscientização. Com isso, espera-se fortalecer a rede de doação e garantir que um maior número de crianças tenha acesso a esse alimento vital.

O Donate além vir a oferecer de um canal de comunicação entre doadoras e instituições, também visa promover a educação sobre a importância do leite materno, seus benefícios nutricionais e o impacto positivo que a doação pode gerar. Segundo a Organização Mundial da Saúde (OMS, 2023), o leite materno é o alimento ideal para as crianças, reduzindo o risco de doenças e fortalecendo o sistema imunológico nos primeiros meses de vida. Dessa forma, este sistema de informação é uma iniciativa que busca incentivar mais mulheres a se tornarem doadoras, fornecendo informações de fontes confiáveis e tornando o processo mais acessível e prático.

Este trabalho está dividido em 10 capítulos.

Capítulo 1 – Introdução

Apresenta o contexto do problema: a importância do leite materno, os desafios da doação no Brasil e a motivação para o desenvolvimento do sistema Donate. Expõe os objetivos do projeto e a relevância social da proposta.

Capítulo 2 – Revisão Bibliográfica

Reúne conceitos e estudos que fundamentam o trabalho: benefícios do aleitamento materno, funcionamento dos Bancos de Leite Humano, políticas de incentivo à doação e experiências relacionadas ao uso de tecnologias digitais aplicadas à saúde.

Capítulo 3 – Objetivo do Projeto

Define o problema central, os objetivos gerais e específicos, a justificativa do projeto, os grupos funcionais atendidos, a aderência aos Objetivos de Desenvolvimento Sustentável (ODS) e o modelo de negócio proposto.

Capítulo 4 – Métodos Gerenciais

Apresenta os métodos e modelos de gerenciamento de projetos utilizados, incluindo o Termo de Abertura, ciclo de vida (Modelo Incremental), recursos necessários, relatórios de desempenho e o gerenciamento de configuração.

Capítulo 5 – Especificação e Análise dos Requisitos

Detalha os requisitos funcionais, não funcionais e as principais regras de negócio do sistema. Também descreve a análise dos requisitos por meio de visão funcional, modelo conceitual de dados e protótipos de interface.

Capítulo 6 – Arquitetura e Projeto do Sistema de Software

Expõe a arquitetura do sistema, incluindo diagramas estruturais, comportamentais e de dados. Também aborda a interação humano-computador, o perfil de usuários, o design da interface e os procedimentos do sistema distribuído.

Capítulo 7 – Implementação do Sistema de Software

Mostra os componentes implementados, as tecnologias empregadas (linguagens, frameworks, padrões de projeto) e como as soluções foram aplicadas na prática.

Capítulo 8 – Plano de Testes

Descreve os procedimentos de teste do sistema: documentos de referência, ambiente utilizado, casos de teste planejados e os resultados obtidos.

Capítulo 9 – Conclusão

Apresenta as considerações finais, destacando as contribuições do projeto, os resultados alcançados e possíveis perspectivas de evolução do sistema.

# 2 REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta os conceitos e estudos que fundamentam o desenvolvimento deste projeto. Inicialmente, são discutidos os benefícios e a importância do aleitamento materno para a saúde infantil e materna. Em seguida, é abordado sobre o funcionamento dos bancos de leite humano, com foco na sua estrutura, atuação e impacto na saúde pública. A terceira seção trata da atuação do Posto de Coleta de Leite Humano do Hospital das Clínicas Samuel Libânio, localizado em Pouso Alegre/MG e sua relevância regional. Por fim, são apresentados trabalhos relacionados, com ênfase no uso de tecnologias digitais aplicadas ao incentivo da amamentação e à doação de leite humano.

## 2.1 O ALEITAMENTO MATERNO

O aleitamento materno é uma das formas mais eficazes de garantir a saúde e a sobrevivência dos recém-nascidos. Ele fornece todos os nutrientes necessários para o recém-nascido e à criança nos primeiros meses de vida e ainda protege contra doenças como diarreia, infecções respiratórias, alergias e até obesidade.

Estudos mostram que crianças amamentadas exclusivamente até os seis meses têm menos chances de desenvolver hipertensão, colesterol alto, diabetes tipo 2 e problemas de sobrepeso na infância e na vida adulta. As evidências científicas indicam que bebês prematuros e ou com patologias, que se alimentam de leite humano no período de internação na Unidade de Tratamento Intensivo (UTI) neonatal, possuem mais chances de recuperação e de terem uma vida mais saudável (Brasil, 2023).

Segundo dados do Instituto Brasileiro de Geografia e Estatística (IBGE), o Brasil registrou, em 2022, aproximadamente 2,5 milhões de nascimentos. Embora esse número seja menor do que em décadas anteriores, ele ainda representa uma população significativa de recém-nascidos que precisam de atenção e cuidados desde os primeiros dias de vida (IBGE, 2023).

A taxa de mortalidade infantil no país também apresenta queda nos últimos anos, mas ainda exige atenção. De acordo com o Ministério da Saúde, em 2021 a taxa foi de 11,2 mortes para cada mil nascidos vivos (Brasil, 2022). A Organização Mundial da Saúde (OMS) destaca que o aleitamento materno pode reduzir em até 13% a mortalidade infantil em menores de 5 anos, principalmente em regiões onde o acesso a serviços de saúde é limitado (OMS, 2023).

Além dos benefícios para o lactente, o aleitamento materno também oferece benefícios para a lactante, como a redução do risco de câncer de mama e ovário, além de contribuir para o fortalecimento do vínculo afetivo entre mãe e filho (Brasil, 2023).

Diante desses dados, é evidente a importância de promover e incentivar o aleitamento materno como prática essencial para melhorar a saúde das pessoas.

## 2.2 BANCOS DE LEITE HUMANO

Os Bancos de Leite Humano (BLHs) são instituições especializadas que têm como principal função coletar, processar, armazenar e distribuir leite humano doado por mulheres lactantes. Esse leite é destinado, principalmente, a recém-nascidos prematuros ou de baixo peso que, por algum motivo, não podem ser alimentados diretamente pelas próprias mães (RBLH-BR, 2023).

O Brasil possui a maior e mais bem organizada rede de BLH do mundo, sendo referência internacional. Essa rede se destaca por oferecer um serviço de alta qualidade técnica com baixo custo, além de contar com profissionais capacitados, equipamentos apropriados e rígidos padrões de controle de qualidade para garantir a segurança do leite oferecido aos lactentes (Brasil, 2023). O Brasil possui, em 2023, 233 Bancos de Leite Humano (BLHs) e 240 postos de coleta, distribuídos por todos os estados e no Distrito Federal, integrando a rede rBLH-BR (Brasil, 2023).

O funcionamento de um BLH segue etapas bem definidas. Primeiramente, as mães doadoras passam por uma triagem, que avalia seu estado de saúde e hábitos de vida. Após aprovadas, elas recebem orientações sobre como fazer a coleta do leite de forma segura em casa. O leite é então armazenado em frascos esterilizados e transportado ao banco de leite, onde passa por um processo de pasteurização. Antes de ser distribuído, o leite é analisado em laboratório para garantir que está livre de contaminações e com valor nutricional adequado para os lactentes que irão recebê-lo.

Além disso, os bancos de leite não apenas realizam a coleta e distribuição, mas também exercem um papel fundamental na promoção e apoio ao aleitamento materno. Eles orientam as lactantes, ajudam a superar dificuldades com a amamentação e estimulam a doação de leite humano como ato solidário.

O BLH é um exemplo de política pública eficiente, que contribui diretamente para a redução da mortalidade neonatal e para o desenvolvimento saudável de milhares de crianças em todo o país.

### 2.2.1 Tipos de Incentivos Oficiais à Doação de Leite Materno

A doação de leite materno é um ato voluntário essencial para a sobrevivência de muitos recém-nascidos, mas ainda enfrenta baixa adesão. Por isso, políticas públicas de incentivo são importantes para estimular novas doadoras.

Um exemplo é a Lei nº 7.711/2025, sancionada no Distrito Federal, que garante isenção da taxa de inscrição em concursos públicos do governo para mulheres que comprovem doação regular de leite materno — pelo menos duas doações mensais durante três meses, nos três anos anteriores à inscrição (Distrito Federal, 2025a; 2025b).

De acordo com a Secretaria de Saúde do DF, essa medida busca valorizar as doadoras e aumentar o número de participantes na rede. Em 2024, foram registradas 6.625 doadoras, e até maio de 2025, 2.613 mulheres já haviam doado (Secretaria de Saúde do DF, 2025c).

Essas ações demonstram como o poder público pode contribuir para ampliar o alcance da doação de leite humano e fortalecer a solidariedade.

## 2.3 ATUAÇÃO DO POSTO DE COLETA DE LEITE HUMANO NO HOSPITAL DAS CLÍNICAS SAMUEL LIBÂNIO

O Hospital das Clínicas Samuel Libânio (HCSL), localizado em Pouso Alegre, Minas Gerais, abriga um Posto de Coleta de Leite Humano que desempenha um papel crucial no apoio a recém-nascidos prematuros e ou de baixo peso que não podem ser amamentados por suas mães. Este posto atua na coleta, armazenamento e distribuição de leite materno para bebês internados em unidades como a UTI Neonatal, UTI Pediátrica e Unidade de Cuidados Intermediários Neonatais (UCIN) do próprio hospital. ​

O processo de doação é simplificado para incentivar a participação das mulheres lactantes. As interessadas podem entrar em contato o HCSL para receber orientações e um *kit* de coleta, permitindo que realizem o procedimento em casa. O leite coletado é armazenado e posteriormente recolhido pelo Corpo de Bombeiros, parceiro do Posto de Coleta do HCSL, que transporta o leite cru armazenado pelas doadoras até o Banco de Leite Humano de Varginha para exames, testes e pasteurização. Após esse processo, o leite retorna ao Posto de Coleta do HCSL, pronto para ser fornecido aos bebês necessitados (HCSL, 2025).

Além da coleta e distribuição, o posto oferece suporte às lactantes e gestantes, promovendo o pré-natal pediátrico e incentivando a amamentação exclusiva nos primeiros meses de vida da criança.

O Posto de Coleta de Leite Humano do HCSL está localizado à Rua Comendador José Garcia, nº 777, no Centro de Pouso Alegre. O atendimento externo ocorre de segunda-feira à sexta-feira, das 7h às 17h, e o atendimento interno, das 7h às 19h. Para mais informações ou para se tornar uma doadora, as mulheres podem entrar em contato pelo telefone (35) 3429-3200, ramal 3276. ​

A doação de leite materno é fundamental para a recuperação e desenvolvimento de bebês prematuros, e o apoio das mulheres doadoras é essencial para manter os estoques adequados e salvar vidas.

## 2.4 TRABALHOS RELACIONADOS

Diversos estudos têm investigado o uso de tecnologias móveis para apoiar a promoção do aleitamento materno, mostrando que essas ferramentas podem ser eficazes na ajuda às mães durante a amamentação. Um estudo de Lopes *et al*. (2022), em uma revisão integrativa da literatura, identificou que aplicativos móveis oferecem uma forma prática e acessível de fornecer informações importantes e suporte contínuo durante o período de amamentação. Esse estudo destaca como essas plataformas podem melhorar o conhecimento das mulheres sobre a amamentação e ajudar a superar desafios comuns nesse processo.

Outro trabalho relevante foi realizado por Silva *et al.* (2022), que analisaram a implementação do aplicativo Doe Leite, uma ferramenta digital criada para facilitar a doação de leite materno. O aplicativo conecta doadoras a bancos de leite humano, permitindo que as usuárias se cadastrem, agendem coletas e acompanhem o impacto das suas doações. Os resultados do estudo indicam que o aplicativo ajudou a aumentar a adesão à doação, facilitando o acesso à informação e proporcionando um meio mais eficiente de organizar as doações.

A Figura 1 a seguir mostra algumas telas do aplicativo Doe Leite.



FIGURA 1 – Aplicativo Doe Leite

Fonte: Silva *et al*. (2022).

O CuidarTech Doe Leite é um aplicativo desenvolvido para facilitar o processo de doação de leite materno, conectando doadoras a bancos de leite humano. Ele oferece funcionalidades como o cadastro de doadoras, agendamento de coletas e acompanhamento das doações realizadas, promovendo um aumento na adesão e no impacto positivo das doações. O uso de tecnologias móveis, como o CuidarTech, proporciona uma maneira prática e eficiente de engajar mais pessoas na doação de leite humano, beneficiando a saúde de lactantes e lactentes.

Além disso, Oliveira, Souza e Lima (2021) realizaram uma pesquisa sobre *websites* destinados à promoção do aleitamento materno. O estudo demonstrou como a criação de *websites* específicos tem sido uma estratégia eficaz para oferecer informações claras e práticas às lactantes, além de promover o aleitamento materno pelas mulheres de forma contínua. Esses *websites* fornecem conteúdos educativos e também facilitam o contato com profissionais de saúde, contribuindo para uma experiência de amamentação mais informada e tranquila.

Esses trabalhos refletem a relevância das tecnologias de informação e comunicação digitais para o apoio ao aleitamento materno e oferecem uma base sólida para o desenvolvimento de soluções tecnológicas voltadas ao aumento da doação de leite humano. A implementação de tais soluções é fundamental para melhorar o acesso à informação e aumentar a adesão de lactantes ao processo de amamentação e doação de leite.

No próximo capítulo apresentam-se os objetivos do presente trabalho.

# 3 OBJETIVO DO PROJETO

Neste capítulo é apresentada a formulação do problema, que enfatiza a necessidade da realização deste projeto, os objetivos almejados, a justificativa e os níveis de decisão e grupos funcionais atendidos pelo presente projeto. Também constam a contribuição deste projeto para a sustentabilidade e o modelo de negócio previsto.

## 3.1 FORMULAÇÃO DO PROBLEMA

A escassez de informações sobre os métodos de doação, coleta e uso do recurso do leite materno acarreta em problemas para as lactantes que produzem muito leite e acabam não o aproveitando, quanto para os recém-nascidos que que podem possuir problemas com o leite da própria mãe ou ela não produzir leite o suficiente para alimentar a criança.

É importante destacar o depoimento da enfermeira Jamila Leal, profissional da HCSL. Em entrevista para o G1 Sul de Minas, a enfermeira ressalta a importância da doação de leite materno para salvar vidas e a dedicação da equipe em oferecer suporte completo às doadoras, desde a orientação até a coleta segura do leite na residência das lactantes. Ela enfatiza que cada gesto de solidariedade ajuda a garantir a nutrição adequada dos recém-nascidos internados, especialmente os mais vulneráveis, e reforça o convite para que mais lactantes participem dessa corrente do bem (G1 Sul de Minas, 2025).

