



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I
CENTRO DE CIÊNCIA E TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

RAQUEL GOMES DE VASCONCELOS DA SILVEIRA

**DESENVOLVIMENTO FRONT-END PARA
APRIMORAMENTO DA EXPERIÊNCIA DO USUÁRIO EM
UM APLICATIVO DE DOAÇÃO DE SANGUE**

**CAMPINA GRANDE
2025**

RAQUEL GOMES DE VASCONCELOS DA SILVEIRA

DESENVOLVIMENTO FRONT-END PARA
APRIMORAMENTO DA EXPERIÊNCIA DO USUÁRIO EM
UM APLICATIVO DE DOAÇÃO DE SANGUE

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciência e Tecnologia da Universidade Estadual da Paraíba como requisito parcial à obtenção do título de Bacharelado em Ciência da Computação.

Área de concentração: Usabilidade e Fatores Humanos

Orientador: Prof. Dr. Daniel Scherer

CAMPINA GRANDE

2025

É expressamente proibida a comercialização deste documento, tanto em versão impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que, na reprodução, figure a identificação do autor, título, instituição e ano do trabalho.

S587d Silveira, Raquel Gomes de Vasconcelos da.
Desenvolvimento front-end para aprimoramento da experiência do usuário em um aplicativo de doação de sangue [manuscrito] / Raquel Gomes de Vasconcelos da Silveira. - 2025.
104 f. : il. color.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Ciência da computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2025.

"Orientação : Prof. Dr. Daniel Scherer, Departamento de Computação - CCT".

1. Aplicativo tecnológico. 2. Front-end. 3. Usabilidade e fatores humanos. 4. Desenvolvimento de software. 6. Usabilidade e fatores humanos. I. Título

21. ed. CDD 005.3

RAQUEL GOMES DE VASCONCELOS DA SILVEIRA

DESENVOLVIMENTO FRONT-END PARA APRIMORAMENTO DA EXPERIÊNCIA
DO USUÁRIO EM UM APLICATIVO DE DOAÇÃO DE SANGUE

Trabalho de Conclusão de Curso
apresentado à Coordenação do Curso
de Ciência da Computação da
Universidade Estadual da Paraíba,
como requisito parcial à obtenção do
título de Bacharela em Computação

Aprovada em: 12/06/2025.

BANCA EXAMINADORA

Documento assinado eletronicamente por:

- **Katia Elizabete Galdino** (***.461.834-**), em **16/06/2025 21:08:45** com chave **3c1ff36e4b0f11f0907e1a1c3150b54b**.
- **Daniel Scherer** (***.694.200-**), em **17/06/2025 12:48:12** com chave **79a7b36a4b9211f0a9f32618257239a1**.
- **Wellington Candeia de Araujo** (***.655.074-**), em **16/06/2025 19:12:18** com chave **f7c8d2a44afe11f09f501a1c3150b54b**.

Documento emitido pelo SUAP. Para comprovar sua autenticidade, faça a leitura do QrCode ao lado ou acesse https://suap.uepb.edu.br/comum/autenticar_documento/ e informe os dados a seguir.

Tipo de Documento: Folha de Aprovação do Projeto Final

Data da Emissão: 20/06/2025

Código de Autenticação: fdfc80



AGRADECIMENTOS

Agradeço primeiramente à Deus por sempre ter guiado o meu caminho e as minhas decisões.

Aos meus pais José Edson Ramos da Silveira e Valquíria Gomes de Vasconcelos da Silveira pela fé, educação e ensinamentos.

Aos meus irmãos e minha família pelo apoio e ajuda. Ao meu marido, Arthur Souza, pela ajuda, disponibilidade, apoio e conselho.

Ao meu orientador Daniel Scherer, pela paciência, disponibilidade em ajudar nas correções, e ensinamento.

RESUMO

Este trabalho é apresentado o projeto de um aplicativo de doação de sangue, em que possui o objetivo de simplificar e tornar eficiente o agendamento e a triagem presencial, abordando as problemáticas de carência das instituições que adquirem esses recursos e visa atender suas necessidades, bem como dos doadores e interessados em contribuir para a manutenção de estoques saudáveis. A estrutura do projeto envolve a sequência de definição de requisitos, seleção de tecnologias, modelagem do sistema, que foi construída a partir da criação de diagramas de caso de uso, de atividade, sequência e histórias do usuário, em que descrevem o fluxo que o sistema terá e a interação entre o usuário e o sistema será feita. Após isso, a implementação surgiu com a construção do software com as tecnologias Angular e Ionic, em que oferecem uma aplicação de multiplataformas usando um único código. Proporcionando assim, uma abordagem eficiente e escalável para o desenvolvimento da interface do usuário. A escolha dessas ferramentas visa garantir uma experiência de usuário fluída e responsiva, crucial para uma interação intuitiva. Após a finalização do software, espera-se otimizar o processo de doação de sangue, incentivando a participação ativa da sociedade e garantindo o estoque adequado de sangue para as instituições. Este trabalho representa um passo significativo na convergência da tecnologia e solidariedade, com o objetivo de encontrar soluções inovadoras, como: uso de frameworks que funcionam em diferentes dispositivos e sistemas operacionais sem necessitar qualquer alteração e a inserção de funcionalidades como o histórico de doações, que possibilita ao usuário doador ter informações sobre seu processo de doação de sangue e atender a demanda de carências de bolsas de sangues nas instituições às instituições que emitiram o alerta.

Palavras-chave: Aplicativo para doação de sangue; Front-end; TypeScript; Angular; Ionic.

ABSTRACT

This work presents the project of a application with front-end development, with the objective of simplifying and making the blood donation process efficient, solving the problems of lack of institutions that acquire these resources and aiming to meet their needs, as well as well as donors and those interested in contributing to the maintenance of healthy stocks. The project structure involves the sequence of requirements definition, technology selection, system modeling and implementation, with the construction of the application through code. The technological base used is TypeScript, the Angular and Ionic frameworks and other front-end tools. Thus providing an efficient and scalable approach to user interface development. The choice of these tools aims to ensure a fluid and responsive user experience, crucial for intuitive interaction with the application. After the completion of the application, it is expected to optimize the blood donation process, encouraging the active participation of society and ensuring an adequate blood supply for institutions. This work represents a significant step in the convergence of technology and solidarity, with the aim of finding innovative solutions in the area of blood donation.

Key-words: Blood donation application; Front-end; TypeScript; Angular; Ionic.

LISTA DE ILUSTRAÇÕES

Figura 1 – Sistemas operacionais <i>mobile</i> mais utilizados no Brasil.	14
Figura 2 – Frameworks web mais utilizados no mundo.	17
Figura 3 – Mapa HemorRede do Rio de Janeiro.	20
Figura 4 – Tela de Início - Hemobrás.	21
Figura 5 – Tela de login - Hemovida.	22
Figura 6 – Tela de login - Hemogram.	23
Figura 7 – Tela de login - Um só sangue.	24
Figura 8 – Etapas do Processo de Desenvolvimento.	26
Figura 9 – Versão Android.	30
Figura 10 – Versão IOS.	30
Figura 11 – Diagrama de casos de uso (gerenciamento administrativo).	31
Figura 12 – Diagrama de casos de uso (funcionalidades do usuário).	32
Figura 13 – Diagramas de Atividade de Responder Questionário do Doador.	33
Figura 14 – Diagrama de Atividade de Buscar Instituições de Doação de Sangue.	34
Figura 15 – Diagrama de Atividade de Emitir Alerta.	35
Figura 16 – Diagrama de Responder Questionário.	36
Figura 17 – Diagrama de Sequência de Buscar Instituições.	37
Figura 18 – Tela de login - <i>mobile</i>	41
Figura 19 – Tela de menu - <i>mobile</i>	42
Figura 20 – Tela de instituições - <i>mobile</i>	43
Figura 21 – Tela de instituições - parte 02 - <i>mobile</i>	44
Figura 22 – Tela de buscar instituições - parte 03 - <i>mobile</i>	45
Figura 23 – Tela de agendamento - <i>mobile</i>	46
Figura 24 – Tela de agendamento parte 02 - <i>mobile</i>	47
Figura 25 – Tela de agendamento parte 03 - <i>mobile</i>	48
Figura 26 – Tela de notificação - <i>mobile</i>	49
Figura 27 – Tela de Login - <i>web</i>	50
Figura 28 – Tela de instituições - <i>web</i>	51
Figura 29 – Tela de visualizar agendamento das instituições - <i>web</i>	52
Figura 30 – Tela de visualizar histórico de agendamento - <i>web</i>	53
Figura 31 – Tela de emitir alerta - <i>web</i>	54
Figura 32 – Tela de home - <i>web</i>	55

LISTA DE TABELAS

Tabela 1 – Comparação do Android e o IOS.	15
Tabela 2 – Comparativo dos trabalhos relacionados.	25
Tabela 3 – Comparação entre <i>frameworks</i>	29

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Definição do problema	11
1.2	Objetivos	11
1.2.1	<i>Objetivos específicos</i>	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	Aplicativos móveis	13
2.2	Sistema operacional móveis	13
2.3	Sistema operacional Android	14
2.4	Sistema Operacional IOS	15
2.5	Front-end	16
2.6	Framework	16
2.6.1	<i>Angular</i>	17
2.6.2	<i>Ionic</i>	18
2.6.3	<i>Bootstrap</i>	18
2.6.4	<i>Apache Cordova</i>	18
2.7	HTML e CSS	19
2.8	Typescript	19
3	REVISÃO BIBLIOGRÁFICA	20
3.1	HemorRede	20
3.2	HemoBrás	21
3.3	HemoVida(ConectSUS)	21
3.4	Hemogram	22
3.5	Um Só Sangue	23
3.6	Comparativo dos Trabalhos Relacionados	24
4	METODOLOGIA	26
4.1	Definição dos Requisitos	26
4.2	Seleção de Tecnologias	27
4.3	Modelagem do Sistema	27
4.4	Implementação	27
5	RESULTADOS	28
5.1	Descrição do sistema	28
5.2	Definição do <i>Framework Web e mobile</i>	28
5.3	Diagramas de casos de usos	31
5.4	Histórias do usuário	37
5.4.1	<i>História de Usuário - Login</i>	38
5.4.2	<i>História de Usuário - Enviar Questionário do Doador</i>	38

5.4.3	<i>História de Usuário - Responder Questionário do Doador . . .</i>	38
5.4.4	<i>História de Usuário - Buscar Instituições de Doação</i>	39
5.4.5	<i>História de Usuário - Agendar Entrevista</i>	39
5.4.6	<i>História de Usuário - Visualizar Agendamentos</i>	39
5.4.7	<i>História de Usuário - Emitir Alerta</i>	40
5.5	Apresentação da Aplicação mobile e Web	40
5.5.1	Aplicação mobile	40
5.5.1.1	<i>Tela de Login</i>	40
5.5.1.2	<i>Tela de Menu</i>	41
5.5.1.3	<i>Tela de instituições</i>	42
5.5.1.4	<i>Tela de agendamento</i>	45
5.5.1.5	<i>Tela de notificação</i>	48
5.5.2	Aplicação web	49
5.5.2.1	<i>Tela de Login</i>	49
5.5.2.2	<i>Tela de instituições</i>	50
5.5.2.3	<i>Tela de agendamento</i>	51
5.5.2.4	<i>Tela de emitir alerta</i>	54
6	CONSIDERAÇÕES FINAIS	57
6.1	Trabalhos futuros	58
	REFERÊNCIAS	59
	APÊNDICE A – CÓDIGOS	62

1 INTRODUÇÃO

A tecnologia móvel é um avanço tecnológico que permite ao usuário utilizar várias ferramentas em um único aparelho, ter acesso à informação de forma rápida e eficaz, desempenhando, assim, a função de um computador de bolso. Portanto, o desenvolvimento desses smartphones revolucionaram a comunicação móvel e sua aplicabilidade na sociedade.

De acordo com um estudo realizado pela FGV em 2023, o Brasil registrou a marca de 249 milhões de smartphones, o que equivale a mais de dois dispositivos digitais por habitante. Entre os 149 milhões de usuários de internet no país, 92 milhões acessam exclusivamente por meio de seus celulares (FGV, 2023). Esses números destacam a preferência predominante dos brasileiros por se comunicar e buscar informações por meio da tecnologia móvel.

O smartphone oferece praticidade, agilidade na comunicação e transferência de informações, atualmente também é possível encontrar diversas funcionalidades que ultrapassam as fronteiras de apenas realizar/receber ligações e mensagens. O usuário pode acessar a calculadora, editar um documento online, encontrar parentes de longa distância pelas redes sociais e marcar/agendar consultas em um mesmo aparelho (LIMA et al., 2023).

Além de oferecer diversas possibilidades, na área da saúde, também é reconhecido como um grande facilitador. Sendo capaz de reduzir a distância de entre o paciente e as instituições de saúde, bem como, os profissionais encarregados de atendê-los. A evolução das Tecnologias da Informação permitiu a criação de diversas intersecções nos campos da saúde, tecnologia e comunicação. Concretizando assim, desenvolvimento de aplicativos exclusivo para esses smartphones (LEITE; ROSA, 2017).

A mobile health (mHealth) pode ser descrita por práticas médicas e de saúde pública que utilizam aparelhos de monitoramento de pacientes, celulares e outros aparelhos sem fio. Utilizam a tecnologia móvel para serviços multimídia (MMS) e de mensagens curtas (SMS), assim como para aplicativos mais sofisticados, que contenham telecomunicações móveis de terceira, quarta e quinta geração (3G, 4G e 5g), tecnologia Bluetooth e sistemas de localização (WATTANAPISIT et al., 2020).

Por isso, o aplicativo Vida+, que visa estreitar a comunicação entre instituições de coleta de sangue, hospitais e doadores de sangue e medula óssea. O aplicativo oferece funcionalidades como a teletriagem, informações sobre elegibilidade para doação, alerta de carências de suprimentos, buscas de instituições de doação de sangue próximas, agendamento online. Além disso, os usuários podem acessar seus históricos de doações e agendar compromissos de acordo com a disponibilidade das instituições selecionadas. As

instituições, por sua vez, podem visualizar os agendamentos e emitir alertas para mobilizar doações de sangue e medula óssea.

1.1 Definição do problema

O aumento de acidentes, cirurgias e doenças teve por consequência o crescimento da demanda mundial por sangue. A necessidade é imediata e alarmante de transfusões de sangue em comparação a restrição ao suprimento de sangue. O desafio dos hemocentros consiste em recrutar e reter doadores voluntários qualificados o recrutamento e a retenção de doadores voluntários qualificados (SARDI et al., 2019).