Uma entrevista presencial com a Sara Elisa Capelo de Lima, estudante do 5º período de Enfermagem da FAI, também permitiu conhecer e analisar as atividades e entender algumas dificuldades que essa área da saúde enfrenta (Lima, 2025).

Com base nas informações obtidas, é possível averiguar que a visibilidade e meios de contato com os órgãos responsáveis pela coleta, armazenamento e distribuição são feitas por meio de algumas campanhas realizadas ao longo do ano pelo Governo Federal, sem um recurso tecnológico mais prático. Também é possível perceber que o processo de divulgação da coleta ou distribuição não é realizado de forma categórica, sem datas e intervalos padronizados ou postos de coleta com fácil localização.

Os desafios aqui citados permitem entender que uma solução automatizada por meio de um sistema de informação pode contribuir com a redução das dificuldades enfrentadas pelas mulheres, lactantes e instituições de postos de coleta ou BHL, de modo geral.

## 3.2 OBJETIVOS

O objetivo geral deste projeto de conclusão de curso é desenvolver um sistema de informação, para uso com tecnologias *Web* e *mobile*, a fim de oferecer à população um sistema automatizado para a divulgação das campanhas de coletas de leite humano, localizações de postos de coleta mais próximos ao cidadão, bem como as formas de coleta em casa, armazenamento e locais de retirada.

Os objetivos específicos são:

1. desenvolver um aplicativo *Web* destinado a apoiar as instituições voltadas para a saúde das lactantes e lactantes;
2. desenvolver um aplicativo *mobile*, destinado às pessoas interessadas nas técnicas para coleta e armazenamento do leite materno em casa, pessoas com interesse em receber o leite materno e campanhas de bem-estar da mulher e da criança;
3. realizar a integração desses aplicativos por meio de uma API;
4. divulgar esclarecimentos e informações sobre o leite materno com fontes confiáveis em eventos públicos.

Por meio de um aplicativo *Web*, será intermediado o processo de divulgação de campanhas de coletas, retiradas do leite materno, bem como campanhas de saúde paras as mães, lactantes e cuidados com o lactente. Por meio do aplicativo *mobile* será possível que as pessoas interessadas tenham acesso a postos de coleta em sua região, informativos sobre campanhas e instruções de como coletar e armazenar o leite materno.

O sistema de software deverá permitir que seus usuários:

1. o cadastro como doadora do leite materno;
2. o cadastro como recebedora do leite materno;
3. o acesso às campanhas realizadas pela área da saúde;
4. o acesso aos informativos e campanhas sobre o cuidado com a mulher e a criança.

## 3.3 JUSTIFICATIVA

Com base nas entrevistas e pesquisas realizadas, é possível identificar que as principais dificuldades estão a falta de informações além dos postos de saúde de como receber o leito armazenado e como doar o leite, há também a falta de uma divulgação mais assertiva.

Tendo como base essas informações um sistema que facilite a divulgação dessas informações e campanhas para a população no geral ajudará o fluxo de doação, abrindo assim, possibilidade de alcançar mais pessoas.

## 3.4 NÍVEIS DE DECISÃO E GRUPOS FUNCIONAIS ATENDIDOS

O sistema de software atuará apenas com informações em nível operacional, de acordo com o seguinte:

1. a usuária interessada na doação do leite materno fará um cadastro para ter acesso às funções do sistema e poderá apontar que tem interesse em doar o leite humano;
2. o(a) usuário(a) interessado na participação das campanhas de bem-estar e coleta do leite materno irá indicar o interesse. Entretanto, será colocada em uma lista de seleção, onde a instituição responsável pela campanha e coleta, dado os critérios estabelecidos pela mesma.

O sistema de software a ser desenvolvido tem como público de interesse as pessoas que desejam doar ou receber o leite materno, aquelas que realizam campanhas de cuidado e bem-estar em saúde e os hospitais e postos de saúde responsáveis.

## 3.5 ADERÊNCIA AOS OBJETIVOS DE DESENVOLVIMENTO SUSTENTÁVEL (ODS)

O projeto Donate tem relação direta com alguns dos ODS da Agenda 2030, principalmente no que diz respeito à saúde e ao combate à desnutrição infantil (ODS Brasil, 2025).

Um dos pontos principais está no ODS 3 – Saúde e Bem-Estar, que busca reduzir as mortes evitáveis de recém-nascidos e crianças menores de cinco anos. O projeto poderá ajudar nesse objetivo ao facilitar o processo de doação de leite materno, conectando doadoras, postos e bancos de leite e famílias que precisam. Com isso, crianças prematuras ou em situação de risco de morte terão mais chances de receber a nutrição adequada para sobreviver e se desenvolver de forma saudável.

Além disso, este projeto também se relaciona com o ODS 2 – Fome Zero e Agricultura Sustentável, em especial com a meta de acabar com todas as formas de desnutrição. O leite materno é o alimento mais completo para os primeiros meses de vida e, ao tornar o acesso a ele mais rápido e organizado, o projeto contribuirá para que crianças em situação de vulnerabilidade tenham melhores condições de saúde e qualidade de vida.

Assim, o Donate não é apenas um sistema tecnológico, mas também uma ferramenta social que pode ajudar o Brasil a avançar em metas importantes dos ODS, reforçando a importância da solidariedade e do cuidado com a saúde das crianças.

## 3.6 MODELO DE NEGÓCIO

O modelo de negócio representa a forma como o sistema Donate cria, entrega e captura valor para seus usuários e instituições envolvidas no processo de doação de leite humano. Essa estrutura permite visualizar os principais elementos estratégicos do projeto, como o segmento de clientes, a proposta de valor, os canais de comunicação, as atividades-chave, os recursos necessários, as parcerias estratégicas, além da análise de custos e potenciais fontes de receita.

# 4 MÉTODOS GERENCIAIS

Nesse capítulo são apresentados os métodos e modelos de gerenciamento que é utilizado no projeto com o objetivo de demostrar os esforços e conceitos de gestão de projetos aplicados para abertura, planejamento, monitoramento e controle e encerramento bem-sucedido deste empreendimento temporário.

## 4.1 GERENCIAMENTO DO PROJETO

O Termo de Abertura do Projeto apresenta os aspectos iniciais do projeto, incluindo seus objetivos, escopo, justificativa e premissas básicas. Já o *Business Case* oferece uma análise aprofundada dos benefícios, custos e impactos esperados, auxiliando na tomada de decisão e na viabilidade da iniciativa.

As saídas dos processos de gerenciamento de projetos encontram-se no Apêndice A.

## 4.2 MODELO DE CICLO DE VIDA

Um modelo de ciclo de vida para desenvolvimento de sistemas de software pode conter várias atividades sendo as principais: especificação, projeto, implementação, verificação e validação, manutenção e evolução. Estas atividades geram as necessidades mínimas para que se possa obter um produto de software de qualidade.

Para realização deste projeto, utiliza-se o Modelo Incremental, cujos incrementos distribuídos por fase estão descritos no Quadro 1.

|  |  |  |
| --- | --- | --- |
| Incremento | Descrição da entrega | Data prevista |
| Entregas do *Business Case,* Termo de Abertura de Projeto (TAP)e início da documentação do projeto. | Primeira entrega. | 29/03/25 |
| Realizado a alteração na documentação, correção dos apontamentos da professora orientadora. | Correção e melhorias. | 19/04/25 |
| Elaboração do resumo, preenchimento da seção de métodos gerenciais, definição do gerenciamento de configuração, continuidade do gerenciamento do projeto, finalização do Capítulo 6 e conclusão parcial da fase. | Segunda entrega. | 02/05/2025 |
| Criação dos slides para apresentação oral | Apresentação oral das Fases 1 e 2. | 16/05/2025 |
| Realização das correções apontadas pela professora orientadora. | Entrega das melhorias. | 24/05/2025 |
| Últimos ajustes na documentação para entrega final do semestre. | Entrega da documentação e da codificação. | 07/06/2025 |
| Apresentação para a banca de qualificação. | Bancas de qualificação dos projetos | 14/06/2025 a 17/06/2025 |
| Entrega do projeto após apresentação. | Entrega da documentação e da codificação. | 28/06/2025 |
| Correção da Fase 2 | Entrega da documentação e codificação corrigida após a banca de qualificação. | 30/06/2025 |
| Avaliação de conhecimentos da fase 3 | Apresentação oral Fase 3 | 05/09/2025 a 12/09/2025 |
| Entrega da documentação após apresentação | Entrega da documentação | 06/09/2025 |
| Entrega das correções solicitadas | Entrega da documentação | 27/09/2025 |
| Apresentação do projeto na FAITEC | Apresentação pública do projeto na feira tecnológica FAITEC | 01/10/2025 a 03/10/2025 |
| Avaliação de conhecimentos da fase 4 | Apresentação oral Fase 4 | 07/11/2025 |
| Entrega das correções solicitadas | Entrega da documentação | 15/11/2025 |
| Entrega do projeto, vídeo e dos slides para a banca examinadora final | Entrega de artefatos | 22/11/2025 |

QUADRO 1 – Incrementos previstos

## 4.3 RECURSOS NECESSÁRIOS

Para realizar o projeto, são necessários três graduandos em Sistemas de Informação, sendo eles: Dilton Thales Melo da Silva, Lucas dos Reis Severini, Mateus Boche Daniel.

É de responsabilidade de todos os membros, em conjunto: análise, projeto, confecção deste documento, programação de código-fonte e realização de testes com várias estratégias.

Os recursos de software necessários para este projeto:

1. Sistema Operacional Windows 10 de 64 bits;
2. IntelliJ IDEA 2021.3.3;
3. PostgreSQL 14;
4. PgAdmin 4 v1.2;
5. Office 2016;
6. Visual Studio Code 1.32.3;
7. Ideas Modeler 11.96;
8. Postman 11.20.0;
9. Git 2.49.0;
10. Github;
11. Java SDK 17;
12. Bootstrap; 5.3.3
13. Google Forms;
14. Adobe Photoshop 2022 v23.3.1.426;
15. HTML 5;
16. JavaScript ES14;
17. ChatGPT 4.1;
18. Copilot 1.7.4421;
19. Node.js com Express 4.19.2;
20. Sprint Boot 3.5.3.

Os recursos de hardware necessários para este projeto são:

1. computador *desktop* RAM de 16 Gbytes, SSD de 1 Tbyte Nvme2 e processador AMD 5 3600;
2. *laptop* Samsung com RAM de 4Gbytes, SSD de 256 Gbytes e processador Intel i5 12ª gen.;
3. *smartphone* Moto G60, Android 12, com 6 GB de RAM e 128 Gbytes de armazenamento interno, para testes do aplicativo móvel.
4. *smartphone* Xiaomi, com Android 11, 4 GB de RAM e 64 Gbytes de armazenamento interno;
5. *laptop* Acer, com RAM de 8 Gbytes, SSD de 512 Gbytes e processador Intel i3 10ª gen.

## 4.4 RELATÓRIO DE DESEMPENHO

O Relatório de Desempenho do Projeto encontra-se no Apêndice B.

4.5 GERENCIAMENTO DE CONFIGURAÇÃO

Para manter a organização e o controle dos artefatos produzidos ao longo do desenvolvimento deste projeto é utilizado o sistema de controle de versão Git, com o repositório GitHub, com serviço em Nuvem. Esse ambiente permite registrar todas as alterações realizadas, facilitando o acompanhamento da evolução dos documentos, códigos e diagramas, além de garantir que todos os integrantes tenham acesso às versões atualizadas do projeto.

Os integrantes da equipe participam de forma igualitária em todas as etapas, colaborando tanto na documentação quanto na construção do código-fonte do sistema. Todas as atividades são feitas em conjunto, com foco na cooperação, revisão mútua e melhoria contínua do conteúdo. A divisão do trabalho é feita de forma equilibrada, e os três integrantes participam ativamente do uso do repositório, edição dos arquivos e controle de versões.

Cada artefato produzido, como textos, códigos, imagens e diagramas, é identificado por nome e salvo com versionamento no GitHub. Sempre que uma alteração é realizada, uma operação de *commit* é criada com uma mensagem clara, informando o que foi modificado, por quem e em qual data. Isso possibilitou um acompanhamento completo das evoluções do projeto, além de permitir retornar a versões anteriores sempre que necessário.

O repositório está organizado com pastas bem-definidas, visando facilitar a navegação e manter a estrutura limpa. As principais pastas criadas estão na mostradas nas figuras a seguir.



FIGURA 2 – Estrutura de pastas do projeto



FIGURA 3 – Subdivisão das pastas da documentação do projeto.



FIGURA 4 – Subdivisão das pastas de codificação do projeto.



FIGURA 5 – Repositório do Github

Atualmente os projetos estão sendo desenvolvidos em pastas locais e conforme há atualizações os projetos atuais são enviados para a main. Para realizar testes o projeto é clonado novamente e realizado as alterações, caso tudo corra bem essa nova versão é enviada para a main.

# 5 ESPECIFICAÇÃO E ANÁLISE DOS REQUISITOS

Este capítulo apresenta os requisitos necessários para o desenvolvimento do sistema de software Donate. A Especificação dos requisitos é fundamental para garantir que o sistema atenda às necessidades dos usuários e cumpra seu propósito de forma eficaz. Para isso, são definidos os requisitos funcionais e não funcionais, descrevendo as principais funcionalidades e restrições que devem ser consideradas durante o desenvolvimento do software.

Os requisitos levantados têm como base pesquisas sobre a doação de leite humano materno, entrevistas com profissionais da área e a análise de sistemas similares existentes. A seguir, são detalhados os requisitos funcionais e não funcionais e a modelagem de análise realizada para o sistema.

## 5.1 REQUISITOS DO SISTEMA DE SOFTWARE

Os requisitos do sistema de software definem as funcionalidades e restrições necessárias para o desenvolvimento da aplicação. Eles garantem que o sistema atenda às necessidades dos usuários e cumpra seu propósito de facilitar a doação de leite materno

Nesta seção, os requisitos são classificados em três categorias descritas a seguir.

Essenciais: requisitos fundamentais para o funcionamento do sistema. Sem eles, a aplicação não atenderá aos seus objetivos principais.

Importantes: requisitos que melhoram a experiência do usuário e a eficiência do sistema, mas que não são indispensáveis para o funcionamento básico.

Desejáveis: requisitos adicionais que agregam valor ao sistema, proporcionando maior conveniência e aprimoramento, mas que não são prioritários na implementação inicial.

A seguir, são apresentados os requisitos funcionais do sistema, organizados conforme sua classificação.