No Brasil, muitos pacientes em hospitais brasileiros e em outros países acabam indo à óbito por falta do tipo sanguíneo solicitado não estar disponível ou em carência isso se dá devido aos bancos de estoques não conseguirem suprir a grande demanda requerida. Decorrendo na priorização de pacientes mais urgentes na fila (ALEXANDRE et al., 2023).

Quatorze em cada mil brasileiros doaram sangue regularmente em 2023, o que caracteriza 1,4 por cento da população, e na Paraíba, foram registrados 87 mil doadores no ano de 2022, segundo o Ministério da Saúde. Apesar desse número estar dentro da regulação estabelecida pela OMS, ainda é número alarmante. A doação de sangue e os benefícios que ela traz à sociedade não são popularmente conhecidos. Além de salvar vidas, o doador recebe alguns privilégios decorrentes à sua escolha. Um dos benefícios, é a declaração do banco de estoque que o doador se fideliza, possibilitando a isenção em taxas de concursos públicos e ausência justificada no serviço público no dia em que a doação é feita. Além de ter acesso à resultados dos exames para patologias hemotransmissíveis (Ministério da Saúde, 2023).

Uma das alternativas que podem ajudar a sanar essa problemática que envolve as etapas do processo de doação de sangue é a teletriagem (DE, 2017). É um termo que se refere ao processo de triagem realizado de forma remota ou virtual, geralmente por meio de tecnologias de comunicação como aplicativos online. A teletriagem oferece as seguintes vantagens, acessibilidade, eficiência de tempo, identificação de elegibilidade e orientação pré-doação (DE, 2017). Dessa forma, é proposto nesse trabalho o desenvolvimento de um novo aplicativo que forneçam ferramentas que visam simplificar as etapas de doação de sangue.

1.2 Objetivos

Este trabalho tem como objetivo desenvolver uma aplicação web e mobile para diferentes sistemas operacionais. Com a finalidade de realizar agendamentos de doação de sangue online e mostrar alertas de tipos sanguíneos em carência no banco de estoque da cidade de Campina Grande, na Paraíba.

1.2.1 *Objetivos específicos*

- Facilitar a adesão de um futuro doador ao sistema de doação de sangue por meio do agendamento online e teletriagem.
- Disponibilizar alertas aos doadores aptos quando o banco de estoque estiver com carência de bolsas de sangue.
- Aplicar princípios de engenharia de software para garantir a aplicabilidade e manutenibilidade da aplicação.
- Assegurar a compatibilidade e performance da aplicação mobile em ambientes iOS e Android.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os conceitos fundamentais necessários para uma compreensão abrangente do presente trabalho. Assim, são apresentados os aplicativos móveis, os sistemas operacionais, o que é *front-end*, os *framework* e as linguagens de programação utilizadas para o desenvolvimento *front-end* da aplicação.

2.1 Aplicativos móveis

Aplicativo Móvel é um programa projetado especificamente para dispositivos como *tablets*, *smartphones*, *smartwatches*, computadores e *smart TVs*. Ademais, alguns são incluídos nos dispositivos de fábrica ou podem ser baixados posteriormente em lojas diversas de aplicativos pelo usuário. Assim, ele pode ter sua especificação e restrição dependendo da forma que foi desenvolvido. Possuem instalações pagas, como também, outras são disponibilizadas gratuitamente.

Esses *softwares* tem a finalidade de facilitar a rotina do usuário, onde o acesso à informação e comunicação cabem no bolso. Além disso, oferecem funcionalidades que englobam as necessidades frequentes da sociedade, sendo essas: agendamento de consultas médicas, reuniões online, comunicação por chat em redes sociais, edição de documentos, fotos e vídeos, traduções de texto, compras online, entre outras.

2.2 Sistema operacional móveis

Sistemas operacionais são softwares complexos e responsáveis por gerenciar componentes fundamentais para funcionamento de um computador. Ademais, fornece uma interface gráfica que facilita sua interação com hardware (TANENBAUM, 2008).

Além disso, é incluído os Sistemas Operacionais para Dispositivos móveis. Dessa forma, recebem essa classificação por serem executados em aparelhos como *smartphones* e *tablets*.

Os dispositivos móveis oferecem novas formas de interação com o usuário (sensores, GPS, teclados etc) tornando seu desenvolvimento totalmente diferente de projetar aplicações para computadores (WASSERMAN, 2010).

Os principais sistemas operacionais móveis são : Android, IOS, Windows Mobile e BlackBerry. Nesse contexto , IOS e Android são as principais plataformas de mercado (GOADRICH; ROGERS, 2011). A seguir são apresentados os sistemas operacionais Android e IOS, bem como a definição das suas características.

2.3 Sistema operacional Android

O Android é um sistema operacional móvel do Google e líder mundial nesse segmento. No Google I/O 2017 foi anunciado que já existem mais de 2 bilhões de dispositivos Android ativados no mundo.

Está disponível para diversas plataformas, como smartphones e *tablets*, TV (Google TV), relógios (Android Wear), óculos (Google Glass), carros (Android Auto), e é o sistema operacional móvel mais utilizado no mundo (LECHETA, 2017).

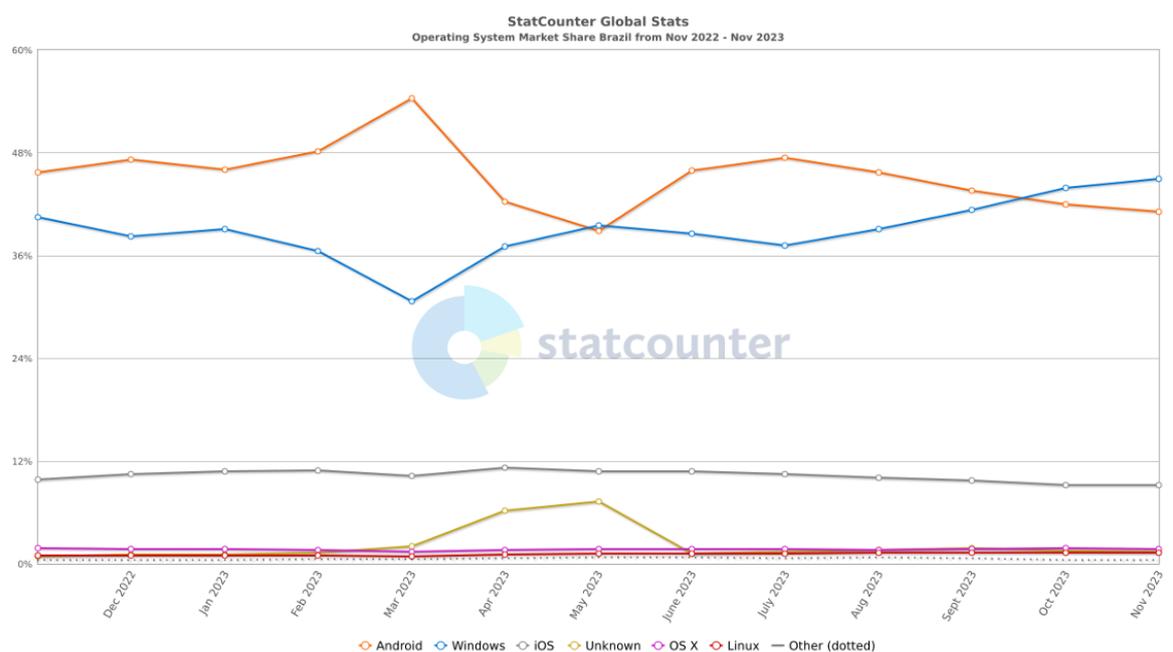
O sistema operacional é de código-fonte aberto e gratuito. Isso permite ver o código-fonte do Android e como seus recursos são implementados (DEITEL et al., 2013).

Atualmente é mantido pela Open Handset Alliance (OHA). As aplicações desenvolvidas para Android foram criadas com a linguagem Java, o que facilita muitos programadores com conhecimentos em Java a desenvolver aplicações para a plataforma.

Um dos benefícios do Android é a permitir que o usuário associe sua conta do google ao dispositivo e obtenha acesso aos seus dados nos aplicativos padrões do sistema. O funcionamento como qualquer outro sistema operacional para dispositivos móveis, ofertando funções de gerenciamento de processos dos aplicativos e manipulação de recursos de hardware presentes no dispositivo.

De acordo com Statcounter (2023), o Android é o sistema operacional mais utilizado atualmente no Brasil em aparelhos móveis, como apresentado na Figura 1, e iOS o terceiro mais utilizado.

Figura 1 – Sistemas operacionais *mobile* mais utilizados no Brasil.



Fonte: Adaptado da (Statcounter, 2023).

2.4 Sistema Operacional IOS

IOS é a abreviatura para iPhone Operation System sendo desenvolvido pela Apple, foi baseado no Sistema Operacional MAC OS X. O sistema realiza uma abstração entre a comunicação do hardware com o aplicativo (Apple Inc, 2023).

O sistema operacional iOS foi desenvolvido com a linguagem de programação Objective-C. O iOS é a plataforma utilizada em produtos Apple como o iPhone, iPad e iPod. Assim, oferece recursos que objetivam atender critérios específicos de um dispositivo móvel da Apple, visando a entrada e a saída de dados armazenados.

O iOS possui alguns aplicativos nativos, como aplicativo de Safari, iBook, Mapas, Notas, assistente de voz « Siri », FaceTime, Livros, Contatos, serviço de nuvem « iCloud » e etc. O usuário tem a opção de baixar novos aplicativos para o sistema através da loja virtual de aplicativos exclusivos da Apple, a App Store (Apple, 2023).

Sendo assim, na Tabela 1 é apresentado um comparativo dos sistemas Android e IOS.

Tabela 1 – Comparação do Android e o IOS.

Características	Android	IOS
Liberdade de desenvolvimento	x	
Emulador e debugador no SDK	x	x
Facilidade de portabilidade		x
Código Aberto	x	
Construtor de interface	x	x
IDE	Eclipse, Netbens, etc.	XCode
Linguagem de programação	Java com XML	Ojective-C
Nível de Fragmentação	Alto	Baixo
Nível de segurança	Baixo	Alto

Fonte: Adaptado da (MENDONÇA; BITTAR; DIAS, 2016).

A partir do sistemas operacionais escolhidos é necessário entender sobre o funcionamento do front-end.

2.5 Front-end

O Front-end desempenha o papel visual de uma aplicação. Em essência, é responsável por tudo o que os usuários experienciam durante a interação. Abrangendo desde a arquitetura até a programação da interface gráfica de um sistema, o Front-end é a estrutura que possibilita a comunicação direta com o usuário.

O propósito primordial do Front-end é proporcionar aos usuários uma experiência interativa e intuitiva ao utilizar uma aplicação. O desenvolvimento Front-end é interpretado pelos navegadores ou sistemas operacionais, tornando eficaz a exibição dos elementos visuais aos usuários que desejam interagir com eles (Kenzie, 2022).

Diversas linguagens desempenham papéis vitais na formatação visual, sendo as mais proeminentes neste trabalho, o HTML, CSS e TypeScript. Cada uma dessas linguagens contribui com funcionalidades distintas, porém possuem a mesma finalidade quando aplicadas no contexto do Front-end. Em seguida, é necessário definir os frameworks.

2.6 Framework

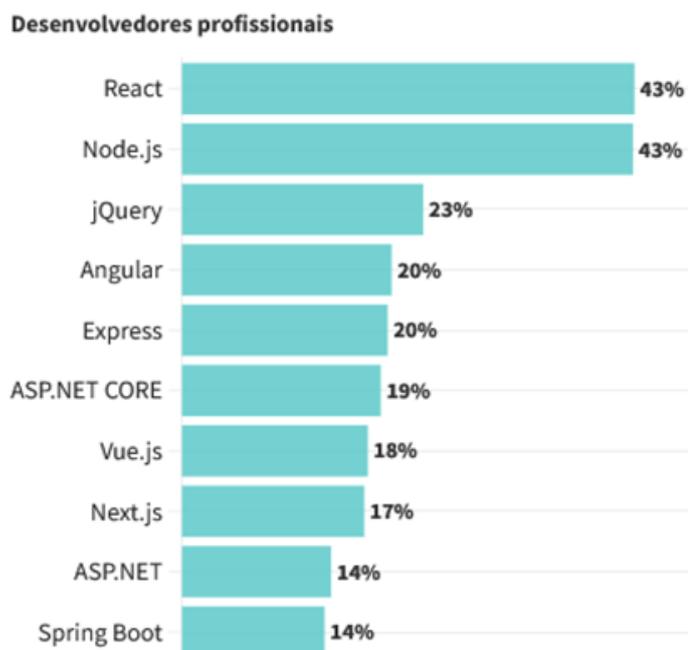
Os frameworks surgiram como métodos preestabelecidos, com o objetivo de auxiliar e direcionar a execução de ações, atendendo qualquer requisição solicitada, desde que a tecnologia seja apontada para a função desejada. projetando a resolução de problemas recorrentes com uma abordagem genérica. Dessa forma, é desnecessário a reescrita de softwares, focando seus esforços para solucionar contratempos. Além disso, auxiliam na conclusão rápida e segura de aplicações (Hostgator, 2021).

Dessa maneira, os frameworks proporcionam a vantagem de agilizar tarefas, devido realizar a padronização de projetos e configurações, ofertando aos profissionais somente a prática do desenvolvimento. Ademais, atendem qualquer requisição solicitada, desde que a tecnologia selecionada Assim, por ser um trabalho colaborativo, mais soluções são implementadas à ferramenta (BITTENCOURT; MURILO, 2021).

Os frameworks de desenvolvimento *mobile* multiplataforma permitirão uma construção híbrida, possibilitando um único código ser executado em vários sistemas operacionais móveis, como Android, Windows Phone, iOS e BlackBerry OS. Nesse sentido, isso ocorre devido a compatibilidade de arquitetura entre eles, tornando estas aplicações mais práticas e econômicas para o mercado.

Por outro lado, os frameworks web possuem características semelhantes ao mobile sendo populares entre os profissionais da área de desenvolvimento. Em 2023, os mais utilizados de acordo com Stack Overflow Developer Survey 2023 foram, React, Node.js, jQuery, Angular Express e etc. Como pode ser observado na Figura 2 (Stack Overflow, 2023).

Figura 2 – Frameworks web mais utilizados no mundo.



Fonte: Adaptado da (Stack Overflow, 2023).

Após a análise do estudo acerca dos frameworks web, foram selecionados os que se adequavam à etapa de requisitos. Dessa forma, para a escolha foi levado em consideração, a arquitetura do sistema a ser desenvolvido, a facilidade de implementação e a compatibilidade com o sistemas operacionais convencionais. Sendo assim, os frameworks que se destacaram foram, Ionic e Angular, que são abordados neste capítulo.

2.6.1 *Angular*

Angular, é um framework front-end de código-aberto do JavaScript que promove produtividade na experiência do desenvolvimento web. Dessa forma, foi criado por Miško Hevery e Adam Abron em 2009, sua primeira versão intitulada AngulasJS, foi lançada 2010 mas reescrita em 2006 sendo nomeada Angular 2. Assim, sua arquitetura é formada por blocos que são inseridos na medida em que são requisitados (Angular, 2023).

Por conseguinte, é utilizado para construir Single-Page Applications (SPA) aplicativos para web constituídos por uma única página dinâmica, semelhantes a de um aplicativo desktop. Nesse contexto, os desenvolvedores conseguem projetar menus animados para páginas de internet advindas de HTML.

O desenvolvimento da aplicação web é mais flexível devido a arquitetura do Angular ser baseada em componentes, que funcionando como blocos de construção, sendo inseridos na medida em que são requisitados. Devido a isso, o código pode ser reutilizado sem a necessidade de ser reescrito.

2.6.2 Ionic

O SDK, do inglês Software Development Kit, traduzido com Kit de Desenvolvimento de Software Ionic, concebido por Max Lynch, Ben Sperry e Adam Bradley da Drifty Co. em 2013, apresentou sua versão inicial no mesmo ano, baseando-se em AngularJS e Apache Cordova. Funcionando como um kit de ferramentas de interface do usuário, o Ionic permite a criação eficiente de aplicativos móveis e de desktop com alto desempenho, utilizando tecnologias web como HTML, CSS e JavaScript. Suportando integrações com estruturas renomadas como Angular, React e Vue (Ionic, 2023), destaca-se por sua ênfase na interação UX e na interface do usuário, abrangendo desde controles até animações e gestos.

A acessibilidade e facilidade de aprendizado são atributos marcantes do Ionic, facilitando a integração com outras bibliotecas ou estruturas, como Angular, React ou Vue. Além disso, sua versatilidade permite a utilização independente, dispensando frameworks de front-end através de um simples script.

O Ionic, enquanto framework de desenvolvimento híbrido, oferece uma vantagem significativa ao possibilitar o uso de um único código para diversas plataformas, incluindo Android, iOS e Windows Phone. Essa abordagem proporciona benefícios como simplificação do desenvolvimento, redução de custos e maior praticidade na manutenção, estabelecendo-se como uma solução abrangente para aplicações multiplataforma.

2.6.3 Bootstrap

O Bootstrap, um framework front-end destinado ao desenvolvimento de aplicações web e sites mobile-first, destaca-se pela sua capacidade de adaptação ao layout do dispositivo do usuário. Essa característica proporciona aos desenvolvedores um conjunto de sintaxes específicas que agilizam a criação de sites, eliminando a necessidade de se preocupar com comandos básicos ou funções adicionais (ROVEDA, 2022).

Nesse sentido, os engenheiros Jacob Thorton e Mark Otto desenvolveram o "Bootstrap" com o propósito inicial de resolver incompatibilidades internas na equipe. O objetivo era adotar uma estrutura única, criando condições mais propícias e reduzindo inconsistências decorrentes das diversas abordagens de codificação adotadas por diferentes profissionais (ROVEDA, 2022).

2.6.4 Apache Cordova

Cordova é uma estrutura de desenvolvimento móvel de código aberto que oferece suporte à criação de aplicativos multiplataforma, utilizando tecnologias padrão da Web, tais como HTML5, CSS3 e JavaScript (Apache Cordova, 2023). Ao optar por Cordova, é possível evitar a necessidade de utilizar a linguagem de desenvolvimento nativa de cada plataforma móvel.

Assim como o Ionic, Cordova é um framework destinado à construção de aplicativos móveis híbridos. Ionic combina as funcionalidades do Apache Cordova com o Angular, simplificando o processo de desenvolvimento. Dessa maneira, é possível aproveitar todas as ferramentas oferecidas pelo Angular para criar o aplicativo, enquanto o Apache Cordova fornece a base para a execução multiplataforma. Em seguida, são apresentadas as linguagens de estruturação de páginas web e de construção de layout, HTML e CSS, respectivamente.

2.7 HTML e CSS

HTML, é definida por HiperText Markup Language, sendo traduzida como Linguagem de Marcação de Hipertexto, é o componente fundamental do desenvolvimento web, permitindo a inserção de conteúdo e a definição da estrutura básica de um website. Esse elemento desempenha um papel crucial ao organizar as informações em uma página online. Essencial para o navegador interpretar textos como elementos e carregar diversos conteúdos.

Por outro lado, CSS, ou Cascading Style Sheets, funciona como uma camada de personalização ao código gerado por linguagens como HTML, XML ou XHTML. Ao oferecer uma abordagem prática, o CSS possibilita alterações rápidas no layout, incluindo definições de cores e fontes. Além de facilitar a personalização, essa camada contribui para a redução da repetição de conteúdo na estrutura do código, proporcionando uma experiência mais eficiente e fluida na apresentação visual. Com a linguagem de programação estruturada faz-se necessário a utilização de uma linguagem de programação orientada a objetos, que é o Typescript.

2.8 Typescript

O Typescript é um pré-processador de código-aberto do JavaScript, foi criado por Anders Hejlsberg e mantido pela Microsoft. Dessa forma, colabora no desenvolvimento de código considerando as fundamentações de orientação a objetos, como classes, tipagens e interfaces no Javascript (ADRIANO, 2021).

O código Typescript somente é utilizado em desenvolvimento, já que é totalmente convertido para Javascript no pré-processamento de codificação. No entanto, podem ser alguns benefícios se destacam, sendo elas, usar funcionalidades que não são nativas, a eficiência em descobrir erros durante a produção, conhecimento prévio de formato de parâmetros, métodos e valores que poderão ser usados e requisitados, entre outros (FERNANDES, 2019). No próximo capítulo, revisão bibliográfica, são apresentados alguns dos aplicativos semelhantes para esse tipo de aplicação.

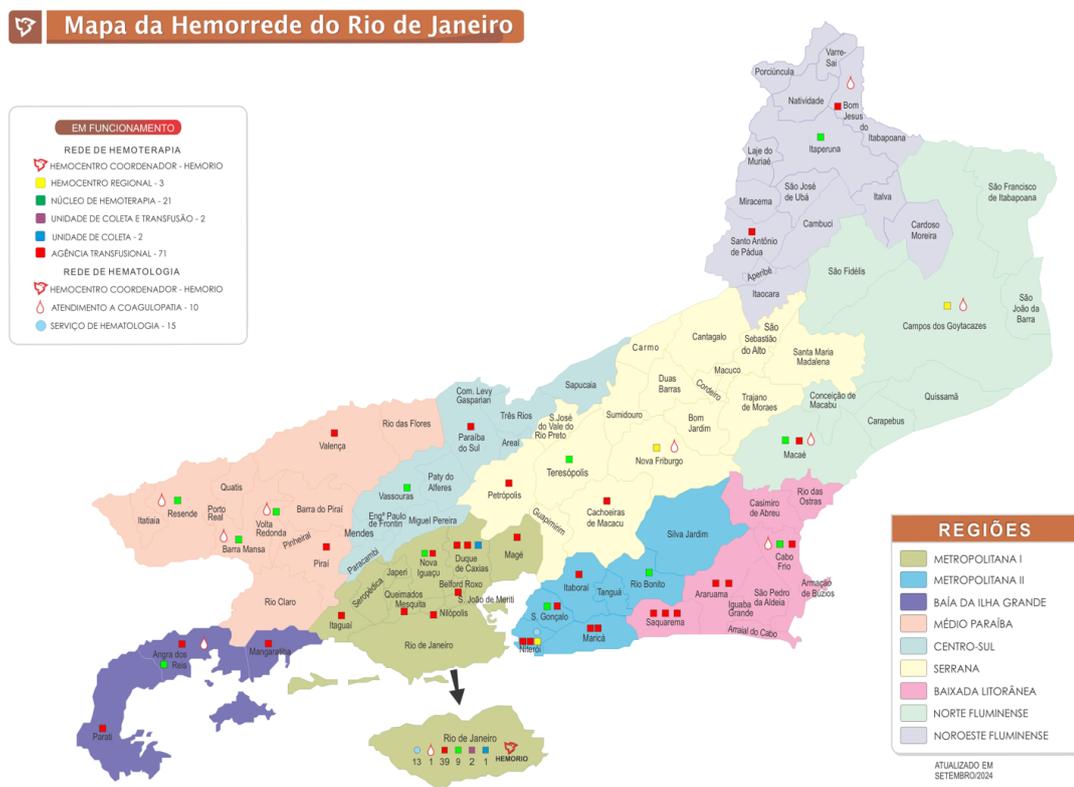
3 REVISÃO BIBLIOGRÁFICA

Neste capítulo é apresentado a revisão bibliográfica que no contexto deste trabalho, se concentra na apresentação de aplicativos relacionados. O objetivo do capítulo é compor o estado da arte para o tema destacando as principais contribuições de cada aplicação, bem como identificar as limitações das soluções apresentadas e as oportunidades de melhorias existentes.

3.1 HemorRede

É um conjunto de Serviços de Hemoterapia e Hematologia, organizados de forma hierarquizada e regionalizada, de acordo com o nível de complexidade das funções que desempenham e a área de abrangência para assistência. Tem como objetivo fornecimento de sangue e hemocomponentes em quantidade suficiente para atender à a população de cada região (HEMOREDE, 2023). Como por exemplo o Hemorio, que lista os hemocentros da região do Rio de Janeiro, como apresentado na Figura 3.

Figura 3 – Mapa HemorRede do Rio de Janeiro.



Fonte: Adaptado de (HEMORREDE, 2023).

3.2 HemoBrás

Na Figura 4 é apresentado o site do Hemobrás.

Figura 4 – Tela de Início - Hemobrás.



Fonte: Adaptado de (HEMOBRÁS, 2023).

A Hemobrás, em conjunto ao Ministério da Saúde, é uma indústria farmacêutica da mais alta relevância para o país. Possui a função de pesquisar, desenvolver e produzir medicamentos hemoderivados e biotecnológicos para atender prioritariamente ao Sistema Único de Saúde (SUS) (HEMOBRÁS, 2023).

3.3 HemoVida(ConectSUS)

Na Figura 5 é apresentado o aplicativo HemoVida do ConectSUS.

Figura 5 – Tela de login - Hemovida.



Fonte: Adaptado de (HEMOVIDA, 2023).

O Hemovida é um aplicativo móvel fornecido pelo governo federal que tem o objetivo de reter doadores de sangue e divulgar informações necessárias ao público acerca das doações de sangue.

Desenvolvido especificamente para bancos de sangue, o Hemovida tem como objetivo informatizar todo o ciclo de doação de sangue, desde a captação até a distribuição do material, controlando cada etapa do processo. Permite aos diversos gestores nas esferas Federal, Estadual e Municipal, o pleno acesso aos dados indispensáveis à tomada de decisões quanto ao gerenciamento em todo o processo do sangue na HEMOREDE. As principais funcionalidades da plataforma são, possibilitar o cadastramento de doador, dados sobre histórico de doenças, transfusões de sangue, pré-triagem e supervisão de infusão de derivados (HEMOVIDA, 2023).

3.4 Hemogram

Na Figura 6 é apresentada a tela inicial da aplicação Hemogram.

Figura 6 – Tela de login - Hemogram.



Fonte: Adaptado de (Hemogram, 2023).

O Hemogram é um aplicativo colaborativo desenvolvido pelo Instituto das Cidades Inteligentes. O intuito é fortalecer o laço entre a população e as organizações de saúde que mantêm os bancos de sangue, os postos de coleta e redes de distribuição vinculados ao Centro de Hematologia e Hemoterapia do Paraná (Hemepar). Tem a finalidade de promover e incentivar a doação de sangue. As principais funcionalidades são, histórico de doação, próximas doações, localizar postos de atendimento, informativos sobre medula óssea, critérios para doação (Hemogram, 2023).

3.5 Um Só Sangue

Na Figura 7 é apresentado a tela inicial do aplicativo Um só sangue.

Figura 7 – Tela de login - Um só sangue.



Fonte: Adaptado de (Um Só Sangue, 2023).

O Programa Um Só Sangue foi criada com o intuito de enfatizar a importância do processo de doação de sangue. Assim, o Programa objetiva estimular e estabelecer um calendário anual perene que torne a doação de sangue no Brasil um ato constante e sob o olhar de todos . Uma das principais funcionalidades são, agendar doação, histórico de agendamentos, cadastro de instituições de doação de sangue, informativos sobre a doação de sangue (Um Só Sangue, 2023).

3.6 Comparativo dos Trabalhos Relacionados

Na Tabela 2 é apresentado o comparativo com entre os principais aplicativos já existentes.

Tabela 2 – Comparativo dos trabalhos relacionados.

Funcionalidade	Aplicativos	Vida+	Hemobrás	Hemovida (ConectSUS)	Hemogram	Um Só Sangue
Agendamento online		Sim	Não	Sim	Sim	Sim
Histórico de doações		Sim	Não	Não	Não	Não
Localização das instituições de doação de sangue		Sim	Sim	Sim	Sim	Sim
Alerta de carência de bolsas de sangue de doação de sangue		Sim	Não	Não	Não	Não
Informativos sobre a doação de sangue		Sim	Sim	Sim	Sim	Sim
Informativos sobre a doação de medula óssea		Sim	Sim	Sim	Sim	Sim
Teletriagem		Sim	Não	Sim	Não	Não
Questionário de Saúde do Doador		Sim	Não	Não	Não	Não
Exibir Instituições de Doação de Sangues Próximas		Sim	Não	Sim	Sim	Sim
Cadastrar Unidades de Coleta		Sim	Não	Sim	Sim	Sim
Cadastro de Usuário Doador		Sim	Não	Sim	Sim	Sim

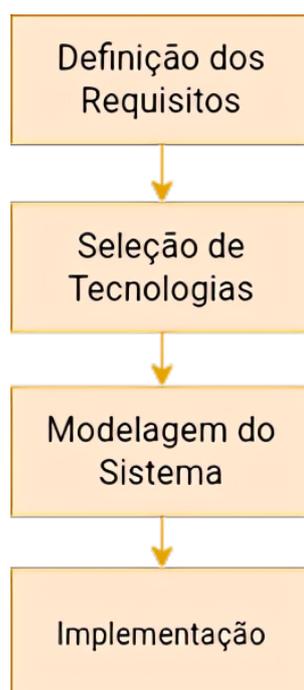
Fonte: Autoria própria.

No próximo capítulo é apresentada a metodologia para desenvolvimento do trabalho.

4 METODOLOGIA

Neste capítulo são abordados os elementos selecionados para a realização do trabalho, apresentando a sucessão das principais atividades desenvolvidas. Dessa forma, é exposto na Figura 8 a sequência do desenvolvimento para a implementação da aplicação.

Figura 8 – Etapas do Processo de Desenvolvimento.



Fonte: Autoria Própria.