### 5.1.1 Requisitos Funcionais

Os requisitos funcionais descrevem as funcionalidades que o sistema deve oferecer para atender às necessidades dos usuários. Eles foram classificados de acordo com sua prioridade, podendo ser essenciais, importantes ou desejáveis.

RF01 – Cadastro de usuário

O sistema deverá permitir o cadastro do usuário, seja doadora, receptora ou profissional de saúde, por meio do preenchimento apenas dos dados pessoais necessários para o funcionamento do software, como nome, e-mail, telefone, tipo de usuário e endereço completo. O cadastro só será validado se todos os dados obrigatórios forem fornecidos corretamente.

Além disso, o usuário deverá criar uma senha segura para acessar o sistema, garantindo a proteção de acesso aos seus dados pessoais. O aceite dos termos de uso é obrigatório para a conclusão do cadastro, assegurando que o usuário esteja ciente das políticas e responsabilidades ao utilizar o sistema.

As responsabilidades de cada tipo de usuário são definidas da seguinte forma:  
a) Doadora: poderá registrar doações de leite materno, atualizar informações sobre disponibilidade e visualizar orientações para coleta e armazenamento;  
b) Profissional de saúde: poderá validar cadastros, acompanhar o histórico de doações, fornece orientações técnicas e gerenciar dados relacionados ao postos de coleta e ou bancos de leite;  
c) Administrador: poderá designar outros usuários como administradores, incluir bancos de leite, municípios e excluir usuários.

Prioridade: essencial.

RF02 – Cadastro de ponto de coleta ou banco de leite humano

O sistema deverá permitir o cadastro de pontos de coleta e bancos de leite humano, coletando dados como nome, endereço, dados de contato, horários de funcionamento e localização geográfica (latitude e longitude). Esses dados devem ser acessíveis no sistema para facilitar a localização e o acesso das usuárias doadoras e receptoras ao banco de leite mais próximo.

O cadastro de bancos de leite humano será de responsabilidade exclusiva dos profissionais de saúde previamente cadastrados e validados no sistema. Essa restrição garante a veracidade e confiabilidade dos dados inseridos, além de preservar a integridade do sistema.

Prioridade: essencial.

Requisito relacionado: RN04 – Validação dos postos de coleta e bancos de leite.

RF03 – Autenticação de usuário

O sistema deverá permitir que o usuário faça autenticação utilizando seu e-mail e senha previamente cadastrados. Caso as credenciais não correspondam, o acesso será bloqueado até que a autenticação seja realizada.

Prioridade: essencial.

Requisito relacionado: RN02 – Autenticação e acesso ao sistema.

RF04 – Localização de ponto de coleta e banco de leite humano

O sistema deve fornecer uma lista dos pontos de coleta ou bancos de leite próximos à doadora, com base na localização informada, facilitando o acesso às informações do banco de leite humano. Para isso, a doadora precisa autorizar o uso da sua localização pelo sistema.

Prioridade: essencial.

RF05 – Notificação sobre o andamento da doação

O sistema deverá enviar notificações automáticas à doadora sobre o andamento de sua doação, informando mudanças de situação como: pendente, agendada para coleta, concluída ou cancelada. O objetivo é manter a usuária informada em tempo real sobre o processo.

Prioridade: importante.

Requisito relacionado: RN05 – Notificações sobre o status da doação.

RF06 – Registro de intenção de doação

O sistema deverá permitir que a doadora registre a intenção de doar leite materno, informando a quantidade de leite disponível para doação e a localização para agendamento da coleta.

Prioridade: essencial.

Requisito relacionado: RN03 – Registro de intenção de doação.

RF07 – Apresentar informações sobre doação de leite materno

O sistema deverá disponibilizar materiais educativos sobre a importância da doação de leite materno, com informações sobre boas práticas de coleta, armazenamento e transporte, oferecendo suporte e conhecimento sobre o processo.

Prioridade: essencial.

Requisito relacionado: RN06 – Conteúdo educativo.

RF08 – Apresentar suporte e dados para contato com o administrador do sistema.

O sistema deverá oferecer um canal de suporte eficiente para que os usuários possam receber dúvidas ou relatar problemas sobre o processo de doação, com uma resposta garantida dentro de até 48 horas úteis.

Prioridade: essencial.

Requisito relacionado: RN07 – Suporte ao usuário.

RF09 – Agendamento de coleta

O sistema deverá permitir que a usuária doadora agende a coleta de leite materno com base nas datas e horários disponíveis. Após a solicitação, o profissional de saúde responsável pelo posto de coleta/banco de leite humano deverá confirmar o agendamento, garantindo que a coleta ocorra de forma organizada e eficiente.

A administração envolverá o cadastro e gerenciamento do ponto de coleta/banco de leite, feito exclusivamente por profissionais de saúde validados. A confirmação, reagendamento ou cancelamento das coletas, com comunicação automática para a doadora conforme descrito no requisito relacionado.

Prioridade: essencial.

Requisito relacionado: RN05 – Notificações sobre o status da doação.

RF10 – *Dashboards* e relatórios de acompanhamento

O sistema deverá oferecer aos profissionais de saúde acesso a um painel administrativo (*dashboard*) que apresente informações em tempo real sobre o processo de doação e coleta de leite materno. Esse painel deverá incluir: número de doações em andamento, pendentes, concluídas e canceladas dentro de um determinado período de datas informado pelo usuário; agendamentos futuros e passados; e situação das doações por região e ponto de coleta/banco de leite.

Essas funcionalidades têm como objetivo apoiar a gestão, análise e tomada de decisões relacionadas ao funcionamento e eficiência da rede de doação.

Prioridade: Baixa.

RF11 – Atualização dos dados do usuário

O sistema deverá permitir que qualquer usuário autenticado (doadora, receptor (a), profissional de saúde) atualize seus dados pessoais e de perfil, incluindo nome, telefone e senha.

A atualização de dados pessoais, como e-mail e senha, exigirá a confirmação da identidade do usuário por meio de autenticação (reentrada da senha atual).

O sistema deverá garantir a integridade e a segurança das informações durante o processo de atualização.

Prioridade: essencial.

RF12 – Exclusão de conta e anonimização de dados

O sistema deverá permitir que o usuário solicite a exclusão da conta. Ao confirmar a exclusão, os dados pessoais deverão ser anonimizados, mantendo apenas registros estatísticos ou históricos que não permitam a identificação direta ou indireta da pessoa. A exclusão deverá ser precedida de uma confirmação explícita do usuário e de uma mensagem explicativa sobre os efeitos da exclusão.

Prioridade: essencial.

RF13 – Controle de histórico de senhas

O sistema deverá manter um histórico das senhas utilizadas por cada usuário, armazenando o *hash* de cada senha junto à data de criação. Durante o processo de redefinição de senha, o sistema deverá verificar se a nova senha já foi utilizada anteriormente. Caso tenha sido, a redefinição será rejeitada, e o usuário deverá informar uma senha diferente. O número de senhas anteriores a serem armazenadas e verificadas poderá ser definido por configuração do sistema (ex.: últimas 3 senhas).

Um aspecto importante relacionado às senhas é o período definido para sua atualização obrigatória.

Prioridade: essencial.

Requisito relacionado: RNF05 – Políticas de formação de senhas.

RF14 – Acesso ao painel da doadora

O sistema deverá disponibilizar à usuária doadora um painel exclusivo, onde será possível visualizar:

1. as doações anteriores com data, quantidade e *status* (pendente, agendada, concluída, cancelada);
2. agendamentos de coleta futuros e já realizados;
3. informações do posto de coleta/banco de leite ao qual está vinculada;

Prioridade: essencial

Requisitos relacionados: RF05, RF06, RF09.

RF15 – Registro de nova doação

Após a coleta de leite materno, a doadora deverá poder visualizar no sistema um registro da doação realizada, contendo:

1. Data e hora da coleta;
2. Quantidade coletada;
3. Posto de coleta ou banco de leite responsável;
4. Situação da doação.

Esse registro poderá ser gerado automaticamente pelo sistema após a confirmação da coleta pelo profissional de saúde.

Prioridade: essencial

Requisito relacionado: RF09.

RF16 – Gerenciamento de disponibilidade para doação

A doadora poderá atualizar o agendamento já feito, informando a nova data e hora para a coleta. Essa informação auxiliará na organização dos agendamentos por parte do posto de coleta/banco de leite.

Prioridade: importante.

Requisito relacionado: RF06

RF17 – Acesso a conteúdo e orientações

O painel da doadora deverá oferecer acesso rápido aos conteúdos educativos sobre: coleta adequada; técnica de coleta e armazenamento do leite materno; cuidados com a saúde da mulher e da criança; incentivo às doações e conscientização e notícias.

Prioridade: essencial.

Requisito relacionado: RF07

### 5.1.2 Requisitos Não Funcionais

Os requisitos não funcionais definem as propriedades e restrições do sistema, estabelecendo critérios como desempenho, segurança e usabilidade. Esses requisitos são essenciais para a qualidade da aplicação e sua aceitação pelos usuários (Sommerville, 2019).

A seguir, são listados os principais requisitos não funcionais do sistema de software Donate.

5.1.2.1 Requisitos de Produto

a) Requisitos de Eficiência

RNF01 – Tempo de resposta

O sistema deve processar as requisições de cadastro, atualização e registros de doação em até 2 s.

Prioridade: essencial.

RNF02 – Escalabilidade

O sistema deverá ser capaz de suportar um aumento no número de usuários sem comprometer seu desempenho quanto ao tempo de resposta ao usuário.

Prioridade: essencial.

b) Requisitos de Segurança (confiabilidade, disponibilidade, autenticidade e integridade)

RNF03 – Confiabilidade e disponibilidade

O sistema deverá estar disponível pelo menos 99% do tempo, garantindo funcionamento contínuo e minimizando períodos de inatividade.

Prioridade: essencial.

RNF04 – Autenticidade e proteção de dados

O sistema deverá utilizar o protocolo seguro (Hyper Text Transfer Protocol Secure (HTTPs) para garantir a transmissão segura dos dados entre cliente e servidor.

As senhas dos usuários deverão ser armazenadas de forma criptografada utilizando a técnica *bcrypt* e *JSON Web Tokens* (JWT), assegurando a proteção contra acessos não autorizados e vazamentos de dados pessoais. O algoritmo de criptografia HMAC-SHA256 deverá garantir a segurança na autenticação.

Prioridade: essencial.

RNF05 – Políticas de formação de senhas

O sistema deve implementar políticas de segurança para a criação de senhas, a fim de aumentar a proteção das contas dos usuários. As regras mínimas incluem:

1. mínimo de 8 caracteres e máximo de 16;
2. pelo menos uma letra maiúscula, uma letra minúscula, um número e um caractere especial;
3. impedimento de reutilização de senhas anteriores em redefinições de senha.

Para cumprir o item c), o sistema deverá manter um histórico das senhas anteriores com a data de criação, registrando de forma segura (*hash* com *bcrypt*). Esse controle deve ser implementado conforme um requisito funcional específico.

Essas medidas têm como objetivo reduzir riscos de acesso não autorizado e promover boas práticas de segurança.

Prioridade: essencial.

c) Requisitos de Portabilidade

RNF06 – Compatibilidade

O sistema deverá ser compatível com os principais navegadores modernos, garantindo uma experiência consistente e funcional para todos os usuários. A compatibilidade mínima deve abranger:

1. Google Chrome: versão 90 ou superior;
2. Mozilla Firefox: versão 88 ou superior;
3. Microsoft Edge: versão 90 ou superior;

Além disso, o aplicativo *mobile* deverá ser responsivo e funcionar corretamente em dispositivos móveis com sistemas operacionais Android, versão 8.0 (Oreo) ou superior.

Essas definições visam garantir ampla acessibilidade, especialmente para doadoras e receptoras que podem acessar o sistema principalmente via *smartphone*.

Prioridade: essencial.

d) Requisitos de Usabilidade

RNF06 – Interface de usuário intuitiva

O sistema deverá proporcionar uma interação com usabilidade e acessibilidade, garantindo que usuários com diferentes níveis de experiência consigam utilizá-lo facilmente.

Prioridade: essencial.

RNF07 – Apresentar informações de uso

O sistema deverá disponibilizar informações claras e acessíveis sobre como utilizar suas funcionalidades, por meio de seções de ajuda, FAQ (Perguntas Frequentes), dicas contextuais (*tooltips*) e guias passo a passo, especialmente, para as funcionalidades mais importantes, como cadastro, registro de doações, solicitações e agendamentos.

Essas informações devem estar disponíveis diretamente na interface de usuário do sistema, sem a necessidade de o usuário buscar suporte externo, promovendo maior autonomia e facilidade de uso.

Prioridade: essencial.

5.1.2.2 Requisitos Organizacionais

a) Requisitos de Entrega

RNF08 – Disponibilidade da versão inicial

O sistema deverá ter versões incrementais e funcionais entregues conforme o Quadro 1.

Prioridade: essencial.

b) Requisitos de Modelagem e Codificação

RNF09 – Boas práticas de desenvolvimento

O sistema deverá ser desenvolvido seguindo boas práticas de codificação, garantindo a modularização, legibilidade, padronização e facilidade de manutenção e expansão do código.

A implementação deverá utilizar as seguintes tecnologias e *framework*s:

1. *Front-end:* Angular (com TypeScript) e Node.js;
2. *Back-end* (API REST): Spring Boot (com SDK Java);
3. Folhas de estilo para páginas HTML: Bootstrap;
4. Aplicativo *mobile*: Flutter (com Dart).

Deverão ser seguidos padrões, como:

1. separação de responsabilidades com os padroes *Model-View-Controller* (MVC), *services* e *components*;
2. reutilização de código (funções utilitárias, componentes reutilizáveis);
3. uso de controle de versão do Git com mensagens de *commit* padronizadas;
4. comentários claros e documentação básica para API e funções críticas.

Prioridade: essencial.

5.1.2.3 Requisitos Externos

a) Requisitos de Interoperabilidade (hardware, software, comunicações)

RNF10 – Integração com APIs externas

O sistema deverá permitir integração com APIs externas para garantir funcionalidades essenciais, como:

1. Utilização de API de mapas para facilitar a localização dos postos de coleta e bancos de leite humanos, com provedor OpenStreetMap integrado por meio da biblioteca Leaflet, adotado neste projeto por ser leve, de fácil implementação e não exigir chave de API;
2. Serviços de envio de e-mail: Brevo para envio de código de recuperação de senha;
3. Sistema Gerenciador de Bancos de dados (SGBD): PostgreSQL, com servidor para disponibilidade e segurança dos dados persistidos;
4. Servidor de aplicação: Node.js para o *front-end e* SpringBootpara o *back-end* e API.