A priori, foram definidos os requisitos. A posteriori, foram selecionadas tecnologias que atendessem às exigências da primeira etapa, bem como, a modelagem dos diagramas, visando analisar o funcionamento do sistema. Por fim, o desenvolvimento do aplicativo, como apresentado em (BATISTA, 2022). A análise de cada uma das etapas apresentadas na Figura 8 são descritas a seguir.

4.1 Definição dos Requisitos

A primeira etapa iniciou com o levantamento dos requisitos, analisando as necessidades de maior prioridade dos usuários alvo. Visualizando assim, as funcionalidades essenciais que um aplicativo mHealth deveria conter, tornando eficaz a sua utilização. Em seguida, pesquisas foram realizadas acerca de aplicações semelhantes, evidenciando as que possuem controle de usuário, relatando histórico de doações, alertas das instituições sobre o tipo sanguíneo em carência e teletriagem. A proposta do trabalho surgiu da

fundamentação desses dados adquiridos, tendo em vista a definição dos requisitos e o escopo do sistema.

4.2 Seleção de Tecnologias

Na segunda etapa uma busca acerca das tecnologias Front-end mais utilizadas no desenvolvimento mobile foram realizadas. Inicialmente foi necessário fazer um levantamento das tecnologias Front-end, com o objetivo de escolher a tecnologia adequada para o desenvolvimento da aplicação.

4.3 Modelagem do Sistema

Na terceira etapa é feita a representação dos requisitos e funcionalidades do sistema. A partir da criação dos diagramas de casos de uso, diagramas de atividade, diagramas de sequência e da engenharia de requisitos com a criação das histórias do usuário.

4.4 Implementação

A implementação consiste na parte prática do desenvolvimento *front-end* do aplicativo. Onde as funcionalidades definidas são inseridas no código. O processo foi fundamentado a partir dos requisitos e da modelagem apresentados na etapa 2 e 3, respectivamente.

A etapa de implementação é dividida em dois ciclos. A primeira dá ênfase à criação de telas para versão *mobile* e *desktop*, possuindo a finalidade de serem usuais, melhorarem a experiência do usuário no aplicativo e serem intuitivas.

A segunda pode ser resumida na codificação do sistema web, onde as instituições de coleta de recursos podem ser cadastradas, bem como, a inserção das demais funcionalidades.

5 RESULTADOS

Neste capítulo são apresentados os resultados deste trabalho, onde a modelagem de uma aplicação web e mobile foi criada a partir do agendamento online, questionário do doador, alerta de carência de bolsas de sangue para um usuário doador e uma instituição de doação de sangue que deseja otimizar seus atendimentos diários.

5.1 Descrição do sistema

A aplicação visa facilitar e otimizar o atendimento das instituições de doação de sangue e a experiência do usuário durante as etapas do processo. Para utilizar o aplicativo é necessário o cadastro prévio. O potencial doador após a conclusão do cadastro deverá atualizar seu perfil para realizar o agendamento online. Assim deverá inserir alguns dados como: foto de perfil, nome, filiação, idade, sexo biológico, RG, CPF, telefone, CEP, profissão, trabalho atual, peso, altura, tipo sanguíneo e fator de rh. Após a atualização do perfil, o usuário pode pesquisar instituições ou alertas em que há carência de bolsas de sangue que o usuário é compatível.

Além disso, o usuário pode escolher a instituição em que agendará a entrevista de acordo com a disponibilidade da unidade selecionada. Bem como, responder as perguntas do questionário do doador, que consistem em perguntas enviadas pelas instituições sobre o estado de saúde do potencial doador a fim de agilizar o processo da entrevista durante as etapas do processo.

Hemocentros e hemonúcleos também podem se cadastrar, devem inserir dados com CNPJ, CNES, razão social, endereço, telefone e e-mail. Assim que as informações forem validadas, a instituição pode realizar login e finalizar o cadastro, inserindo data e horário de disponibilidade, bem como, adicionar foto do local de doação e anexar questionário do doador, caso queiram otimizar o atendimento presencial. Nesse contexto, eles também podem visualizar o agendamento de entrevista de potenciais doadores que queiram doar para a organização e emitir alertas para doadores compatíveis caso ocorra de haver carência no estoque de bolsas de sangue e cadastrar o tipo sanguíneo em falta.

5.2 Definição do *Framework Web e mobile*

Na metodologia foi citado que os *frameworks* que se destacaram foram o React e o Angular. Logo após, um levantamento sobre os *frameworks* mais populares e utilizados foi realizado. Assim, na Tabela 3 é apresentado uma comparação entre eles, a partir desse quadro decidimos qual *framework* utilizar no desenvolvimento do sistema.

Tabela 3 – Comparação entre *frameworks*.

	Angular	React	Vue.js
Linguagem de programação	JavaScript, TypeScript, Dart, HTML e CSS.	JavaScript, HTML e CSS.	JavaScript, HTML e CSS.
Empresa	Google	Meta	Financiado e apoiado integralmente pela comunidade de código aberto.
Documentação	Documentação extensa e difícil de navegar, porém completa e técnica.	Documentação detalhada e pequena mas com muitos exemplos com trechos de códigos.	Documentação curta, porém apresenta guias e tutoriais.
Curva de Aprendizado	Alta, demanda conhecimento em typescript e estrutura MVC.	Média, conceito como hooks e fluxo de dados unidirecional levam tempo para dominar.	Baixa, intuitivo e fácil.
Sobrecarga	Alta, é um <i>framework</i> complexo de estrutura robusta, acompanha complementos. Ocasionalmente sobrecarga de código.	Moderada, o desenvolvedor pode selecionar apenas o que precisa, não causa sobrecarga, mas tem a capacidade de suportar bibliotecas de terceiros caso queira usar a funcionalidade completa.	Baixa, escala de acordo com a complexidade do projeto, não ocasiona sobrecarga.
Bibliotecas	É um <i>framework</i> completo e técnico, não necessita de bibliotecas adicionais.	Necessita de bibliotecas adicionais para desenvolvimento de interfaces.	Não possui a mesma quantidade de recursos como o React e o Angular.

Fonte: Autoria própria.

Depois de feita a comparação entre o *Frameworks*, foram vistas algumas vantagens em utilizar o *Framework* Angular para o desenvolvimento da aplicação. Essas vantagens foram:

- O Angular é desenvolvido com TypeScript, o que traz segurança para o código pois permite detectar erros no desenvolvimento antes da execução.
- O Angular é uma estrutura completa e suporta arquitetura robusta, não requer bibliotecas adicionais.
- O Angular possui uma interface do usuário simples e intuitiva, devido seus layouts, possui uma variedade de segmentos de design pré-estabelecidos, o que facilita o desenvolvimento.

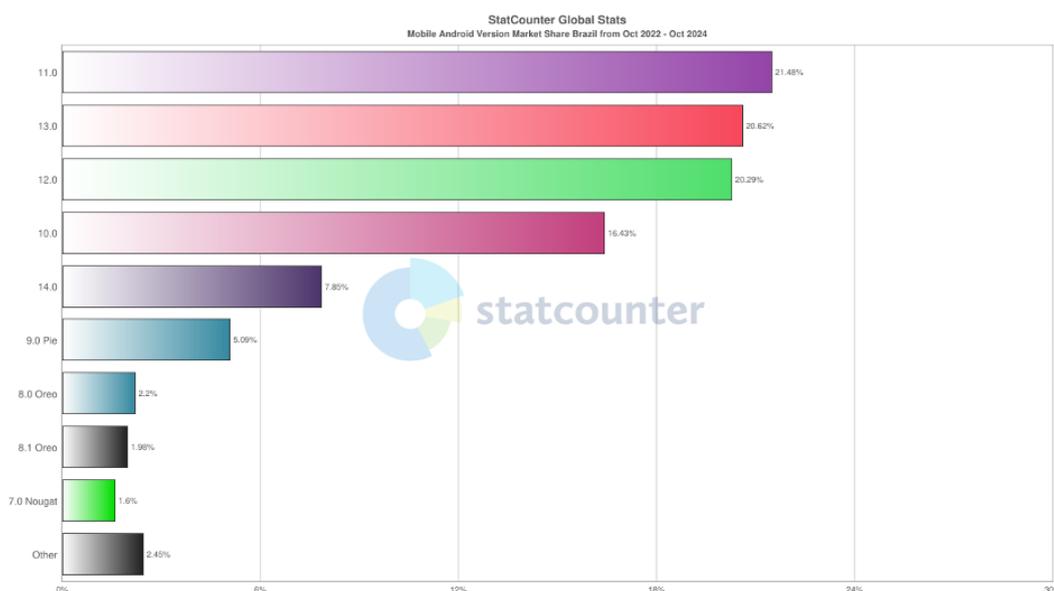
Dessa forma, nesse trabalho será utilizado o *Framework* Angular. Após a escolha do *framework* web, demos prosseguimento no levantamento de um *framework* mobile que fosse compatível com o Angular. Entre os que suportam o Angular, o selecionado foi o Ionic, sendo os benefícios principais sendo:

- O Ionic enquadra a facilidade de desenvolvimento web com a habilidade de acessar recursos nativos utilizando uma única base de código, por serem similares.
- É multiplataforma, com o Ionic um único código pode ser executado em diferentes sistemas operacionais como Android, Windows e IOS.
- O Ionic se baseia em linguagens de programação já conhecidas como HTML, CSS e JavaScript, obtendo assim, um curva de aprendizado suave, tornando a transição para o desenvolvimento móvel intuitiva.

- É possível realizar testes no browser, com o Ionic.

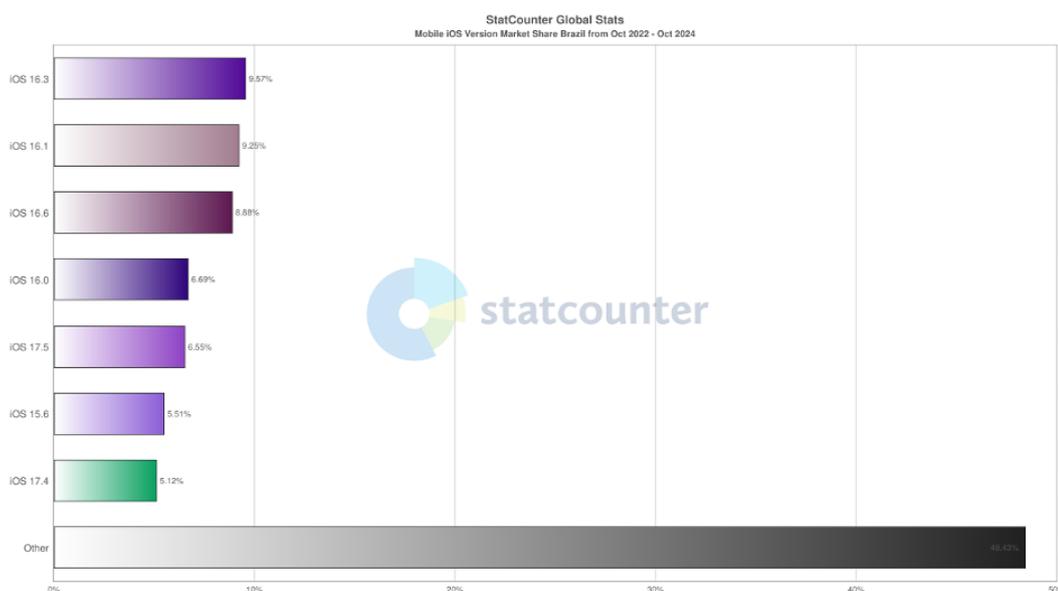
Assim, a aplicação será desenvolvida para Windows, Android e IOS. De acordo com o (Statcounter, 2024), na Figura 9, um site de análises de tráfego web, o Android 11 é a versão que vem sendo mais utilizada no Brasil nos últimos dois anos e prosseguindo a pesquisa no (Statcounter, 2024) a versão mais utilizada no IOS no Brasil atualmente são as abaixo da 14, podendo ser vista na Figura 10.

Figura 9 – Versão Android.



Fonte: Adaptado de (Statcounter, 2024).

Figura 10 – Versão IOS.



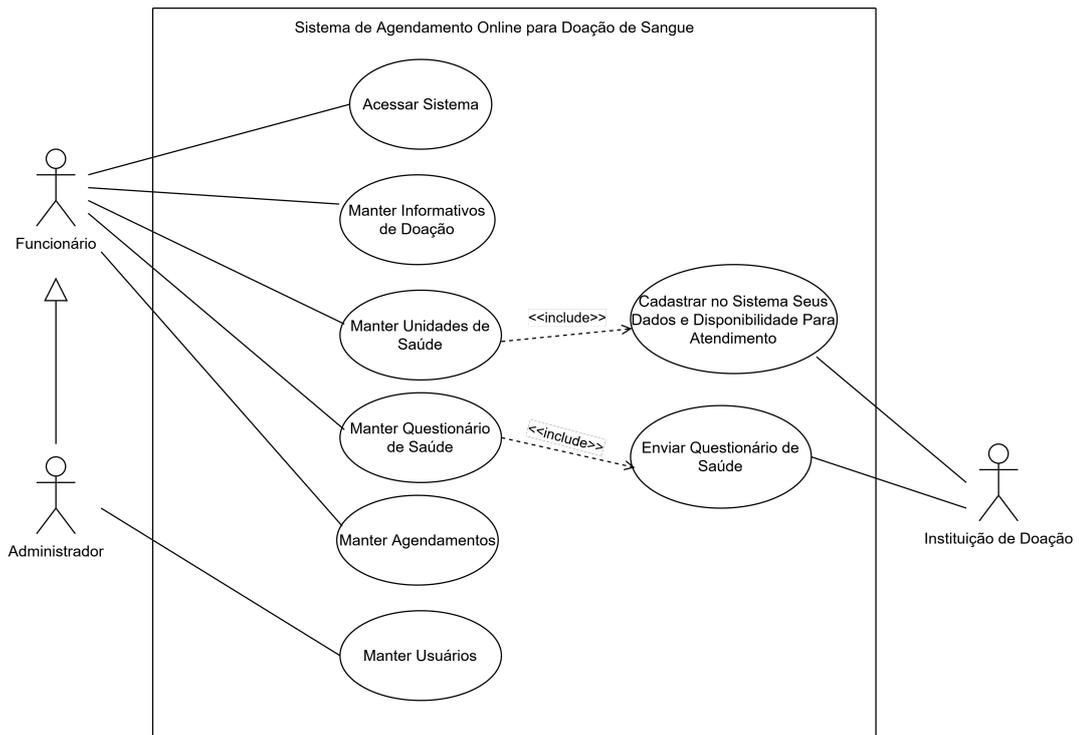
Fonte: Adaptado de (Statcounter, 2024).

Após as escolhas dos *Frameworks* foi feito o diagrama de casos.