O sistema deverá garantir a configuração segura dessas integrações, incluindo o uso de variáveis de ambiente para armazenar chaves, credenciais e rotas, evitando travar "*localhost*" no código-fonte para facilitar a implantação posterior.

Prioridade: essencial.

b) Requisitos Éticos

RNF11 – Transparência na utilização dos dados

O sistema deverá fornecer informações claras sobre o tratamento dos dados pessoais dos usuários, garantindo conformidade com boas práticas éticas.

Prioridade: essencial.

c) Requisitos de Privacidade e Proteção dos Dados

RNF12 – Conformidade com a Lei Geral de Proteção de Dados (LGPD)

O sistema deve estar em conformidade com a LGPD (Brasil, 2018), de tal forma que os dados pessoais sejam protegidos e utilizados apenas com consentimento do usuário.

### 5.1.3 Principais Regras de Negócio

As regras de negócio estabelecem diretrizes e restrições que orientam o funcionamento do sistema, de tal forma do que esse opere conforme os objetivos propostos. Essas regras definem processos e condições que devem ser seguidos dentro do sistema de software.

A seguir, são apresentadas as principais regras de negócio do sistema, juntamente com sua relação com os requisitos funcionais.

RN01 – Cadastro de usuários

O cadastro de usuários somente será concluído se todos os dados obrigatórios forem fornecidos e os termos de uso para tratamento dos dados pessoais serem consentidos.

RN02 – Autenticação e acesso ao sistema

Apenas usuários cadastrados poderão acessar o sistema. A autenticação deve ser feita utilizando e-mail e senha cadastrados. Senhas incorretas bloqueiam o acesso após três tentativas consecutivas incorretas.

RN03 – Registro de intenção de doação

Somente usuárias cadastradas como doadoras podem registrar uma intenção de doação. A quantidade de leite disponível deve ser informada no momento do registro.

RN04 – Validação dos bancos de leite humano

Apenas Postos de Coleta e bancos de leite reais e públicos poderão ser cadastrados no sistema.

RN05 – Notificações sobre o andamento da doação

O sistema deverá notificar a doadora sempre que houver uma atualização sobre a situação em que se encontra a doação (pendente, agendada, concluída ou cancelada).

RN06 – Conteúdo Educativo

Apenas profissionais de saúde autorizados poderão publicar ou atualizar conteúdos educativos dentro do sistema.

RN07 – Suporte ao Usuário

O sistema deve oferecer um canal de suporte para dúvidas e problemas, garantindo resposta em até 48 horas úteis.

## 5.2 ANÁLISE DOS REQUISITOS

### 5.2.1 Visão Funcional

A visão funcional do sistema é baseada no modelo de casos de uso, que descreve as interações entre os usuários e o sistema. Essa abordagem permite entender as funcionalidades que o sistema deverá oferecer, representando os principais fluxos de eventos e as interações que ocorrem com seus atores.

O modelo de casos de uso consiste no diagrama de casos de uso, que ilustra as diferentes ações que os usuários podem realizar no sistema, e na descrição dos fluxos de eventos, que detalha os passos envolvidos em cada caso de uso (Sommerville, 2019).

O diagrama e as descrições dos fluxos de eventos estão no Apêndice C.

### 5.2.2 Modelo Conceitual dos Dados

Esta seção descreve o modelo de dados de um sistema com alto nível de abstração no qual as relações são construídas por meio da associação de um ou mais atributos das entidades. Trata-se do Modelo Conceitual representado por meio do Diagrama de Entidade-Relacionamento. O diagrama encontra-se no Apêndice D.

### 5.2.3 Modelo Inicial da Interface de Usuário

O modelo inicial da interface de usuário visa representar, de forma visual, como será a interação entre o sistema e os seus usuários. São modeladas páginas simples e intuitivas, respeitando os princípios de usabilidade e acessibilidade, para garantir uma boa experiência ao público de interesse: doadoras, receptoras e profissionais da saúde.

As páginas apresentam os principais fluxos de navegação do sistema, como autorização de acesso, cadastro, agendamento de coleta e acesso a orientações sobre doação de leite humano. Esses modelos ajudam na validação da proposta e orientam a construção visual do sistema na fase de desenvolvimento.

As imagens com os protótipos das interfaces estão organizadas no Apêndice E, conforme a estrutura deste trabalho.

# 6 ARQUITETURA E PROJETO DO SISTEMA DE SOFTWARE

Este capítulo aborda a arquitetura e o projeto do sistema de software Donate e compreende a definição da estrutura técnica do sistema, envolvendo a seleção das tecnologias utilizadas, a modelagem da aplicação, a organização dos módulos e componentes, bem como a descrição das principais funcionalidades implementadas.

A arquitetura de software representa um dos elementos fundamentais para assegurar a qualidade, escalabilidade, segurança e manutenibilidade de um sistema. Assim, são adotadas boas práticas de engenharia de software, aliadas a tecnologias modernas e adequadas aos objetivos do projeto, para se ter uma base sólida para o desenvolvimento e evolução da aplicação.

Neste capítulo são apresentados o modelo arquitetural adotado, os diagramas da *Unified Modeling Languag*e (UML), versão 5.2, que ilustram a estrutura e o funcionamento do sistema, os principais módulos desenvolvidos, além das decisões técnicas tomadas ao longo do processo de construção. Dessa forma, evidenciam-se as visões que permitirão o cumprimento dos requisitos funcionais e não funcionais previamente estabelecidos no capítulo anterior.

## 6.1 VISÃO ESTRUTURAL

Esta seção apresenta a visão estrutural do sistema, representando a organização lógica dos componentes tanto das camadas de *front-end* quanto da camada de *back-end*. O objetivo do diagrama é ilustrar como o sistema foi modularizado em pacotes, evidenciando a estrutura de alto nível e os relacionamentos entre os agrupamentos de funcionalidades.

No *front-end*, os pacotes estão organizados segundo as responsabilidades da aplicação cliente: segurança, interface de usuário e comunicação com o *back-end*. Já no *back-end*, os pacotes seguem uma separação por responsabilidades de controle, roteamento, lógica de negócio e persistência de dados.

A definição dos pacotes segue boas práticas de arquitetura de software, visando modularidade, reutilização de código e facilidade de manutenção. Os diagramas completos e detalhados desta seção estão disponíveis na pasta do Apêndice F.

### 6.1.1 Diagrama de Pacotes

No *front-end,* o código-fonte está estruturado em pacotes, conforme as seguintes suas responsabilidades funcionais:

Pacote *guards*

Contém mecanismos de proteção de rotas, como o *AuthGuard*, que restringe o acesso a determinadas páginas apenas para usuários autenticados. Segue o padrão de segurança da aplicação.

Pacote *pages*

Engloba todas as telas (componentes de interface) da aplicação. Cada subpacote representa uma funcionalidade ou módulo visual, como:

agendamento: página para agendamento da coleta de leite.

banco-próximo: página que exibe postos de coleta e bancos de leite próximos, utilizando geolocalização.

*login*: página de autenticação do usuário.

*register*: página de cadastro;

painel: página painel principal após *login* (*home*).

*map*: fornece um mapa interativo com os pontos de coleta e bancos de leite.

Pacote services

Responsável por conter os serviços que fazem a comunicação com o *back-end* via HTTPs. É onde se centraliza a lógica de acesso a dados, garantindo separação entre visual e lógica de negócios.

*Back-end* (Node.js com Express)

No *back-end*, os pacotes estão organizados com base na arquitetura *Model-View-Controller* (MVC), respeitando os princípios de responsabilidade única.

Pacote controllers

Contém os códigos responsáveis por controlar o fluxo das requisições e respostas. Cada *controller* lida com um recurso da aplicação (ex: usuário, agendamento).

Pacote middlewares

Contém funções intermediárias que interceptam requisições, como autenticação de JWT e validações. São utilizadas pelos roteadores antes de acionar os *controllers*.

Pacote *models*

define os esquemas de dados e modelos utilizados na aplicação, geralmente com Sequelize, Mongoose (biblioteca JavaScript) ou diretamente em PostgreSQL, garantindo a integridade da base de dados. Neste caso, utiliza-se a API de comunicação com PostgreSQL.

Pacote *routers*

responsável por mapear as rotas da aplicação, direcionando as requisições HTTPs para os *controllers* adequados. Utiliza *middlewares* sempre que necessário.

### 6.1.2 Diagramas de Classes

A classe Usuario é de muita importância no sistema. Ela contém atributos como nome, e-mail, telefone, CPF, senha e localização (latitude e longitude). Além disso, possui marcadores booleanos para identificar se o usuário é doador, receptor ou profissional da saúde. Cada objeto da classe usuário está associado à um objetivo da classe Município por meio do campo id\_municipio, representando a localização do mesmo. Os métodos da classe Usuario permitem criar, atualizar, consultar usuários e listar todos os usuários cadastrados.

A classe Município possui os atributos de identificação, nome e unidade da federação. Está associada a múltiplos objeto da classe usuário (associação \*), o que representa que um município pode estar associado 0 ou mais usuários.

A classe BancoLeite representa os pontos de coleta e bancos de leite cadastrados no sistema. Seus atributos incluem nome, descrição, telefone, e-mail, endereço e localização geográfica. Está associada a um objetivo da classe usuário (relação 1:\*), indicando que um usuário (geralmente administrador ou profissional) poderá ser responsável pelo cadastro de vários postos de coleta ou bancos de leite. As operações desta classe permitem criar, atualizar, consultar e listar.

A classe Notificação é responsável pelas mensagens enviadas pelo sistema. Cada notificação está ligada a um objeto da classe BancoLeite (relação 1:\*) e possui atributos como código, data de envio e mensagem. Ela permite criar, consultar e listar notificações.

Esse diagrama de classes organiza as responsabilidades de cada parte do sistema de forma clara, facilitando a manutenção e a implementação das funcionalidades propostas.

## 6.2 VISÃO COMPORTAMENTAL

Esta seção apresenta a visão comportamental do sistema, que modela como os diferentes componentes interagem entre si ao longo do tempo para atender aos requisitos funcionais propostos. A visão comportamental é essencial para compreender o fluxo de execução das funcionalidades, destacando os eventos, ações e respostas do sistema diante de interações iniciadas por usuários ou sistemas externos.

Para isso, são utilizados diagramas de sequência, que ilustram a comunicação entre os objetos e os módulos do sistema em momentos específicos, como autenticação, cadastro, agendamento de coleta e busca postos e ou bancos de leite. Esses diagramas modelam o comportamento dinâmico do sistema, tornando visíveis as dependências e responsabilidades de cada elemento envolvido.

Todos os diagramas apresentados nesta seção foram organizados no Apêndice F. Esses diagramas complementam a visão arquitetural estática já apresentada e oferecem uma perspectiva completa do funcionamento interno do sistema.

### 6.2.1 Projeto das Interações entre Objetos

O projeto das interações entre objetos modela como os diferentes elementos do sistema — como componentes da interface, serviços, controladores e modelos de dados — interagem entre si para realizar as funcionalidades propostas. Essa modelagem é fundamental para garantir que o sistema atenda corretamente aos requisitos funcionais e não funcionais estabelecidos.

As interações seguem um padrão conforme segue:

O usuário interage com a interface (componente do pacote *pages* do *front-end*);

A interface invoca os métodos dos serviços (*services*), que são responsáveis por se comunicar com a API (*back-end*);

No *back-end,* os *routers* recebem a requisição, direcionam para os *controllers*, que por sua vez validam dados com auxílio dos *middlewares* e interagem com os *models*, responsáveis pela manipulação do banco de dados.

Essas interações foram projetadas com base em princípios de separação de responsabilidades, promovendo maior organização, legibilidade e facilidade de manutenção do sistema.

6.2.1.1 Diagramas de Sequência

Os diagramas de sequência a seguir representam os principais fluxos de interação entre os objetos. Cada cenário está modelado considerando uma execução bem-sucedida, sem falhas, desde a interface até o posto de coleta/banco de dados, com resposta clara ao usuário ao final de cada ação. A seguir, estão descritos os principais fluxos mapeados.

O primeiro cenário mostra o processo de entrar do usuário. Ao abrir o aplicativo, o usuário informará seu e-mail e senha. Esses dados serão enviados ao *back-end,* que realizará a verificação das credenciais no banco de dados. Se os dados de entrada estiverem corretos, o sistema gerará um *token* de autenticação e retornará para o aplicativo, liberando o acesso para uso.

No segundo cenário, é descrito o agendamento de uma coleta de leite humano. O (a) usuário (a) acessará a opção "Agendar Coleta", preencherá os dados solicitados, como data, horário e local, e confirma o envio. O sistema receberá esses dados, realizará o registro no banco de dados e retornará uma mensagem de confirmação, informando que o agendamento foi realizado com sucesso.

O terceiro diagrama representa o fluxo de orientações. Ao acessar a tela inicial, o usuário seleciona a opção “Orientações” e, em seguida, poderá escolher entre diversos temas informativos, como quem pode doar, como preparar o frasco, formas de coleta e transporte do leite humano. Após a escolha de um tema, o sistema recuperará os dados do banco de dados e exibirá na tela do aplicativo, permitindo que o usuário tenha acesso ao conteúdo de forma rápida e clara.

O último fluxo representa a verificação de autenticação ao acessar áreas restritas do aplicativo. Sempre que o usuário tentar acessar uma funcionalidade protegida, como o agendamento ou o histórico de coletas, o sistema verificará automaticamente se o *token* de autenticação ainda está válido. Caso esteja, a autorização será concedida sem necessidade de nova autenticação, mantendo a segurança e a fluidez do uso.

## 6.3 VISÃO DOS DADOS

Nesta seção, é apresentado o desenvolvimento da modelagem lógica e a criação do banco de dados relacional do sistema de software Donate, utilizando o PostgreSQL como sistema gerenciador de banco de dados. Aqui se mostra a estrutura de uma base de dados que permitirá o registro e a organização dos dados dos usuários (doadoras, receptoras e profissionais de saúde), municípios, postos de coleta e bancos de leite e às doações realizadas. Para isso, constam as entidades principais, seus respectivos atributos e domínios, além das restrições de integridade, como chaves primárias e estrangeiras. A estrutura proposta garante a consistência e a integridade dos dados, facilitando a manipulação e o relacionamento entre as informações. A partir desse modelo, foi gerado o código em Structured Query Language (SQL) responsável pela criação das tabelas no banco de dados, permitindo a implementação prática da estrutura lógica planejada.

### 6.3.1 Modelo Lógico

Com base no modelo conceitual elaborado no Capítulo 5, nesta seção está o modelo lógico de dados, também conhecido como modelo operacional. Este modelo tem como objetivo representar, de forma mais próxima da implementação, as estruturas de dados que serão de fato persistidas no banco de dados relacional.

O modelo lógico define as entidades, seus atributos, domínios, tipos de dados e restrições de integridade, como chaves primárias e estrangeiras, garantindo a consistência e a normalização dos dados. Neste projeto, o modelo suportar a gestão dos usuários do sistema (como doadoras, receptoras e profissionais de saúde), aos municípios, aos postos de coleta e bancos de leite humano e às doações realizadas.

Cada tabela foi estruturada considerando as necessidades funcionais do sistema e o relacionamento entre as entidades, respeitando as boas práticas de modelagem de dados. O resultado é um modelo que serve de base direta para a criação do banco de dados físico em PostgreSQL, conforme descrito na próxima seção.

O modelo lógico completo, representado em formato descritivo e visual, encontra-se disponível no Apêndice D.

### 6.3.2 Dicionário de Dados do Modelo Lógico

O dicionário de dados encontra-se disponível exclusivamente no Apêndice E.

## 6.4 PROJETO DA INTERAÇÃO HUMANO-COMPUTADOR

A interação humano-computador (IHC) visa garantir que os usuários consigam utilizar o sistema de forma simples, eficiente e satisfatória, oferendo para isso, interfaces de usuário com alto grau de usabilidade. Para isso, são considerados os perfis de usuários, os padrões ergonômicos e as heurísticas de usabilidade mais adequadas ao contexto do sistema de software Donate.

### 6.4.1 Perfil de Usuário

Existem ao menos três perfis principais de usuários que irão interagir com o sistema: doadoras, receptoras e profissionais da saúde. As doadoras são mulheres que estão em fase de amamentação e desejam contribuir com a doação de leite humano para auxiliar outras crianças. Esse grupo necessita de orientações claras, acessíveis e com linguagem simples, além de informações seguras sobre os benefícios e o processo da doação. As receptoras são mães ou responsáveis por lactentes que receberão o leite doado. Elas buscam confiança no processo e desejam compreender como é feito o controle de qualidade e a triagem das doações. Já os profissionais da saúde incluem estudantes e técnicos de enfermagem, bem como outros colaboradores que atuam diretamente no setor da Saúde. Esses profissionais utilizam o sistema para orientar as doadoras, monitorar coletas e garantir a segurança e eficiência de todo o processo.

Com o objetivo de entender melhor as necessidades desses usuários, foram realizadas entrevistas com alunos do curso de enfermagem e conversas com a coordenação. A partir desses contatos, foi elaborado um questionário simples e direto, com perguntas objetivas sobre o projeto, abordando aspectos como acessibilidade, clareza das informações e utilidade do sistema. O questionário foi aplicado de forma anônima, entre os dias 15 e 25 de abril de 2025, com o objetivo de entender a percepção das pessoas sobre o sistema Donate. Ele foi enviado para alunos da FAI, amigos, colegas e familiares de mães que amamentam. Ao todo, 25 pessoas responderam ao formulário. As respostas ajudaram a melhorar a interface e o conteúdo do sistema, tornando-o mais claro, útil e fácil de usar. Os resultados completos encontram-se no Apêndice G deste trabalho.

### 6.4.2 Projeto da Interface de Usuário

O projeto da interface de usuário do sistema Donate é desenvolvido com foco na simplicidade, clareza e acolhimento, considerando que o público de interesse são pessoas interessadas em doar leite materno. Para isso, definiu-se uma padronização visual que auxilia tanto na usabilidade quanto na identidade do sistema.

As cores principais escolhidas são as cores rosa e o vermelha, transmitindo empatia, cuidado e acolhimento, enquanto a azul é utilizada para destacar botões de ação e informações relevantes, remetendo à confiança e segurança. Além disso, foram utilizados tons de fundo suaves em rosa-claro e azul-claro, que ajudam a organizar as seções de forma visualmente agradável e facilitam a leitura do conteúdo.

A tipografia adotada é do tipo sem serifa, por garantir a boa legibilidade em diferentes dispositivos. Os títulos aparecem em negrito e em cores fortes para chamar a atenção, enquanto os textos explicativos são apresentados em cinza-escuro, para contraste adequado e leitura confortável.

Outro ponto importante é a utilização de ícones ilustrativos, como a mãe com bebê, o símbolo de localização e o presente, que reforçam visualmente o significado de cada seção e tornam a interface mais intuitiva. Os cartões que organizam as informações possuem cantos arredondados e sombras leves, transmitindo modernidade e suavidade, enquanto os botões seguem um padrão arredondado, de fácil identificação e clique.

Além disso, a interface foi desenvolvida de forma responsiva, de tal forma que os elementos se adaptem automaticamente a diferentes tamanhos de tela, proporcionando uma experiência consistente em computadores, *tablets* e *smartphones*.

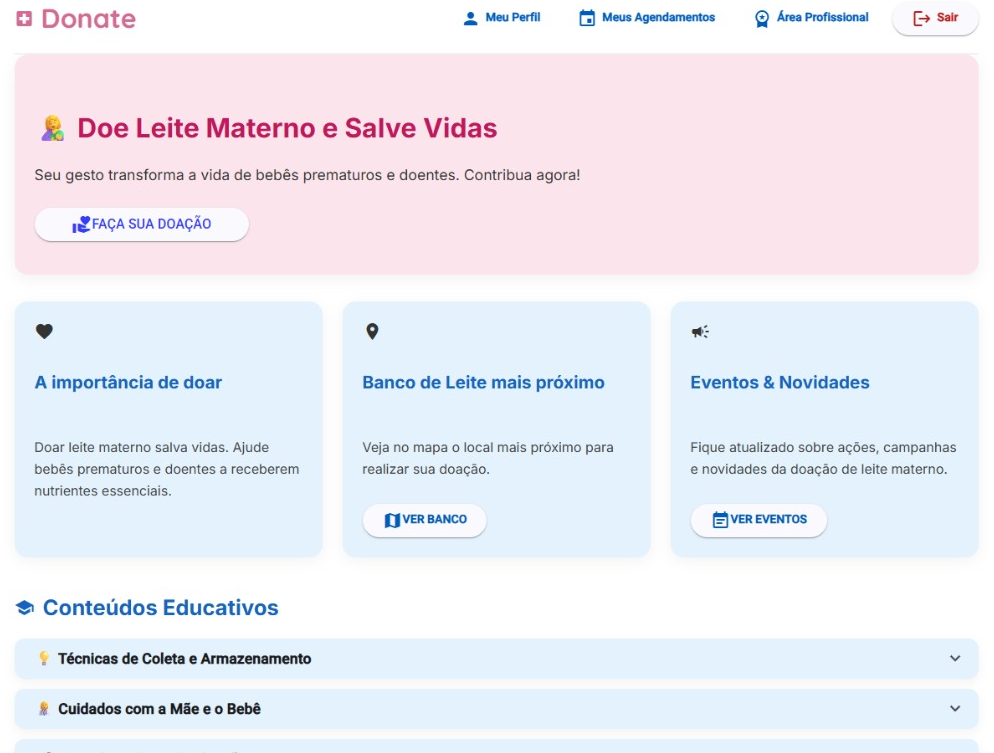


Figura 6 - Interface do usuário

### 6.4.3 Heurísticas de Usabilidade

No desenvolvimento da interface de usuário devem ser observadas as heurísticas de usabilidade de Jakob Nielsen. A visibilidade do status do sistema é garantida por mensagens claras e botões bem destacados, orientando o usuário sobre as ações que podem ser realizadas. A linguagem utilizada corresponde ao mundo real, trazendo mensagens simples e diretas, como “Encontre o posto de coleta ou banco de leite mais próximo” ou “Faça sua doação de leite materno”, acompanhadas de ícones que reforçam visualmente o conteúdo.

A consistência é mantida em toda a aplicação por meio do uso do mesmo padrão de cores, cartões, fontes e estilos. Isso facilita o reconhecimento, evitando que o usuário precise memorizar comandos ou caminhos para realizar suas tarefas. A interface também se mantém minimalista, sem excesso de elementos, priorizando clareza e objetividade.

Outro aspecto importante é o controle e a liberdade do usuário, já que ele pode acessar facilmente áreas como “Meu Perfil” e “Meus Agendamentos”, ou ainda sair do sistema quando desejar. Além disso, os botões são bem rotulados, o que ajuda na prevenção de erros, e o design geral se mostra simples o suficiente para atender tanto usuários iniciantes quanto mais experientes.

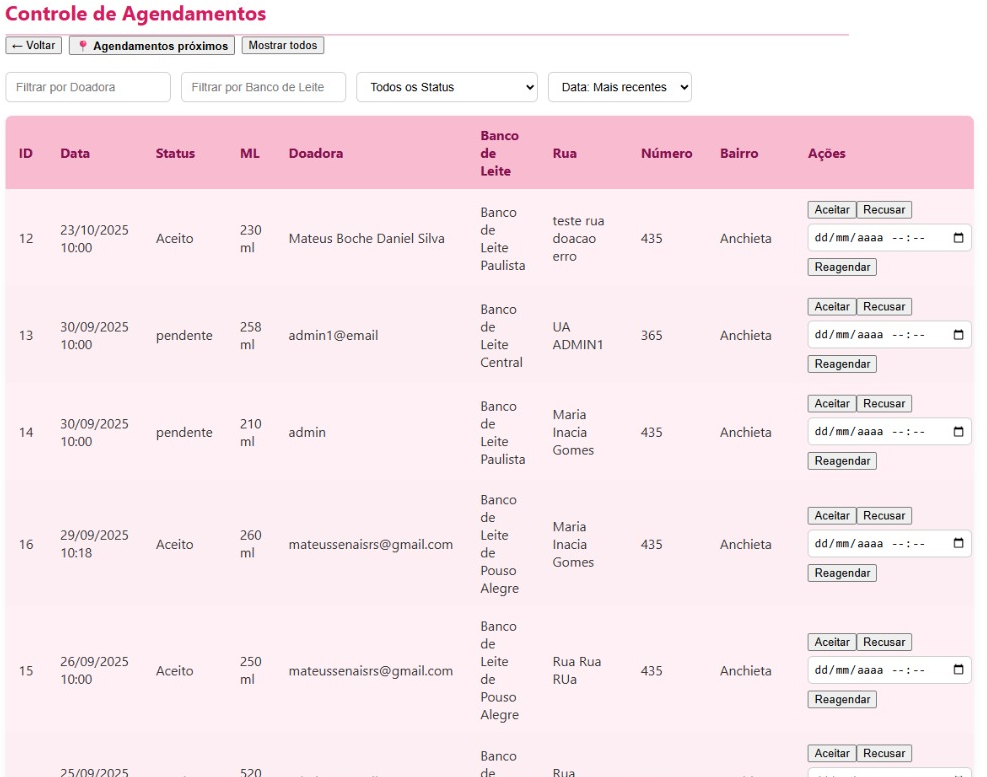


Figura 7 - Controle de Agendamentos

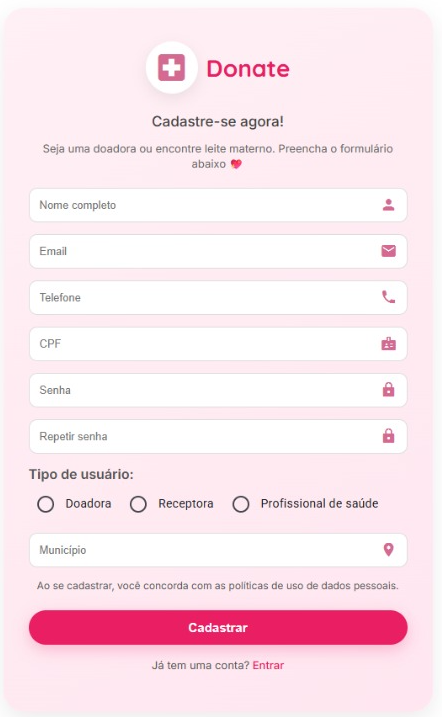


Figura 8 - Cadastro de Usuário

### 6.4.4 Projeto da Acessibilidade

O sistema Donate leva em consideração requisitos mínimos de acessibilidade, de forma a tornar a navegação mais inclusiva e eficiente. Para isso, são aplicados recursos como o contraste adequado entre textos e fundos, facilitando a leitura inclusive para pessoas com baixa visão. Também foram utilizados ícones acompanhados de textos, o que contribui para que os comandos e informações sejam mais facilmente compreendidos por diferentes perfis de usuários.

Outro cuidado importante é o dimensionamento dos botões, que possuem tamanhos maiores e com espaçamento adequado, favorecendo tanto o uso em computadores quanto em dispositivos móveis com telas sensíveis ao toque. Além disso, a estrutura semântica do conteúdo, com títulos, subtítulos e parágrafos bem definidos, oferece compatibilidade com leitores de tela, beneficiando usuários com deficiência visual.

A linguagem adotada no sistema também é simples e objetiva, evitando termos técnicos desnecessários, o que contribui para a clareza e acessibilidade da comunicação. Por fim, destaca-se o *design r*esponsivo, que permite a adaptação automática da interface a diferentes tamanhos de tela, assegurando que o sistema possa ser utilizado sem prejuízos de usabilidade em computadores, *tablets e smartphones*.

## 6.5 PROJETO DO SISTEMA DISTRIBUÍDO

Um sistema distribuído é aquele em que os componentes localizados em um computador estão conectados uns aos outros em uma rede, comunicam-se e coordenam suas ações apenas passando mensagens. Essa definição leva as seguintes características de sistemas distribuídos: dispositivos sendo acessados por vários usuários ao mesmo tempo, falta de relógio global e falhas de componentes individuais (Coulouris; Dollimore; Kindberg, 2007).

### 6.5.1 Procedimentos para Tratamentos dos Desafios

Esta seção descreve os principais requisitos necessários para um sistema distribuído apresentar interoperabilidade funcional. Esses requisitos são: heterogeneidade, escalabilidade, abertura, segurança, manuseio de falhas, concorrência e transparência.

6.5.1.1 Abertura

De acordo com Tanenbaum (2007, p. 4), um sistema distribuído aberto oferece serviços segundo regras padronizadas que descrevem sua sintaxe e semântica. Em geral, esses serviços são especificados por meio de interfaces bem definidas. A principal característica de um sistema aberto é a existência de interfaces documentadas e publicamente disponíveis, incluindo APIs, que descrevem funções, formas de uso e integração com outras aplicações. Um exemplo são aplicações que utilizam as APIs do Google Maps para consumir dados expostos por esses serviços, sem necessidade de contato direto com desenvolvedores do Google.