5.3 Diagramas de casos de usos

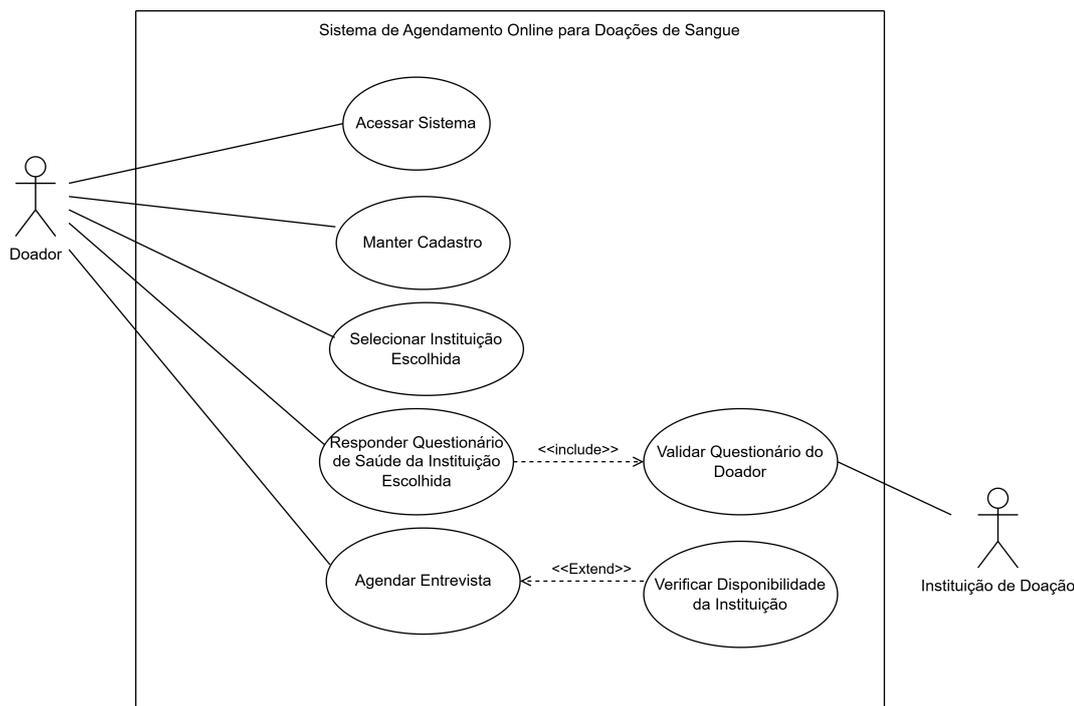
Nas Figuras. 11 e 12 são apresentados os diagramas de casos para a aplicação.

Figura 11 – Diagrama de casos de uso (gerenciamento administrativo).



Fonte: Autoria própria.

Figura 12 – Diagrama de casos de uso (funcionalidades do usuário).



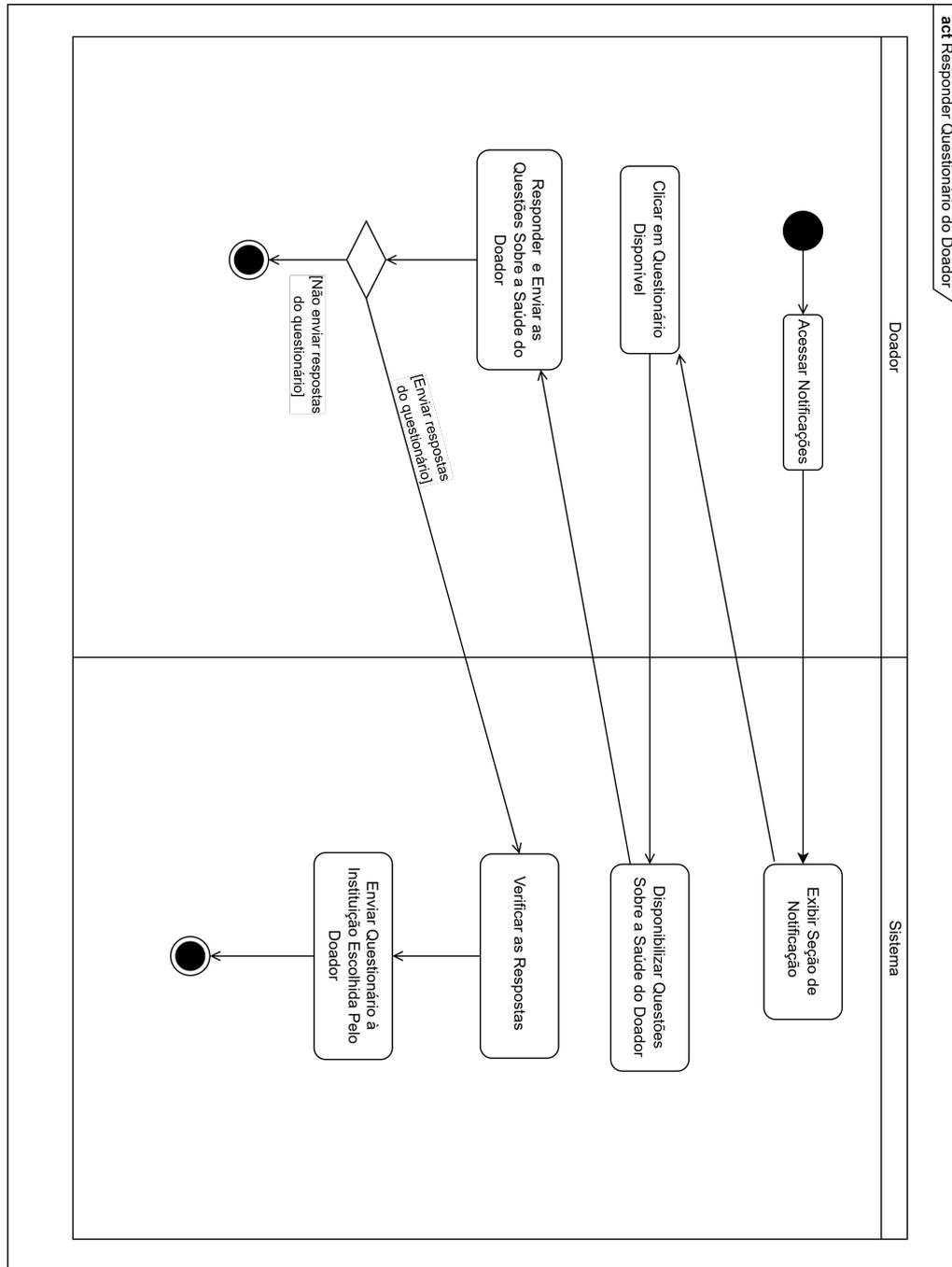
Fonte: Autoria própria.

Nessas figuras, está apresentado o diagrama de caso de uso do aplicativo. Nesse contexto está representado qual ator é responsável por cada funcionalidade do sistema e como é a interação entre eles.

- Funcionário - terá como funcionalidades essenciais manter o informativos de saúde, manter instituições de doação, manter questionário de saúde e manter agendamentos.
- Administrador - herda as funcionalidades do funcionário e, ainda, pode manter os usuários do sistema.
- Instituições de doação - terá como funcionalidade cadastrar as informações do local, como adicionar foto, adicionar questionário, cadastrar alerta, endereço, telefone, disponibilidade de data e horário para agendamento e, ainda, validar a resposta do questionário respondido.
- Doador - terá como funcionalidade manter seu cadastro, selecionar uma instituição, responder questionário do local escolhido e agendar entrevista.

Em seguida, foi desenvolvido o diagrama de atividades, como apresentado nas Figuras 13, 14 e 15.

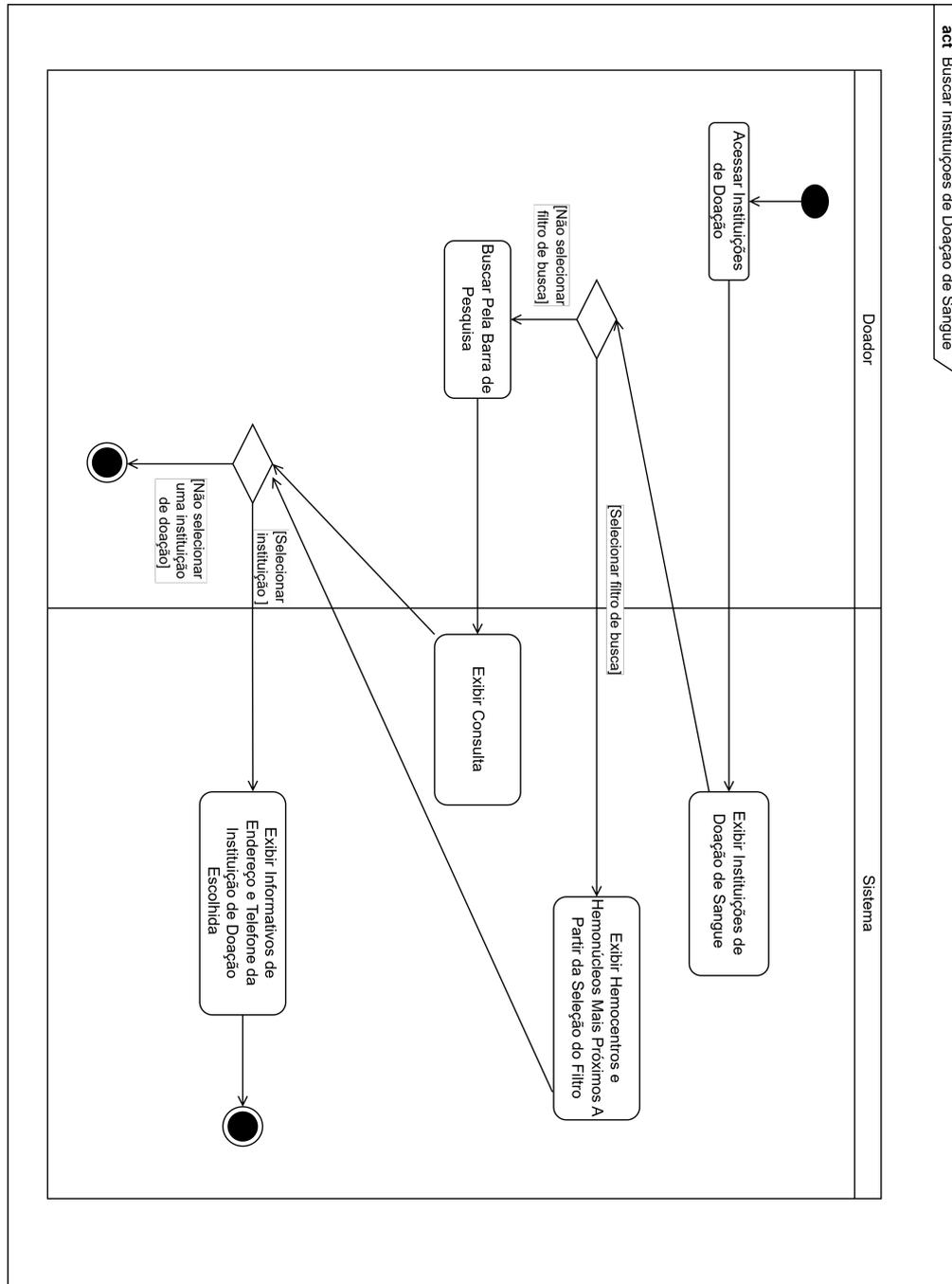
Figura 13 – Diagramas de Atividade de Responder Questionário do Doador.



Fonte: Autoria própria.

O usuário acessa o questionário, o sistema disponibiliza o questionário de saúde do doador. Dessa forma, assim que o questionário for respondido o potencial doador pode optar entre enviar o questionário ou cancelar a operação. Logo, se o questionário for enviado, o sistema irá verificar as respostas e enviar o questionário à instituição escolhida pelo usuário.

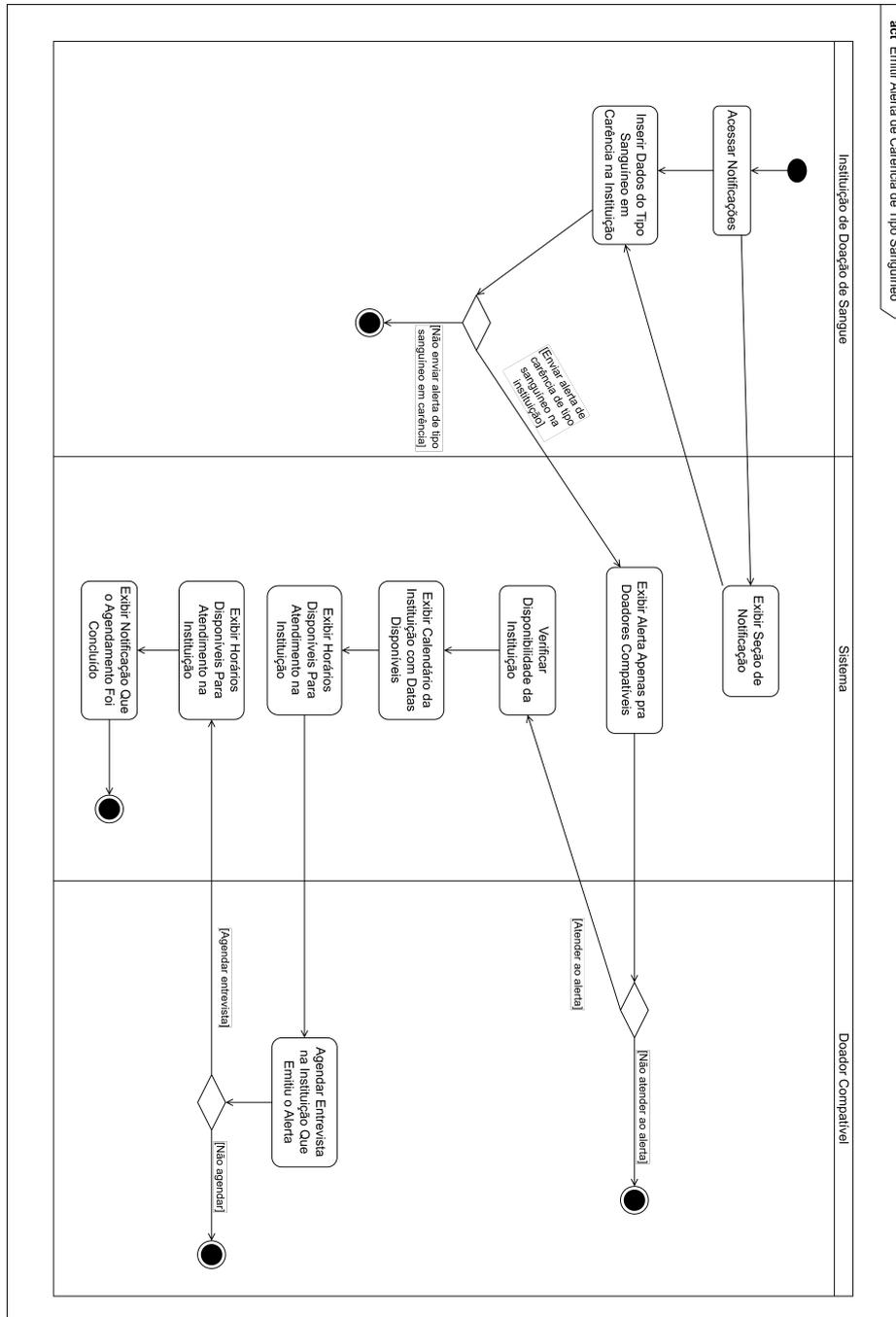
Figura 14 – Diagrama de Atividade de Buscar Instituições de Doação de Sangue.