Testes de contrato: validação de esquema e Pact para garantir compatibilidade entre serviços/consumidores. Evolução de esquemas: compatibilidade retroativa; “ignorar campos desconhecidos”; migrações versionadas.

6.5.1.2 Concorrência

Em um ambiente concorrente, cada recurso deve ser projetado para oferecer consistência nos estados de seus dados. É essencial que todos os recursos estejam disponíveis, com o maior desempenho possível e para o maior número de usuários possíveis simultaneamente (Coulouris; Dollimore; Kindberg, 2007).

Transações com controle de concorrência (otimista/pessimista): garantem que duas edições simultâneas não se sobreponham (ex.: baixa de estoque).

Idempotência nos *endpoints* críticos: impede registros duplicados quando o usuário reenvia uma solicitação (ex.: confirmar a mesma doação duas vezes).

6.5.1.3 Escalabilidade

Um sistema é classificado como escalável se, perante um significativo número de recursos e usuários, ele permanece eficiente (COULOURIS; DOLLIMORE; KINDBERG, 2007).

Escala horizontal com balanceador de carga: várias instâncias da aplicação, dividindo as requisições. Alívio do banco (cache e réplicas de leitura): consultas vão para cache/replicas; o banco principal foca nas escritas.

6.6.1.4 Falha

Coulouris, Dollimore e Kindberg (2007) afirmam que falhas em sistemas distribuídos são parciais, pois todos eles possuem componentes que funcionam independentemente, portanto, mesmo com a falha de um componente, outros podem continuar operando normalmente, dificultando assim o manuseio de falhas. É essencial que todo componente seja desenvolvido com o objetivo de que ele, mesmo ocorrendo falhas dos componentes que depende, funcione ou trate destas falhas apropriadamente.

*Time-outs*, *retries* com *backoff* e *circuit breaker*: evitam travas e quedas em cadeia quando uma dependência está instável. Backups, replicação e testes de restauração: se algo der errado, os dados voltam rápido com perda mínima (RPO/RTO definidos).

6.5.1.5 Heterogeneidade

Com a Internet, é possível aos usuários acessarem serviços e executarem aplicativos por meio de um conjunto heterogêneo de computadores e redes (COULOURIS; DOLLIMORE; KINDBERG, 2007).

A heterogeneidade se aplica aos seguintes aspectos:

1. redes: via protocolos de Internet é possível ser realizada a comunicação;

Padronizar protocolos de app: expor serviços via HTTP/2/HTTP/3 (REST) e gRPC sobre TLS, com timeouts, retries e backoff padronizados para lidar com latência e perdas.

Gateway e normalização: usar API Gateway/Service Mesh para unificar autenticação, CORS, rate limiting e observabilidade, isolando diferenças de rede dos serviços.

1. *hardwares* de computador: diferentes padrões de placas de rede fazem que aconteça diferentes implementações. Exemplo: placa de rede Ethernet têm uma implementação diferente daqueles que possuem placas de outros padrões, isto acontece também com arquiteturas de microprocessadores (Intel, ARM, por exemplo);

Contêineres multi-arquitetura: imagens multi-arch (amd64/arm64) e builds reprodutíveis no CI, garantindo que a mesma versão rode em Intel ou ARM.

Abstração de dependências: priorizar serviços gerenciados (DB, storage, fila) e bibliotecas portáveis; evitar acoplamento a drivers/recursos específicos de NIC/CPU.

1. sistemas operacionais: diferentes sistemas operacionais trazem consigo diferentes formas de programação;

Padronização por contêiner: empacotar serviços em contêineres com base de imagens imutáveis e versões fixas, reduzindo variações entre Linux distros.

Matriz de compatibilidade no CI: *smoke tests* e *health checks* automatizados por ambiente, antes do deploy, para detectar diferenças de SO.

1. linguagens de programação: cada linguagem de programação possui particularidades, seja no tratamento de vetores, registros e variáveis. É necessário tratar essas diferenças para que não haja erros de dados e aconteça uma comunicação efetiva;

Contrato de dados único: API-first com OpenAPI/AsyncAPI e schemas (JSON Schema/Protobuf), UTF-8 e datas ISO-8601, mantendo compatibilidade retroativa.

SDKs oficiais e mensageria: oferecer SDKs (ex.: Java/TS) e integrar serviços via mensageria para comunicação confiável entre linguagens.

1. implementação de diferentes desenvolvedores: é necessário que diferentes desenvolvedores utilizem de padrões e convenções comuns durante a programação para que haja comunicação entre os diferentes sistemas.

Padrões e governança: guias de estilo, *linters/formatters*, convenções de API, ADRs (decisões de arquitetura) e revisão de código obrigatório.

Qualidade no CI/CD: testes (unitário/integração/contrato), *coverage* mínimo, *security scan* e versionamento semântico com plano de depreciação.

6.5.1.6 Segurança

Muitos recursos de informação que se tornam disponíveis e são mantidos em sistemas distribuídos têm um alto valor intrínseco para seus usuários. Portanto, sua segurança é de fundamental importância. As seguranças de recursos de informação têm três componentes: confidencialidade (proteção contra exposição para pessoas não autorizadas), integridade (proteção contra alteração ou dano), disponibilidade (proteção contra interferência com os meios de acesso aos recursos) e autenticidade (provar a identidade do usuário a traves de senhas, certificados digitais, assinaturas digitais) (COULOURIS; DOLLIMORE; KINDBERG, 2007).

6.5.1.7 Transparência

O desafio da transparência tem o objetivo de tornar aspectos da distribuição invisíveis para o desenvolvedor e usuários, a fim de que ele se preocupe apenas com o projeto de seu sistema em particular. O objetivo de tornar ocultos certos aspectos da distribuição é para que este seja percebido como um sistema único em vez de uma coleção de componentes independentes (COULOURIS; DOLLIMORE; KINDBERG, 2007).

6.5.2 Tecnologias e Arquiteturas de Distribuição

A sistema Donate possui uma arquitetura baseada em Cliente-Servidor. Para cada requisição de um cliente é o servidor que irá fazer todo o processamento e retornar as respostas, por meio de protocolos de rede e com passagens de métodos HTTP e Rest.

O servidor é responsável por validar as requisições do cliente, processar, persistir os dados e retornar o resultado. O cliente (um navegador) é o canal de comunicação com o usuário, sendo por intermédio dele que ele terá acesso ao sistema, além de prover a interface gráfica de usuário necessária para a interação.

O diagrama de distribuição do sistema Donate apresentado na Figura 6 mostra como funciona o sistema da perspectiva de um sistema distribuído. Nela está presente os processos clientes, que são constituídos pelos processos:

1. P1 - processo SGBD PostgreSQL: gerenciamento de acessos ao banco de dados;
2. P2 - processo Servidor de e-mail: recebimento e envio de e-mails;
3. P3 - processo *back-end* (Spring Boot): conecta APIs com o banco de dados, gerencia conexões de usuários e alimenta os aplicativos Web e *mobile;*
4. P4 - processo *front-end* (Angular e Node.js): interface gráfica do projeto;
5. P5 - processo do navegador cliente: processamento do aplicativo *Web*;
6. P6 - processo Donate *mobile*: processamento do aplicativo *mobile.*
7. P7 - processo Servidor de mapas: conecta a API de mapa

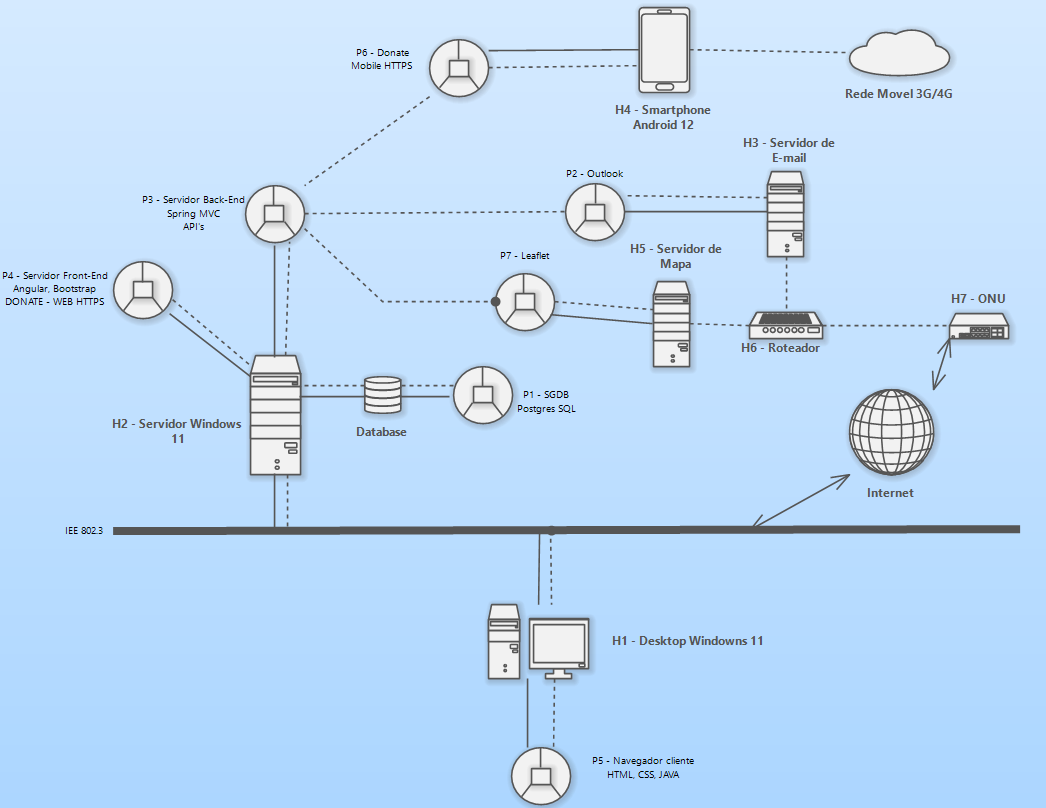


Figura 9- Um diagrama de sistema distribuído do Donate.

# 7 IMPLEMENTAÇÃO DO SISTEMA DE SOFTWARE

A fase de implementação representa a materialização prática de todo o planejamento, modelagem e especificação realizados nas etapas anteriores do projeto. Neste capítulo, são descritos os principais componentes desenvolvidos, as tecnologias empregadas e as soluções técnicas adotadas para transformar os requisitos levantados em um sistema funcional.

Serão detalhados os módulos de software implementados, a integração entre os componentes de front-end e back-end, bem como as bibliotecas, frameworks e linguagens utilizadas. Além disso, são apresentados os padrões de projeto aplicados e as estratégias adotadas para garantir qualidade, segurança e escalabilidade do sistema.

Dessa forma, este capítulo evidencia como as decisões de arquitetura foram traduzidas em código, consolidando a proposta do sistema Donate como uma ferramenta tecnológica voltada ao incentivo e à facilitação da doação de leite materno.

## 7.1 COMPONETES DO SISTEMA DE SOFTWARE

O sistema Donate tem uma arquitetura em camadas, com separação entre a interface de usuário, a lógica de negócio e o armazenamento dos dados. Essa divisão facilita a manutenção, a escalabilidade e deixa o sistema mais organizado.

1. *Front-end* (Aplicativo *Web*): é a camada do sistema com a qual os usuários interagem diretamente. Nela, as doadoras, receptoras e profissionais de saúde conseguem se cadastrar, acessar o sistema, consultar informações sobre pontos de coleta e bancos de leite e acompanhar as doações. Essa interface de usuário utiliza Angular e Node.js, e funciona em qualquer navegador, seja no computador ou *smartphone*, já que é responsiva;
2. *Back-end* (API): essa camada é responsável por processar as regras do sistema, validar cadastros, controlar as doações, gerenciar usuários e fazer a ligação com o banco de dados. Está desenvolvida em Linguagem Java, com Spring Boot, que fornece os serviços necessários e expõe uma API para que o *front-end* consiga se comunicar.
3. Banco de dados: todos registros do sistema são armazenados no banco de dados, como dados dos usuários, registros de doações, localização dos bancos de leite, entre outros. Para isso, utiliza-se o PostgreSQL, um banco de dados relacional confiável e de alto desempenho;
4. Serviço de Geolocalização: o sistema também conta com a funcionalidade de mostrar os postos de coleta/bancos de leite mais próximos do usuário. Para isso, utiliza informações de latitude e longitude registradas no cadastro e faz integração com APIs de mapas (OpenStreetMap), ajudando na localização de forma prática;
5. Serviço de envio de e-mails:

## 7.2 TECNOLOGIAS DE IMPLEMENTAÇÃO

### 7.2.1 Linguagens de Programação e *Frameworks* Adotados

O sistema utiliza tecnologias de software modernas e bem-consolidadas no mercado, conforme descritas a seguir.

1. Angular;
2. Node.js;
3. Spring Boot;
4. PostgreSQL;
5. OpenStreetMap;
6. Flutter.

### 7.2.2 *Design Patterns* Aplicados

Durante a implementação do sistema, são aplicados diversos padrões de projeto com o objetivo de manter a organização, reduzir o acoplamento entre os componentes e facilitar a manutenção do código. Esses padrões aparecem tanto no *back-en*d quanto no *front-end*.

1. MVC: padrão arquitetural aplicado na camada de *back-end*. A camada de modelo (entidades e repositórios) é separada da camada de controle (controladores Rest), enquanto o *front-end* funciona como a camada de visão. Essa divisão melhora a clareza do código e facilita a evolução do sistema;
2. *Repository Pattern***:** utilizado no *back-end* para centralizar e organizar as operações de acesso ao banco de dados. Assim, a lógica de persistência fica isolada em classes específicas, permitindo que as regras de negócio sejam mantidas de forma mais limpa e independente;
3. *Dependency Injection*: aplicado pelo Spring Boot para gerenciar dependências entre classes. Com isso, o acoplamento do código é reduzido e torna-se mais fácil substituir ou ampliar funcionalidades sem grandes modificações;
4. *Proxy Pattern* (AuthGuard no Angular): no *front-end*, o AuthGuard atua como um *proxy*, controlando o acesso às rotas da aplicação. Ele verifica a existência de um token de autenticação antes de liberar a navegação, garantindo maior segurança no acesso às telas restritas.
5. *Chain of Responsibility* (HTTP Interceptor no Angular): o interceptor implementado no Angular segue esse padrão, pois intercepta todas as requisições HTTP antes de chegarem ao servidor. Com isso, é possível adicionar automaticamente o *token* de autenticação, além de centralizar o tratamento de erros;
6. Separação em Camadas (MVC Adaptado no Angular): o Angular organiza o código em *components* (camada de visão), *services* (regras de negócio) e *models* (representação dos dados). Essa estrutura favorece o reuso de código, a testabilidade e a clareza no desenvolvimento.