Fonte: Autoria própria.

O usuário acessa a seção de Instituições de doação, o sistema exibe a tela de buscar instituições dos hemocentros e hemonúcleos cadastrados. O usuário pode escolher entre selecionar um filtro de busca, explorar pela barra de pesquisa ou não realizar nenhuma operação. Assim, se o filtro de busca for selecionado, o sistema exibirá as instituições mais próximas do usuário. Dessa forma, se uma instituição for selecionada, o sistema exibirá as informações de foto, endereço, telefone, datas e horários disponíveis de atendimento.

Figura 15 – Diagrama de Atividade de Emitir Alerta.



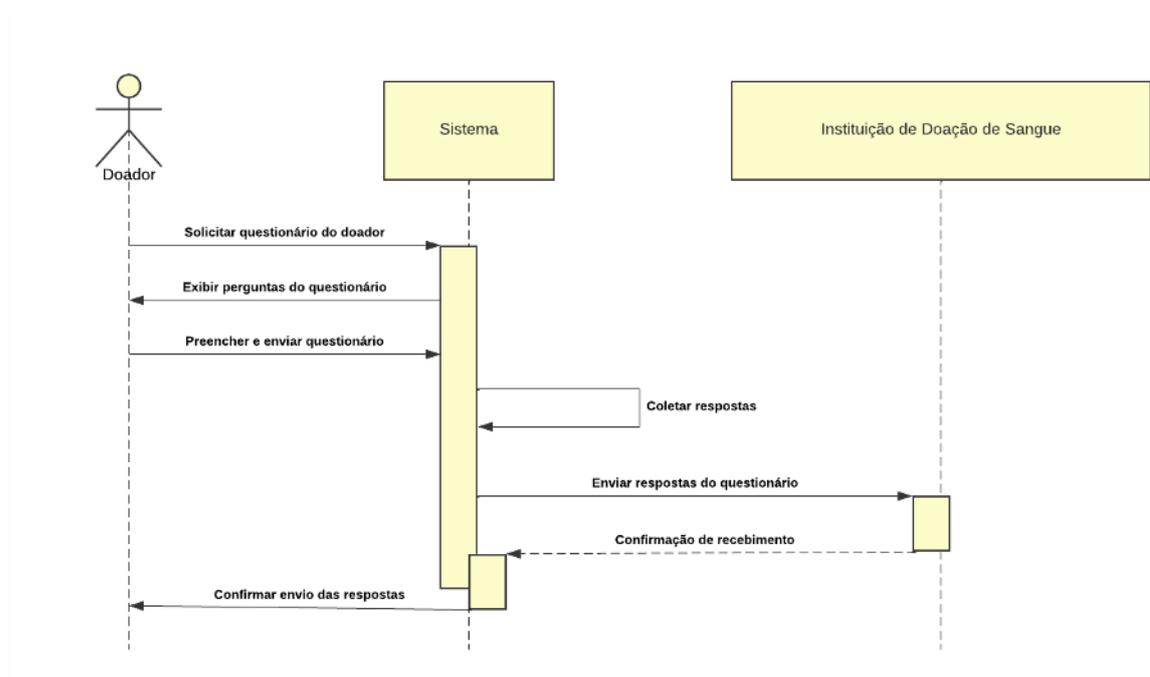
Fonte: Autoria própria.

A instituição acessa as notificações e escolhe cadastrar um alerta, o sistema exibe a tela de cadastro e é inserido as informações do tipo sanguíneo em carência e a disponibilidade do local para atendimento. O usuário pode escolher entre enviar o alerta ou cancelar a opção. Assim, se a opção selecionada for enviar, o sistema exibirá um alerta aos doadores aptos e compatíveis com o alerta emitido. Dessa forma, quem recebe a mensagem pode optar em atender o alerta ou ignorar a mensagem. Logo, se o potencial doador clicar no alerta, o sistema verificará a disponibilidade da instituição que emitiu o alerta e irá exibir

a tela de agendamento com o horário e datas disponíveis da instituição para o usuário marcar a entrevista. Em seguida, foi desenvolvido os diagramas de sequência.

Nas Figuras 16 e 17 são apresentados os diagramas de sequência desenvolvidos para a aplicação.

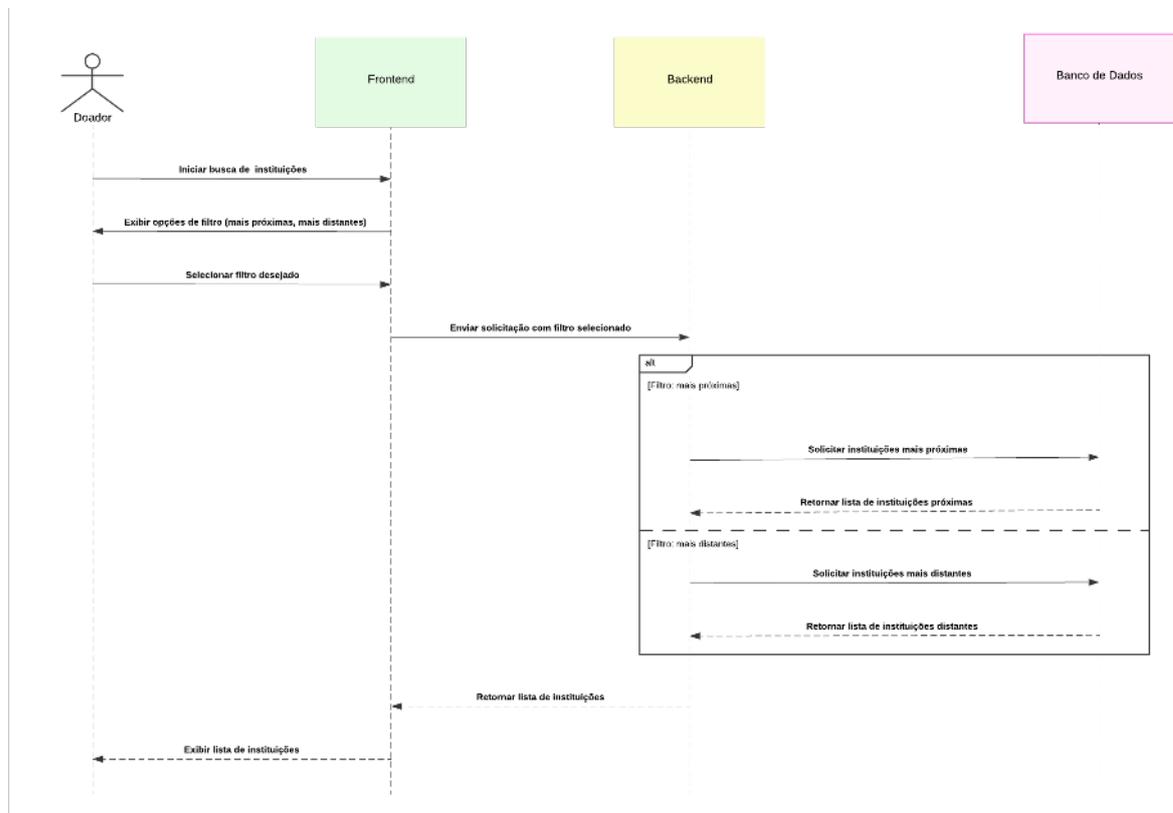
Figura 16 – Diagrama de Responder Questionário.



Fonte: Autoria própria.

Para responder o questionário do doador, o usuário acessa a seção de notificações e clica na caixa de mensagem que indica que o questionário da instituição escolhida está disponível e solicita o questionário. Assim, é liberado o acesso do questionário e após ser respondido o documento é enviado à instituição para ser avaliado. Dessa forma, quando o usuário ir presencialmente à entrevista o processo de doação é otimizado, devido algumas informações já estarem registradas no sistema, sendo necessário somente avaliá-las antes de enviar o potencial doador para a etapa de triagem.

Figura 17 – Diagrama de Sequência de Buscar Instituições.



Fonte: Autoria própria.

Este diagrama apresenta a atividade de buscar instituições, a pesquisa dos hemocentros e hemonúcleos pode ser realizada pela barra de pesquisa ou pela adição de filtros para selecionar o que possui a localização mais próxima. O fluxo do sistema consiste na comunicação de requisição e resposta entre o usuário e frontend, do frontend com o backend e do backend com o banco de dados. Com os diagramas feitos, foi desenvolvido a história do usuário.

5.4 Histórias do usuário

A seguir são apresentadas as histórias do usuário.

5.4.1 *História de Usuário - Login*

Título: Fazer Login
US: Como administrador, desejo que o doador e a instituições consigam fazer o login para terem acesso ao agendamento da entrevista, buscar instituições, edição de cadastro e notificações.
Critérios de Aceitação: 1 - quero ter a opção da senha ser visível no login. 2 - quero ter a opção de manter a conta conectada, para não necessitar inserir os dados várias vezes.

5.4.2 *História de Usuário - Enviar Questionário do Doador*

Título: Enviar questionário do Doador
US: Como instituição, desejo otimizar as etapas iniciais do processo de doação de sangue, tendo em vista, obter mais adesão de potenciais doadores e poder disponibilizar um questionário de saúde do doador para tornar o fluxo de atendimento mais eficaz.
Critérios de Aceitação: 1 - a instituição deve ter ter o CNPJ e o CNES validado pelo sistema. 2 - a instituição deve ter finalizado o cadastrado. 3 - as informações do cadastro devem ter sido avaliadas pelo administrador do sistema.

5.4.3 *História de Usuário - Responder Questionário do Doador*

Título: Responder questionário do Doador
US: Como doador, desejo doar sangue para a instituição que agendei entrevista. Gostaria de agilizar o atendimento respondendo um questionário disponibilizado pela instituição que escolhi e se validado otimizar o processo de entrevista presencial.
Critérios de Aceitação: 1 - o usuário deve ter ter o cpf, o rg validado pelo sistema. 2 - o usuário deve ter o tipo sanguíneo e o fator rh cadastrado no sistema. 3 - o usuário deve ter agendado entrevista na instituição que disponibilizou o questionário.

5.4.4 *História de Usuário - Buscar Instituições de Doação*

Título: Buscar Instituições de Doação
US: Como doador, desejo agendar entrevista em um hemocentro ou hemonúcleo mais perto de onde moro ou poder pesquisar outros, e após escolher, visualizar as informações de endereço, telefone e horários de atendimento da instituição escolhida.
Critérios de Aceitação: 1 - o usuário deve ter realizado o login. 2 - o usuário deve ter inserido os dados sobre o tipo sanguíneo e o fator de rh. 3 - o usuário deve ter finalizado o cadastro.

5.4.5 *História de Usuário - Agendar Entrevista*

Título: Agendar Entrevista
US: Como doador, desejo agendar entrevista em um hemocentro ou hemonúcleo que escolhi e visualizar as datas e horários disponíveis dessa instituição.
Critérios de Aceitação: 1 - o usuário deve ter realizado o login. 2 - o usuário deve ter inserido os dados sobre o tipo sanguíneo e o fator de rh. 3 - o usuário deve ter finalizado o cadastro.

5.4.6 *História de Usuário - Visualizar Agendamentos*

Título: Visualizar Agendamentos
US: Como instituição, desejo visualizar os agendamentos de potenciais doadores que escolheram minha instituição no sistema.
Critérios de Aceitação: 1- a instituição deve ter permissão de acesso a esta seção. 2 - a instituição deve ter o CNPJ e o CNES validado pelo sistema. 3 - a instituição deve ter finalizado o cadastrado. 4- as informações do cadastro devem ter sido avaliadas pelo administrador do sistema.

5.4.7 *História de Usuário - Emitir Alerta*

Título: Emitir Alerta
US: Como instituição, desejo emitir alerta que o nosso hemocentro ou hemonúcleo está com carência de bolsas de sangue e inserir as informações do tipo sanguíneo e disponibilidade de horário para atendimento
Critérios de Aceitação: 1 - as informações do cadastro devem ter sido avaliadas pelo administrador do sistema. 2 - a instituição deve ter permissão de acesso a esta seção. 3 - a instituição deve ter o CNPJ e o CNES validado pelo sistema. 4 - a instituição deve ter finalizado o cadastrado.

Com as histórias do usuário foram desenvolvidas as telas e a apresentação web.

5.5 *Apresentação da Aplicação mobile e Web*

A seguir estão apresentadas as telas da aplicação, onde as funcionalidades principais são: Tela de Login, de Menu, de Instituições de Doação, de agendamento, entre outras.

5.5.1 *Aplicação mobile*

5.5.1.1 *Tela de Login*

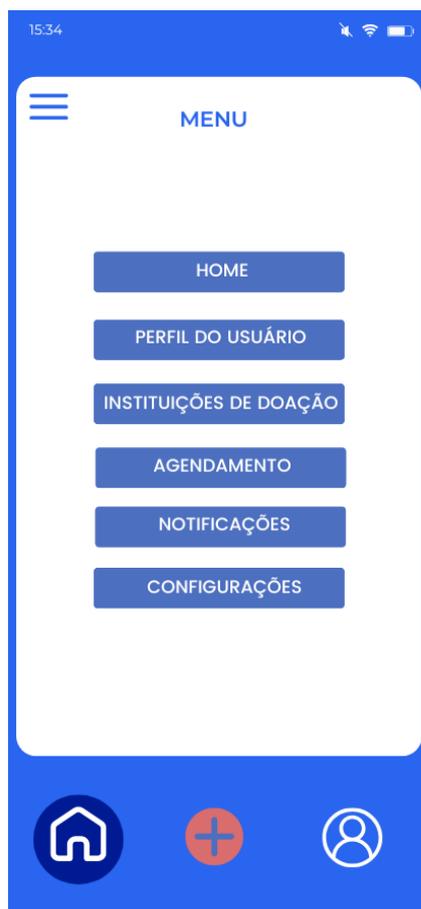
Para ter acesso ao aplicativo móvel é necessário um cadastro prévio. Nesta aplicação potenciais doadores podem registrar-se. Se o doador já for cadastrado, pode realizar a autenticação inserindo o login e senha de acesso correspondente. Assim, o usuário pode manter-se conectado automaticamente ou não e pode redefinir a senha, como observado na Figura 18.