Esses padrões, em conjunto, tornam o sistema mais modular, seguro e escalável, além de facilitarem o trabalho em equipe durante a evolução do projeto.

### 7.2.3 Convenções e Guias para Codificação

Durante o desenvolvimento do projeto Donate, foi fundamental definir um conjunto de convenções de codificação que garantisse padronização, legibilidade e fácil manutenção do código. Essas práticas facilitaram o trabalho em equipe, reduziram ambiguidades e tornaram o sistema mais robusto e organizado.

Adotamos a linguagem Java para o backend e TypeScript/Angular para o frontend, seguindo as boas práticas recomendadas pela comunidade de cada tecnologia. As principais convenções aplicadas foram:

1. Nomenclatura de classes: Utilizamos o padrão PascalCase, em que cada palavra começa com letra maiúscula (ex.: BancoLeite, Usuario, Agendamento). Essa escolha facilita a identificação das classes no projeto.
2. Atributos e variáveis: Optou-se por nomes significativos e descritivos, escritos em camelCase (ex.: idUsuario, dataHora, quantidadeMl), evitando abreviações que pudessem comprometer a clareza.
3. Métodos: Mantiveram-se verbos no infinitivo, também em camelCase, representando a ação que o método executa (ex.: criarUsuario(), atualizarStatusAgendamento(), registrarDoacao()).
4. Constantes: Quando necessárias, foram escritas em letras maiúsculas e, em casos de múltiplas palavras, separadas por sublinhado (ex.: TAXA\_MAXIMA, API\_BASE\_URL).
5. Comentários: Foram adicionados de forma sucinta acima dos métodos e blocos de código que exigiam explicação extra, priorizando a clareza sem excesso de texto.
6. Organização de pacotes: As classes foram agrupadas em pacotes conforme sua responsabilidade, como model, service, controller e repository. Essa separação seguiu o padrão MVC, facilitando manutenção e evolução do sistema.
7. Legibilidade: Seguiu-se a indentação padrão de 4 espaços e linhas curtas para melhorar a visualização do código em qualquer IDE.

Por exemplo, a classe BancoLeite contém atributos simples e autoexplicativos, enquanto seus métodos foram nomeados de modo a indicar claramente sua função, como criarBancoLeite() e listarBancosLeite(). O mesmo padrão foi aplicado às demais classes, como Usuario, Doacao, Agendamento e Municipio.

Essas convenções permitiram que todos os membros da equipe compreendessem facilmente a estrutura do código e colaborassem de forma eficiente. Além disso, o uso de nomes padronizados contribuiu para uma integração mais harmoniosa entre as diferentes camadas do sistema e facilitou futuras manutenções.

### 7.2.4 Estrutura Física do Banco de Dados

A estrutura física do banco de dados do projeto Donate foi projetada para armazenar e gerenciar informações de forma eficiente, garantindo integridade, segurança e consistência dos dados. Para isso, foram criadas tabelas relacionais normalizadas, com chaves primárias, estrangeiras e restrições de integridade referencial, atendendo aos requisitos do sistema.

O banco de dados inclui as principais entidades do sistema, como usuário, bancos de leite, município, unidade federativa, agenda, doação, eventos, histórico de senhas e recuperar senha. Cada tabela possui atributos bem definidos e tipos de dados adequados, como serial para identificação única, varchar para textos, numeric para coordenadas geográficas e timestamptz para registros de datas e horários com fuso horário.

Além das tabelas, o banco contempla relacionamentos que refletem as associações do modelo conceitual, garantindo rastreabilidade e consistência, por exemplo:

1. Um usuário está vinculado a um município e pode ter histórico de senhas ou registros de recuperar senha.
2. Um banco de leite pertence a um município e pode registrar múltiplas doações e agendamentos.
3. Eventos são associados a municípios para fins de controle e campanhas de conscientização.

Para fins de documentação, todos os scripts DDL utilizados para criação das tabelas, bem como informações sobre tipos de usuários e permissões concedidas no banco de dados, devem ser incluídos no Apêndice D. Essa prática permite que qualquer membro da equipe ou avaliador reproduza o banco de dados de forma confiável e serve como referência para manutenção futura.

A estrutura adotada garante desempenho adequado, integridade dos dados e segurança, atendendo aos objetivos do sistema Donate e possibilitando escalabilidade para futuras funcionalidades.

## 7.3 ANÁLISE DE COMPLEXIDADE ALGORÍTMICA

Para análise de complexidade algorítmica, foram selecionados 3 métodos principais do projeto, considerando o parâmetro "n" como o número de registros/entidades na base de dados que são processados em cada operação.

1. Método: DoacaoServiceImpl.buscarTodos()

@Override

public List<DoacaoListDTO> buscarTodos() {

return doacaoDao.listarTodas();

}

Complexidade O(n).

Este método realiza uma consulta simples no banco de dados que retorna todas as doações cadastradas. O tempo de execução cresce linearmente com o número "n" de registros na tabela de doações, pois cada registro precisa ser processado e convertido para DTO.

1. Método: MunicipioServiceImpl.findByNome(String nome)

@Override

public List<MunicipioModel> findByNome(String nome) {

String sql = "SELECT m.id AS m\_id, m.nome AS m\_nome, " +

"uf.id AS uf\_id, uf.nome AS uf\_nome, uf.sigla AS uf\_sigla " +

"FROM municipio m " +

"JOIN unidade\_federativa uf ON m.id\_unidade\_federativa = uf.id " +

"WHERE LOWER(m.nome) LIKE LOWER(?)";

return jdbcTemplate.query(sql, rowMapper, "%" + nome + "%");

}

Complexidade O(n).

A consulta utiliza operador LIKE com wildcard, que requer verificação em todos os registros da tabela municipio. Embora índices possam melhorar a performance, no pior caso a operação precisa varrer toda a tabela (busca sequencial), resultando em complexidade linear em relação ao número "n" de municípios.

1. Método: UserRestController.getEntities()

@GetMapping()

public ResponseEntity<List<UsuarioModel>> getEntities(){

List<UsuarioModel> users = userService.findAll();

return ResponseEntity.ok(users);}

Complexidade O(n).

Similar ao primeiro método, esta operação recupera todos os usuários do sistema. A complexidade é linear O(n) onde "n" representa o número total de usuários cadastrados, pois cada registro é mapeado para o modelo de domínio e retornado na resposta.

Todos os métodos analisados possuem complexidade linear O(n), o que é adequado para operações de consulta em bancos de dados. Para melhorar performance em cenários com grande volume de dados, seria recomendável implementar paginação e otimização de índices no banco de dados.

A complexidade O(n) considera apenas o processamento dos dados retornados, não incluindo o tempo de rede e latência do banco de dados.

8 PLANO DE TESTES

Os testes do sistema Donate estão planejados de forma sistemática para garantir a conformidade entre os requisitos levantados e a implementação desenvolvida. O objetivo central foi avaliar a estabilidade, usabilidade, integridade dos dados e desempenho da solução em cenários reais de uso, assegurando que as principais funcionalidades atendessem às expectativas definidas no projeto.

Neste capítulo é apresentado um plano de testes documentado para *o Donate*.

A descrição dos casos de testes e a demonstração dos resultados obtidos constam no Apêndice J.

## 8.1 FINALIDADE

Este presente plano de testes contempla diferentes estratégias, cada uma com uma finalidade específica:

1. testes unitários: verificam o funcionamento isolado de funções e métodos, como validação de cadastros e autenticação de usuários;
2. testes de integração: asseguram a comunicação entre os módulos da aplicação avaliando entradas, saídas e consistência de dados;
3. testes de sistema: garantem que o conjunto de funcionalidades implementadas, como cadastro de doadoras, registro de doações e consulta de bancos de leite funcione de forma integrada.

## 8.2 ESCOPO

O escopo dos testes abrange os principais módulos do sistema Donate:

1. o cadastro e autenticação de usuários (doadoras, receptoras e profissionais de saúde);
2. a gestão de campanhas de doação realizadas pelas instituições;
3. o registro e acompanhamento de doações de leite humano;
4. a consulta aos postos/bancos de leite humano via geolocalização (integração com OpenStreetMap);
5. a validação da responsividade da aplicação em diferentes dispositivos (*desktop*).

A execução contemplou cenários de uso reais, incluindo tanto fluxos corretos (entradas válidas) quanto fluxos de exceção (erros de preenchimento ou falhas de conexão).

### 8.2.1 Referências aos Documentos Relevantes

|  |  |
| --- | --- |
| Tipo do material | Referência |
| Documento de requisitos | Especificação funcional do sistema Donate. |
| Modelo de dados | DER. |
| Arquitetura de software | Descrição da arquitetura em camadas (MVC – Angular/Spring Boot/PostgreSQL) |
| Casos de uso | Diagramas de casos de uso (Visão Funcional). |

QUADRO 1 – Documentos relevantes para testes

### 8.2.2 Ambiente para a Realização dos Testes

|  |  |  |
| --- | --- | --- |
| Equipamento | Marca/modelo/configuração | Finalidade |
| Laptop 1 | Dell Inspiron 15, Intel i7, 16 Gbytes RAM e SSD 512 Gbytes. | Desenvolvimento e execução de testes locais |
| Laptop 2 | Acer Aspire 5, Intel i5, 8 Gbytes de RAM, SSD 256 Gbytes.   |  | | --- | |  | | Testes de compatibilidade e ambiente paralelo |

QUADRO 2 – Equipamentos para a realização dos testes

|  |  |  |
| --- | --- | --- |
| Software/versão | Fabricante | Finalidade |
| Angular 17 | Google | Desenvolvimento e testes no *front-end.* |
| Spring Boot 3.2 | Pivotal/VMware | Implementação e validação da API no *back-end.* |
| PostgreSQL 16 | PostgreSQL | Armazenamento e verificação da integridade dos dados. |
| PgAdmin 4 | PostgreSQL | Administração e execução de queries de teste. |
| Postman 11.20 | Postman Inc. | Testes de integração da API. |
| IntelliJ IDEA 2021.3.3 | JetBrains | Desenvolvimento e execução de testes unitários. |
| JUnit 5 | JUnit Team | Execução de testes unitários em Java, validação de métodos e lógica de negócios |
| Google Chrome | Google | Testes de interface e compatibilidade entre navegadores. |

QUADRO 4 – Softwares para a realização dos testes

## 8.3 ESPECIFICAÇÃO DOS CASOS DE TESTES

## A descrição detalhada dos casos de teste e os resultados de execução estão documentados no Apêndice J.

## Para cada caso de teste, foram especificados os valores de entrada e os resultados esperados, garantindo rastreabilidade e clareza na execução. Entre os principais testes realizados, destacam-se:

## Cadastro de novo usuário com dados válidos e inválidos, verificando validação de campos e mensagens de erro apropriadas;

## Autenticação de usuários com credenciais corretas e incorretas, garantindo o correto acesso ao sistema;

## Registro de doação associada a uma doadora, validando o vínculo com o banco de leite correspondente;

## Consulta de bancos de leite próximos utilizando geolocalização, avaliando precisão e retorno adequado de informações;

## Teste de responsividade, avaliando a interface do aplicativo em diferentes dispositivos e navegadores, assegurando usabilidade e consistência visual.

## Os casos de teste foram organizados em uma sequência lógica, de modo que a execução de cada teste dependesse do estado do sistema após os testes anteriores, garantindo coerência e confiabilidade nos resultados.

## 8.4 RESULTADOS DOS TESTES

Os testes realizados conforme este plano de teste evidencia que o sistema Donate apresentou bom desempenho e aderência aos requisitos levantados.

1. qualidade geral: satisfatória, com estabilidade observada em todos os fluxos principais;
2. conformidade: funcionalidades centrais de cadastro, doação e consulta de bancos de leite validadas com sucesso;
3. melhorias identificadas: ajustes finos na responsividade em telas menores e melhoria nas mensagens de erro exibidas ao usuário;
4. situação final: o sistema encontra-se funcional e apto para uso em ambiente controlado de testes, com recomendações de rodadas adicionais conforme evolução do projeto.

# 9 PLANO PARA IMPLANTAÇÃO

## A implantação do sistema Donate compreende o período em que as versões do software serão disponibilizadas para sua comunidade de usuários. Este plano descreve a estratégia de implantação considerando o atual estágio do projeto, que já foi apresentado na FAITEC e está em uso restrito pela equipe de desenvolvimento e professores orientadores.

## 9.1 METODOLOGIA

A implantação do sistema Donate seguirá uma metodologia faseada e iterativa, adaptada ao contexto acadêmico em que o projeto foi desenvolvido. A abordagem prioriza a validação progressiva, o aprendizado contínuo e o ajuste baseado em feedback real, assegurando que o sistema atenda às necessidades dos usuários antes de uma disponibilização mais ampla.

### 9.1.1 Descrição da Metodologia

A implantação do sistema Donate seguirá uma abordagem evolutiva e controlada, considerando que atualmente o sistema encontra-se em ambiente de desenvolvimento, com acesso restrito à equipe do projeto e professores orientadores. A metodologia será dividida em três fases principais:

1. Implantação experimental: Liberação para uso interno da instituição FAI, incluindo alunos e professores do curso de Enfermagem, para validação em cenário real controlado.
2. Implantação piloto: Estender o uso para profissionais de saúde do HCSL (Hospital das Clínicas Samuel Libânio) e doadoras voluntárias previamente cadastradas.
3. Disponibilização pública: Liberação gradual para a comunidade em geral, inicialmente na região de Santa Rita do Sapucaí e Pouso Alegre.

A sequência de implantação incluirá:

1. Configuração do ambiente de homologação;
2. Migração dos dados de desenvolvimento para produção;
3. Publicação das aplicações web em servidor acessível;
4. Capacitação dos usuários internos;
5. Coleta de feedback e ajustes finais.

### 9.1.2 Matriz de Responsabilidade

O Quadro 4 apresenta as atividades preparatórias para a implantação e as responsabilidades do cliente e dos fornecedores do sistema de software.

Continua.

|  |  |
| --- | --- |
| Atividades | Responsável |
| Planejamento | Equipe do projeto |
| Definição da equipe de implantação | Equipe do projeto |
| Levantamento de recursos necessários de hardware | Equipe do projeto |
| Levantamento de recursos necessários de software | Equipe do projeto |

Conclusão.

|  |  |
| --- | --- |
| Atividades | Responsável |
| Definição de estratégias para conversão e migração de dados (caso exista sistema legado) | Equipe do projeto |
| Programação dos treinamentos | Equipe do projeto |
| Preparação dos testes de aceitação | Equipe do projeto |
| Execução | Equipe do projeto |
| Configuração da infraestrutura de Tecnologia da Informação | Equipe do projeto |
| Instalação e ou configuração do ambiente | Equipe do projeto |
| Conversão e migração de base de dados (caso exista sistema legado) | Equipe do projeto |
| Treinamentos | Equipe do projeto |
| Realização de testes de aceitação | Equipe do projeto e usuários-chave |
| Avaliação | Eunice |
| Acompanhamento pós-implantação | Equipe do projeto |

QUADRO 4 – Papéis e responsabilidades na implantação

## 9.2 TREINAMENTO PREVISTO

O Quadro 5 detalha os treinamentos a serem ministrados para a capacitação dos usuários.

|  |  |  |
| --- | --- | --- |
| Treinamento | Conteúdo | Grupo de usuários |
| Uso do Sistema Donate - Fase Experimental | Cadastro de usuários, agendamento de coletas, consulta a postos de coleta, navegação no sistema | Alunos e professores da FAI |
| Capacitação para Profissionais de Saúde | Gestão de agendamentos, cadastro de bancos de leite, acompanhamento de doações, uso do painel administrativo | Profissionais do HCSL |
| Orientação para Doadoras | Como utilizar o sistema para registrar doações, agendar coletas e localizar postos de coleta | Doadoras voluntárias |

QUADRO 5 – Treinamentos previstos

## 9.3 CRONOGRAMA DE IMPLANTAÇÃO

No Quadro 6 são apresentadas as tarefas previstas durante a implantação, a duração em horas e o período de realização.

|  |  |  |
| --- | --- | --- |
| Tarefas | Duração | Período |
| Preparação do ambiente de homologação | 40 horas | 25/11 a 29/11/2025 |
| Migração do banco de dados | 8 horas | 02/12/2025 |
| Publicação da aplicação web | 8 horas | 03/12/2025 |
| Cadastro de usuários internos | 24 horas | 04/12 a 06/12/2025 |
| Treinamento dos usuários da FAI | 16 horas | 09/12 a 10/12/2025 |
| Coleta de feedback e ajustes | 40 horas | 11/12 a 15/12/2025 |
| Tempo estimado total: | 136 horas | |

QUADRO 6 – Cronograma de atividades da implantação

## 9.4 RECURSOS DE APOIO À IMPLANTAÇÃO

Os recursos oferecidos para apoiar o processo de implantação e posterior uso do sistema são listados no Quadro 7.

|  |  |
| --- | --- |
| Documento | Disponível em: |
| Manual de instalação e ou configuração | Apêndice K |
| Manuais do usuário | Apêndice K. |
| Vídeo demonstrativo | Apêndice K. |
| Outros | Apêndice K. |

QUADRO 7 – Recursos de apoio à implantação

## 9.5 VISÃO DA IMPLANTAÇÃO

Apresenta a arquitetura de implantação do sistema no ambiente real através do Diagrama de Implantação UML (Apêndice K). Ambiente:

1. Servidor: Samsung Server Solution com Windows Server, JSE, Apache Tomcat, App.war e PostgreSQL
2. Clientes: Notebook Samsung Galaxy Book 3 e Smartphone Motorola G60
3. Comunicação: HTTPS entre clientes e servidor

O diagrama no Apêndice K detalha a distribuição física dos componentes e suas conexões.

# 10 CONCLUSÃO

Na Fase 1 deste projeto, a equipe procurou realizar uma introdução ao tema, com pesquisas sobre a fundamentação teórica e definição dos objetivos e requisitos do sistema. Foi feita uma contextualização sobre a importância da doação de leite humano e os desafios enfrentados pelos bancos de leite, o que reforçou a relevância deste projeto. A revisão bibliográfica permitiu conhecer trabalhos semelhantes já realizados e ajudou a identificar as melhores práticas e soluções existentes. Com base nisso, foi possível estabelecer objetivos claros e levantar os requisitos funcionais, considerando com atenção as reais necessidades dos usuários.

Na Fase 2, o projeto avançou com a aplicação de correções e melhorias, além do aprofundamento na modelagem do sistema. A arquitetura da solução foi detalhada, incluindo os aspectos estruturais, de comportamento e de dados. Também foi feita a definição dos perfis de usuários, o que garantiu uma melhor adequação das funcionalidades às características de cada público atendido. Parte dos casos de uso considerados essenciais foi implementada e testada, e foram adotadas práticas para o controle de versão e a qualidade do código-fonte. Com isso, foi possível acompanhar de forma mais eficiente a evolução do sistema.

Na Fase 3, o sistema passou a um estágio mais avançado, com maior consolidação da integração entre os módulos e a realização do plano de testes. Foram aplicados testes unitários, de integração e de validação, que ajudaram a identificar pontos de melhoria e a corrigir inconsistências. A funcionalidade de geolocalização foi ajustada, oferecendo uma melhor experiência ao localizar os pontos de coleta e bancos de leite próximos. Também foram feitos ajustes na interface de usuário, principalmente em relação à responsividade em dispositivos móveis, tornando o sistema mais acessível. A equipe manteve a organização do código e o uso de versionamento, garantindo maior segurança e controle do desenvolvimento.

Na Fase 4, o projeto consolidou-se com a finalização dos aspectos gerenciais e técnicos necessários para a implantação. Foram realizadas correções pendentes das fases anteriores e elaborado o capítulo de Métodos Gerenciais, incorporando os artefatos de Gestão de Projetos II. Incluímos também a Análise de Complexidade Algorítmica (Seção 7.3), assegurando a eficiência das funcionalidades centrais. Além disso, foi estruturado o Plano para Implantação (Capítulo 9), contemplando a metodologia, matriz de responsabilidade, treinamentos, cronograma e recursos de apoio, bem como a Visão da Implantação com o diagrama de implantação UML. Por fim, as referências foram atualizadas e esta conclusão foi revisada, refletindo a evolução completa do projeto.

De modo geral, os objetivos iniciais foram alcançados, e o projeto avançou de forma sólida, com uma base bem estruturada tanto na parte teórica quanto na prática. As funcionalidades desenvolvidas buscam facilitar a interação entre doadoras, receptoras e bancos de leite, oferecendo uma solução inovadora e de fácil uso. Foram identificados, no entanto, pontos que ainda podem ser melhorados, como a ampliação de funcionalidades, novos testes em diferentes cenários e um trabalho contínuo de integração com recursos de geolocalização.

Além dos resultados técnicos, todo o processo trouxe aprendizados importantes. Foi possível perceber o quanto um bom planejamento e a definição clara dos requisitos fazem diferença no andamento do trabalho. A escuta ativa dos usuários, a realização de testes desde os primeiros estágios e a colaboração da equipe mostraram-se fundamentais para garantir um sistema mais funcional e centrado nas pessoas. Para eventuais próximas etapas, o foco será a implementação do plano de implantação elaborado, mantendo o sistema alinhado às necessidades reais do público e reforçando sua contribuição para a doação de leite humano.

# REFERÊNCIAS

BRASIL. **Conheça os benefícios da amamentação**. Brasília: Ministério da Saúde, 2023. Disponível em: <https://www.gov.br/saude/pt-br/campanhas-da-saude/2023/amamentacao/conheca-os-beneficios>. Acesso em: 13 mar. 2025.

BRASIL. **Indicadores de saúde:** mortalidade infantil. Brasília: Ministério da Saúde, 2025. Disponível em: <https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/m/mortalidade-infantil>. Acesso em: 13 mar. 2025.

BRASIL. Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD). Dispõe sobre o tratamento de dados pessoais, inclusive nos meios digitais, por pessoa natural ou por pessoa jurídica de direito público ou privado. **Diário Oficial [da] União**, Brasília, 2018. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm>. Acesso em: 01 maio 2025.

DISTRITO FEDERAL. Lei nº 7.711, de 11 de janeiro de 2025. Dispõe sobre a isenção da taxa de inscrição em concursos públicos para doadoras regulares de leite materno. **Diário Oficial do Distrito Federal**, Brasília, DF, 12 jan. 2025a.

DISTRITO FEDERAL. Secretaria de Estado da Economia. **Mulheres que doam leite materno terão isenção de taxa em concursos do GDF**. Brasília: 2025b. Disponível em: https://www.economia.df.gov.br. Acesso em: 29 jun. 2025.

DISTRITO FEDERAL. Secretaria de Estado de Saúde**. Banco de leite humano**: número de doadoras em 2024 e 2025. Brasília: 2025c.

HOSPITAL DAS CLÍNICAS SAMUEL LIBÂNIO (HCSL). **Posto de coleta de leite humano do HCSL auxilia bebês internados na UTI Neonatal e Pediátrica.** Pouso Alegre, MG: 2025. Disponível em: <https://www.hcsl.edu.br/posto-de-coleta-de-leite-humano-do-hcsl-auxilia-bebes-internados-na-uti-neonatal-e-pediatrica/>. Acesso em: 13 mar. 2025.

G1 SUL DE MINAS. Complexo Hospitalar Samuel Libânio reforça importância da doação de leite materno. **G1 Sul de Minas**, 26 mai. 2025. Disponível em: https://g1.globo.com/mg/sul-de-minas/especial-publicitario/fuvs/noticia/2025/05/26/complexo-hospitalar-samuel-libanio-reforca-importancia-da-doacao-de-leite-materno.ghtml. Acesso em: 29 jun. 2025.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). **Estatísticas do registro civil:** nascimentos. Brasília: IBGE, 2022. Disponível em: <https://www.ibge.gov.br/estatisticas/sociais/populacao/9110-estatisticas-do-registro-civil.html>. Acesso em: 13 mar. 2025.

LIMA, S. E. C. de. **Entrevista**. Santa Rita do Sapucaí: FAI, 2025.

LOPES, M. H.; SILVA, R. A.; PEREIRA, A. L.; SANTOS, F. M. A tecnologia dos aplicativos móveis na promoção ao aleitamento materno: revisão integrativa**. Revista Brasileira de Enfermagem**, 75(1), 2022. Disponível em: <https://www.researchgate.net/publication/360160864_A_tecnologia_dos_aplicativos_moveis_na_promocao_ao_aleitamento_materno_revisao_integrativa>. Acesso em: 13 mar. 2025.

OLIVEIRA, D. S.; SOUZA, A. I.; LIMA, T. H. Aleitamento materno: uso da tecnologia da informação como estratégia para a construção de um website. **Revista de Enfermagem da UFSM**, 11, 2021. Disponível em: <https://periodicos.ufsm.br/reufsm/article/view/64034/html>. Acesso em: 13 mar. 2025.

OLIVEIRA, D. S.; SOUZA, A. I.; LIMA, T. H. Tecnologias em saúde e suas contribuições para a promoção do aleitamento materno. **Ciência & Saúde Coletiva**, v. 26, n. 9, p. 1234–1245, 2019. Disponível em: <https://www.scielo.br/j/csc/a/RG9dKm34fMFyLFXpQswv7Rv/>. Acesso em: 13 mar. 2025.

ORGANIZAÇÃO MUNDIAL DA SAÚDE (OMS). **Breastfeeding**. 2025. Disponível em: <https://www.who.int/news-room/fact-sheets/detail/breastfeeding>. Acesso em: 13 mar. 2025.

REDE BRASILEIRA DE BANCOS DE LEITE HUMANO (RBLH-BR). **O que é a Rede Brasileira de Bancos de Leite Humano**. Rio de Janeiro: Fundação Oswaldo Cruz (Fiocruz), 2023. Disponível em: https://rblh.fiocruz.br/o-que-e-rede-blh. Acesso em: 13 mar. 2025.

SILVA, R. F.; ALMEIDA, K. L.; FERREIRA, P. H.; COSTA, M. A. Aplicativo Doe Leite: tecnologia facilitadora na doação de leite materno. **Revista da Escola de Enfermagem da USP**, 56, 2022. Disponível em: <https://www.scielo.br/j/ape/a/tXmsqRHbThMgSxqFZrzsYmJ/>. Acesso em: 13 mar. 2025.

SOMMERVILLE, I. **Engenharia de Software**. 10. ed. São Paulo: Pearson Education, 2019.

**ODS BRASIL**, 2025 Disponível em:

https://odsbrasil.gov.br/objetivo/objetivo?n=3. Acesso em: 22 set. 2025.

OBRAS CONSULTADAS

ALMEIDA, R. M. **Modelagem de banco de dados relacional**. 4. ed. São Paulo: Érica, 2021.

ARAÚJO, M. P. **Desenvolvimento Web com Node.j**s. 2. ed. Rio de Janeiro: Novatec, 2020.

BEZERRA, E. **Princípios de desenvolvimento ágil**. São Paulo: Casa do Código, 2021.

FERREIRA, A. M. **Banco de dados***: teoria e prática*. 3. ed. Porto Alegre: Bookman, 2019.

NASCIMENTO, J. R. **PostgreSQL**: guia do administrador de banco de dados. 2. ed. São Paulo: Novatec, 2022.

SANTOS, L. G. dos. **Aplicações móveis***:* design e usabilidade. Brasília: SENAC, 2020.

TORRES, G. R. **Angular**: do básico ao avançado. Florianópolis: eBook, 2022.

# APÊNDICE A – GERENCIAMENTO DO PROJETO

O plano de elaboração e gerenciamento do projeto está disponível na pasta “ApêndiceA” que acompanha este documento.

# APÊNDICE B – RELATÓRIO DE DESEMPENHO

Os relatórios de desempenho estão disponíveis na pasta “ApêndiceB” que acompanha este documento.

# APÊNDICE C – VISÃO FUNCIONAL

O modelo de casos de uso está disponível na pasta “ApêndiceC” que acompanha este documento.

# APÊNDICE D – VISÃO DOS DADOS

O modelo conceitual dos dados (representado por meio do modelo entidade-relacionamento), o modelo lógico dos dados e o dicionário dos dados do modelo lógico estão disponíveis na pasta “ApêndiceD” que acompanha este documento.

# APÊNDICE E – MODELO INICIAL DA INTERFACE DE USUÁRIO

Os modelos de interface de usuário de baixa e média fidelidades estão disponíveis na pasta “ApêndiceE” que acompanha este documento.

# APÊNDICE F – VISÕES ESTRUTURAL E COMPORTAMENTAL

Os diagramas elaborados por meio da UML estão disponíveis na pasta “ApêndiceF” que acompanha este documento.

# APÊNDICE G – ENTREVISTAS COM USUÁRIOS

As entrevistas com os potenciais usuários do sistema encontram-se disponíveis na pasta “ApêndiceG” que acompanha este documento.