Figura 18 – Tela de login - *mobile*.

Fonte: Autoria própria.

5.5.1.2 Tela de Menu

Finalizado a etapa de cadastro, ou o usuário que já possui cadastro, será redirecionado para de entrada do aplicativo *mobile*, como apresentado na Figura 19.

Figura 19 – Tela de menu - *mobile*.

Fonte: Autoria própria.

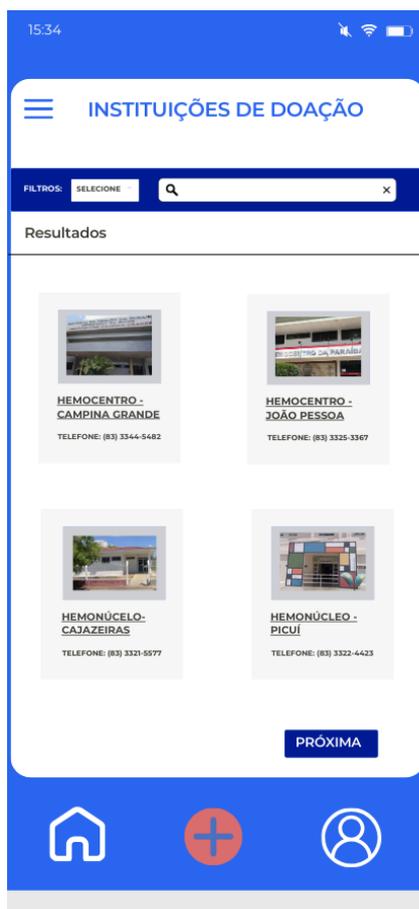
Na Figura 19, o usuário pode selecionar a funcionalidade que deseja utilizar, bem como, visualizar os ícones de acesso rápido, caso queira sair da aba.

Caso o usuário utilize o botão "instituições de doação" ele será direcionado para a tela apresentada a seguir.

5.5.1.3 Tela de instituições

Na tela apresentada na Figura 20, o usuário pode pesquisar hemocentros e hemonúcleos na barra de pesquisa, adicionar filtros que selecionam o mais próximo ou os que emitiram alertas de carência do tipo sanguíneo do usuário. Quando a instituição é selecionada, o sistema exibe os detalhes de informações da opção escolhida, como : endereço, telefone, e horário de funcionamento. O usuário pode escolher agendar entrevista em que a página será redirecionada para a seção agendamento ou voltar.

Figura 20 – Tela de instituições - *mobile*.



Fonte: Autoria própria.

Figura 21 – Tela de instituições - parte 02 - *mobile*.

Fonte: Autoria própria.

Na Figura 21 caso o usuário pressione o sobre a instituição é direcionado para a tela apresentada na Figura 22.

Figura 22 – Tela de buscar instituições - parte 03 - *mobile*.

Fonte: Autoria própria.

A partir dessa tela o usuário pode fazer o agendamento da entrevista necessária para a doação de sangue. Pressionando o botão "agendar entrevista". Dessa forma, forma o usuário será direcionado para a aba de agendamento.

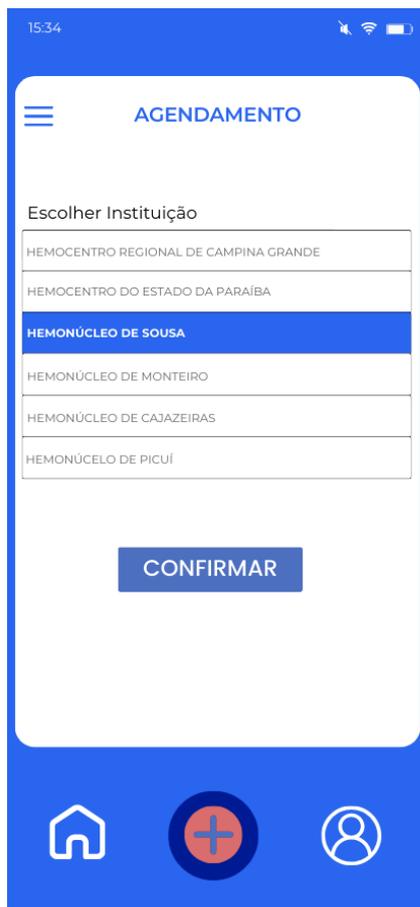
5.5.1.4 Tela de agendamento

Na tela de agendamento, Figura 23 o usuário escolhe a instituição de doação e o sistema oferece os horários disponíveis que a instituição cadastrou. A partir daí é selecionado a data e o horário escolhido e é confirmado o agendamento, Figuras 24 e 25.

Figura 23 – Tela de agendamento - *mobile*.



Fonte: Autoria própria.

Figura 24 – Tela de agendamento parte 02 - *mobile*.

Fonte: Autoria própria.

Figura 25 – Tela de agendamento parte 03 - *mobile*.

15:34

AGENDAMENTO

Escolher Instituição

HEMONÚCLEO DE SOUSA

Data

18/10/24

Horário

07:30

09:00

14:30

16:00

CONFIRMAR

Fonte: Autoria própria.

Nas Figuras 23, 24 e 25 estão representadas as opções escolhidas pelo potencial doador que deseja agendar a entrevista presencial. Dessa forma, os dados de local de atendimento e o dia já foram inseridos, exibindo as opções de horário. Onde o sistema filtra os horários dos agendamentos já realizados naquele dia, exibindo apenas os disponíveis.

Uma outra forma de acessar o agendamento de entrevista é acessa diretamente pela aba de agendamentos, presente dentro do "menu". Em seguida, o usuário pode verificar as notificações presentes na aplicação utilizando a tela de notificações da aba "menu".

5.5.1.5 Tela de notificação

Na tela de notificações do usuário doador, na Figura 26, é exibido o alerta de carência de tipo sanguíneo, em que pode ser atendido ou ignorado. Contudo, se atendido, ou seja, o lembrete for clicado, a página será redirecionada para a subseção de agendamento, a “agendar entrevista” na figura, onde o usuário pode selecionar o local que emitiu o alerta e confirmar a entrevista presencial.

Além disso, são exibidos lembretes de agendamento de entrevistas presenciais, com data, local, horário e assim que feito agendamento, o sistema disponibiliza o questionário

enviado pela instituição em que a entrevista foi marcada.

Figura 26 – Tela de notificação - *mobile*.



Fonte: Autoria própria.

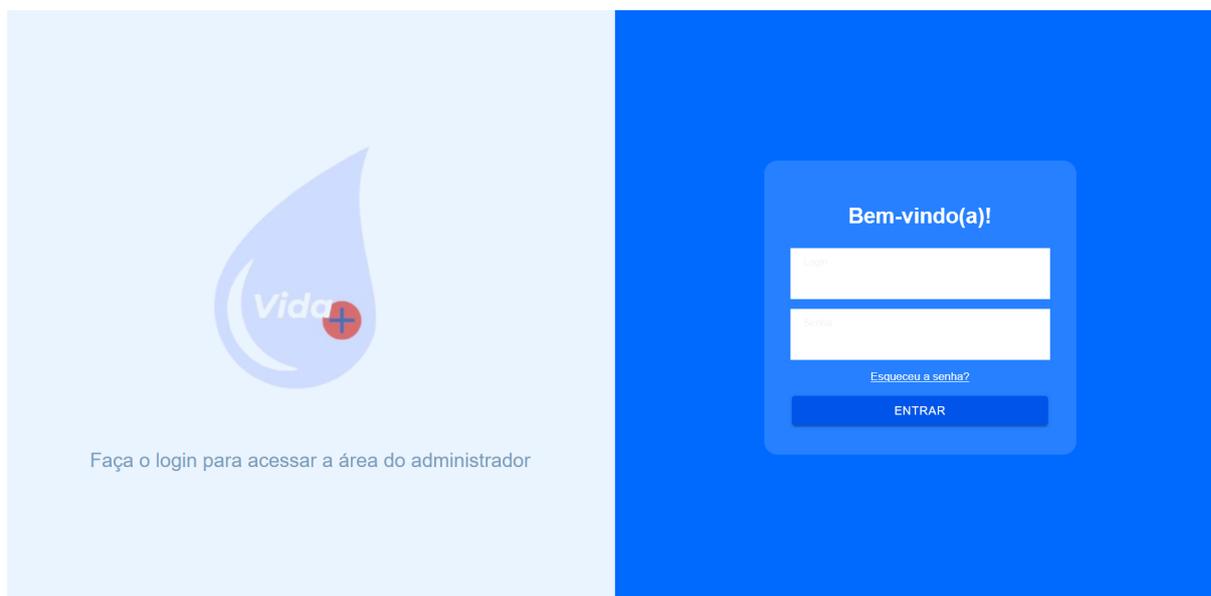
Após a finalização da aplicação mobile foi feito o desenvolvimento da aplicação web, como apresentado a seguir.

5.5.2 Aplicação web

O sistema web será utilizado por Instituições, administradores e funcionários, possuindo login e cadastre a instituição. Nesse contexto, é preciso realizar a autenticação do usuário para utilizar as funcionalidades do aplicativo e no caso das instituições é necessário um cadastro prévio para acessar a aplicação.

5.5.2.1 Tela de Login

Na Figura 27 é apresentada a tela de login da aplicação web.

Figura 27 – Tela de Login - *web*.

Fonte: Autoria própria.

5.5.2.2 Tela de instituições

Na Figura 28 é apresentada a tela de instituições, onde estão cadastrados todos os hemocentros e hemonúcleos do sistema, que podem ser acessados pela barra de pesquisa ou filtrados por id, nome ou município. As propriedades de cada estabelecimento estão discriminadas na tabela inserida na página, onde cada elemento está separado por coluna. Dessa forma, são exibidos dados de id, cadastro de estabelecimento de saúde, que é o cnes, nome da instituição, tipo, categoria, cidade, bairro e telefone.

Figura 28 – Tela de instituições - *web*.

INSTITUIÇÕES DE DOAÇÃO DE SANGUE SAIR

14 Unidades de saúde cadastradas

ID

ID	CNES	Nome	Instituição	Categoria	Cidade	Bairro
1	9466061	HEMATO	HEMOCENTRO	CENTRO DE ATENÇÃO HEMATOLOGICA	JOÃO PESSOA	TORRE
2	2606879	HEMOCENTRO DE SOUSA	HEMOCENTRO	CENTRO DE ATENÇÃO HEMATOLOGICA	SOUSA	JOSE FACUNDO DE LI
3	2399598	HEMOCENTRO DO ESTADO DA PARAIBA	HEMOCENTRO	CENTRO DE ATENÇÃO HEMATOLOGICA	JOÃO PESSOA	CENTENARIO
4	2612828	HEMOCENTRO REGIONAL DE CAMPINA GRANDE	HEMOCENTRO	CENTRO DE ATENÇÃO HEMATOLOGICA	CAMPINA GRANDE	CATOLE
5	2613204	HEMONUCLEO	HEMONUCLEO	CENTRO DE ATENÇÃO HEMATOLOGICA	GUARABIRA	CENTRO
6	2606321	HEMONUCLEO DE CATOLE DO ROCHA	HEMONUCLEO	CENTRO DE ATENÇÃO HEMATOLOGICA	CATOLE DO ROCHA	BATALHÃO
7	2794047	HEMONUCLEO DE ITAPORANGA	HEMONUCLEO	CENTRO DE ATENÇÃO HEMATOLOGICA	ITAPORANGA	CENTRO
8	2363410	HEMONUCLEO DE MONTEIRO	HEMONUCLEO	CENTRO DE ATENÇÃO HEMATOLOGICA	MONTEIRO	CENTRO
9	2794055	HEMONUCLEO DE PICUI	HEMONUCLEO	CENTRO DE ATENÇÃO HEMATOLOGICA	PICUI	MONTE SANTO
10	2794063	HEMONUCLEO DE PRINCESA IZABEL	HEMONUCLEO	CENTRO DE ATENÇÃO HEMATOLOGICA	PRINCESA IZABEL	ALTO DA CASCAVEL
11	2399598	HEMONUCLEO DE ITABAIANA	HEMONUCLEO	CENTRO DE ATENÇÃO HEMATOLOGICA	ITABAIANA	NOVA ITABAIANA
12	2794055	HEMONUCLEO DE PIANCÓ	HEMONUCLEO	CENTRO DE ATENÇÃO HEMATOLOGICA	PIANCÓ	OURO BRANCO
13	2606321	HEMONUCLEO DE CAJAZEIRAS	HEMONUCLEO	CENTRO DE ATENÇÃO HEMATOLOGICA	CAJAZEIRAS	CENTRO

Fonte: Autoria própria.

Ao clicar na instituição, na Figura 28, encontram-se as informações da opção escolhida, endereço, telefone, foto do local e horário de funcionamento. Clicando no botão editar informações apenas o usuário administrador e funcionário da unidade selecionada tem permissão de editar ou excluir as informações desta seção.

5.5.2.3 Tela de agendamento

Na Figuras 29 e 30 são apresentadas as telas de visualizar agendamento e visualizar histórico de doações. Apenas o administrador e os funcionários das instituições têm acesso à esta subseção. Entretanto, os funcionários da instituições só têm acesso aos agendamentos da sua unidade, enquanto o administrador consegue acessar todos os agendamentos e históricos das unidades. Na figura, está representada a subseção que apenas o administrador tem acesso. Assim, ele escolhe a data, horário e instituição e visualiza o perfil do usuário que agendou a entrevista ou pode optar por visualizar todos os históricos de doação da unidade selecionada.

Figura 29 – Tela de visualizar agendamento das instituições - *web*.

AGENDAMENTO				SAIR
VISUALIZAR AGENDAMENTOS				
Nome Completo (Desabilitado)	CPF (Desabilitado)	Email (Desabilitado)	RG (Desabilitado)	
Paulo Silva Barbosa	123.657.890-20	paulo.silva@gmail.com	3.876.089	
Instituição (Desabilitado)	Tipo Sanguíneo (Desabilitado)	Endereço *	Número *	
HEMOCENTRO REGIONAL DE C,	A+	Rua das Castanholas	75	
CEP *	Data	Horário		
28490506	04/06/2025	14:00		
VOLTAR				

Fonte: Autoria própria.

Na Figura 29, é apresentada a subseção visualizar agendamento, onde o administrador pode visualizar o agendamento feito em cada instituição, em que é possível selecionar a instituição desejada, data e o sistema oferece horários preenchidos que o agendamento foi realizado. Em sequência, a página redireciona para o campo de visualizar informações cadastrais, hemocentro e tipo sanguíneo de usuário agendados nos filtros selecionados. Nessa seção, o layout está desabilitado para edição, sendo o acesso restrito apenas para consulta. Adicionalmente, na figura 30, é exposta a tela de visualizar histórico.

```

CODIGO DE VISUALIZAR AGENDAMENTO

formFiltro = this.fb.group({
  instituicao: [null, Validators.required],
  data      : [null, Validators.required],
  horario   : [null, Validators.required],
  historico : ['nao', Validators.required]
});

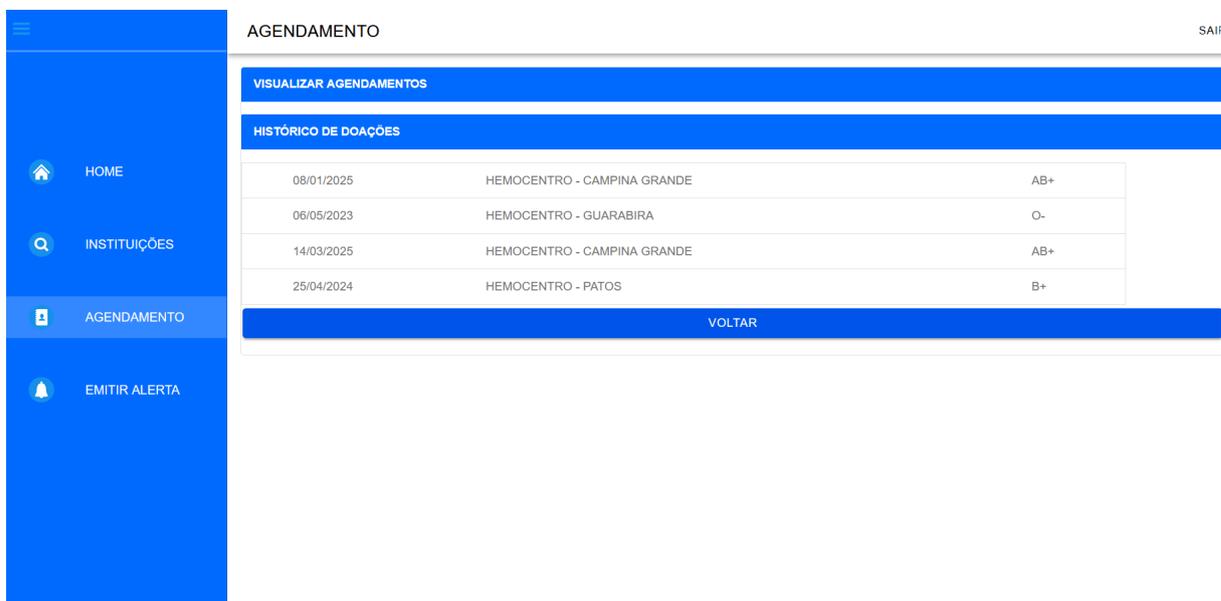
```

Fonte: Autoria própria.

Esse trecho de código está inserido no apêndice A - códigos, na subseção *agendamento.page.ts*. Onde, um formulário é criado a partir de um *FormGroup* com

quatro filtros são inicializados com um valor vazio, menos o histórico e a partir do valor do histórico é definido se o redirecionamento será para o histórico de agendamentos ou para os agendamentos realizados na data escolhida.

Figura 30 – Tela de visualizar histórico de agendamento - *web*.



Fonte: Autoria própria.

Assim, na figura 30 encontra-se a tela de visualizar o histórico da instituição. Nessa subseção o administrador não tem conhecimento sobre os autores das doações mas apenas sobre os dados gerais dos hemocentros ou hemonúcleos para controle de registro. Onde são exibidas as doações realizadas, os dias e o tipo sanguíneo doado. Logo, é possível visualizar na figura 28 os atributos de cada estabelecimento.

CODIGO DE VISUALIZAR HISTORICO DE DOAÇÕES

```
step: 'form' | 'detalhe' | 'historico' = 'form';
historico: RegistroHist[] = [
  { data: '2025-01-08', local: 'HEMOCENTRO -
  → CAMPINA GRANDE', tipo: 'AB+' },
  { data: '2023-05-06', local: 'HEMOCENTRO -
  → GUARABIRA', tipo: 'O-' },
  { data: '2025-03-14', local: 'HEMOCENTRO -
  → CAMPINA GRANDE', tipo: 'AB+' },
  { data: '2024-04-25', local: 'HEMOCENTRO -
  → SOUSA', tipo: 'B+' }];
```

Fonte: Autoria própria.

Esse trecho de código está inserido no apêndice A - códigos, na subseção *agendamento.page.ts* informa como o bloco de histórico foi construído. Os dados preenchidos com os elementos de data, local e tipo, especificam quando a doação ocorreu, em qual unidade foi realizada e qual o tipo sanguíneo registrado. Esse *array RegistroHist[]* ficará pré-carregado e será exibido quando entrar no bloco *step == 'historico'*.

5.5.2.4 Tela de emitir alerta

A subseção emitir alerta, conforme a figura 31 exibe um formulário para inserir qual instituição irá enviar a notificação para os demais usuários doadores, os dados de tipo sanguíneo em carência e o fator rh, se é positivo ou negativo. Após a confirmação, esse alerta é registrado na subseção de emitir alerta como uma notificação e é enviado para a seção home, na figura 32 de todos os doadores compatíveis cadastrados no sistema.

Figura 31 – Tela de emitir alerta - *web*.

A imagem mostra a interface de usuário para emitir um alerta. No topo, há um cabeçalho azul com o texto "EMITIR ALERTA" à esquerda e "SAIR" à direita. Abaixo, há uma barra azul com o texto "EMITIR ALERTA". O formulário principal contém os seguintes campos:

- Hemocentro: HEMOCENTRO CAMPINA GRANDE (com uma seta para baixo indicando uma lista suspensa).
- Tipo Sanguíneo em Carência: B (com uma seta para baixo indicando uma lista suspensa).
- Fator RH: + -
- Botão CONFIRMAR (em azul).

À esquerda do formulário, há uma barra lateral azul com ícones e o texto: HOME, INSTITUIÇÕES, AGENDAMENTO e EMITIR ALERTA (destacado).

Fonte: Autoria própria.

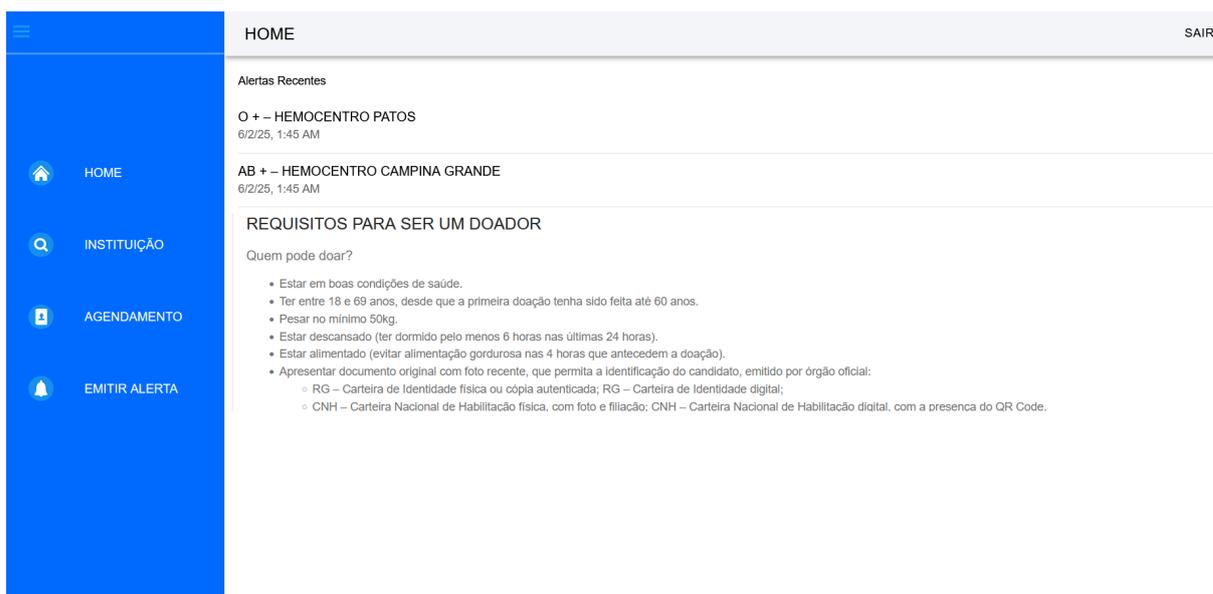
CODIGO DE EMITIR ALERTA

```
const dadosAlerta: Alerta = {
  tipoSanguineo: this.Alerta.value.tipoSanguineo,
  rh: this.Alerta.value.rhPos ? '+' : '-',
  hemocentro: this.Alerta.value.hemocentro,
  timestamp: new Date()
};
this.alertaService.enviarAlerta(dadosAlerta);
```

Nesse bloco, o trecho do código está localizado no apêndice A - códigos, na subseção *notificacoes.page.ts*. O objeto alerta é criado a partir da seleção dos elementos do formulário: hemocentro, tipo sanguíneo e o fator rh. Assim, a função alerta é criada em *this.alertaService.enviarAlerta(dadosAlerta)*. Logo, sempre que uma nova lista de alertas for emitida, ela exibirá as últimas notificações disponíveis.

Dessa forma, conforme a figura 32 tem acesso aos alertas e requisitos sobre quem pode doar. Portanto, o usuário consegue ter todas as informações necessárias para gerir a doação de sangue. Otimizando, dessa forma, o agendamento, o controle de registro de doações, a arrecadação de bolsas de sangue e a captação de doadores. Suprindo assim, a demanda de solicitações de fator sanguíneo das unidades. Facilitando e estreitando a interação do doador e das agências hematológicas.

Figura 32 – Tela de home - web



Fonte: Autoria própria.

CODIGO DE EMITIR ALERTA NA HOME

```
export class HomePage implements OnInit {  
  ...  
  constructor(private router: Router, private  
    ↪ alertaService: AlertaService) {}  
  
  ngOnInit() {  
  
    this.sub =  
    ↪ this.alertaService.alertas$.subscribe(lista => {  
      this.alertas = lista;  
    });  
  }  
  ...  
}
```

Fonte: Autoria própria.

Nesse bloco, está um trecho do código, podendo ser encontrado no apêndice A - códigos, na subseção *home.page.ts* o componente *HomePage* é registrado no no *alertas\$* do serviço *AlertaService* assim que é inicializado (*ngOnInit*). Logo, sempre que um alerta for gerado, a lista de notificações e o *home.page.html* são atualizados e o alerta é exibido ao mesmo tempo em notificações e na página *home*.

6 CONSIDERAÇÕES FINAIS

Neste trabalho foi executada a documentação e a implementação de uma aplicação para doação de sangue, em que o seu planejamento e requisitos podem ser inseridos em diferentes dispositivos. Entretanto, vale salientar que a implementação foi desenvolvida apenas para uma aplicação web, tendo o objetivo primário otimizar o processo de captação de doadores e atender as demandas das unidades hematológicas.

No processo de planejamento do sistema, tornou-se necessário a pesquisa acerca de tecnologias mais utilizadas e compatíveis em desenvolvimento web e mobile que funcionassem em ambas, a fim de otimizar a execução do trabalho. Entretanto, apenas a implementação da versão web foi o foco neste trabalho.

Durante as etapas de modelagem e implementação, foi necessário analisar o sistema como um usuário doador e administrador. Dessa forma, visualizar uma aplicação mais intuitiva que atenda à esses dois públicos e suas respectivas necessidades.

Algumas dificuldades ocorreram durante a execução deste trabalho, utilizando os recursos do framework Ionic. A responsividade da aplicação para mobile em Android e iOS não foi satisfatória, necessitando um estudo mais aprofundado sobre os componentes das tecnologias, apache cordova e angular. Podendo assim, ser executados nos trabalhos futuros.

Além disso, alguns requisitos importantes na aplicação não chegaram à fase de implementação, como: o questionário do doador, implementação da versão mobile, o aprimoramento da funcionalidade instituições, onde mostra os horários disponíveis para atendimento e funções de cadastro, edição e exclusão de instituições na funcionalidade de agendamento.

Assim, como resultado, as funcionalidades desenvolvidas no sistema foram: agendamento online, consulta ao histórico de doações, consulta às instituições cadastradas e emissão de alerta.

Dessa forma, espera-se que o software atenda as demandas dos bancos de sangue, consiga aumentar a captação de doadores e manter os doadores já existentes por meio da teletriagem, em que visa estreitar a comunicação do doador e da instituição hematológica. A fim de promover maior suporte aos hemocentros e hemonúcleos e incentivar à doação voluntária.

6.1 Trabalhos futuros

Como sequência desse trabalho, tem-se alguns pontos que podem ser explorados futuramente, tais como:

- Implementar critérios de dupla autenticação e segurança no sistema, com protocolos de criptografia.
- Implementar o questionário de saúde, a fim de otimizar o processo de registro na triagem presencial.
- Inserir algoritmos analíticos, com o objetivo de prever a demanda das unidades.
- Aprimorar a responsividade da versão mobile da aplicação.
- Implementar o frontend e o backend da versão mobile.
- Implementar a função de cadastrar, editar e excluir unidades hematológicas na seção instituições.
- Implementar o Banco de Dados, a fim de registrar e atualizar modificações realizadas no sistema.
- Contato contínuo com os usuários, detalhando o histórico de doações e implementar notificações push para avisar quando ele está apto para fazer a próxima doação.

REFERÊNCIAS

- ADRIANO, T. D. S. **Guia prático de TypeScript: Melhore suas aplicações JavaScript**. São Paulo: Casa do Código, 2021. Citado na página 19.
- ALEXANDRE, R. C. J. et al. Fidelização de doadores de sangue através de aplicativo móvel: Mecanismos para o incentivo à doação recorrente loyalty of blood donors through a mobile app: Mechanisms to encourage recurrent donation. 2023. Citado na página 11.
- Angular. **Angular**. 2023. Acesso em: 05 dez. 2023. Disponível em: <<https://angular.io/>>. Citado na página 17.
- Apache Cordova. **Apache Cordova**. 2023. Acesso em: 23 dez. 2023. Disponível em: <<https://cordova.apache.org/docs/en/3.1.0/guide/overview/>>. Citado na página 18.
- Apple. **Store**. 2023. Acessado em: 13 de outubro de 2023. Disponível em: <<https://www.apple.com/br/app-store/>>. Citado na página 15.
- Apple Inc. **iOS Overview**. 2023. Acessado em: 13 de maio de 2011. Disponível em: <http://developer.apple.com/library/ios/referencelibrary/GettingStarted/URL_iPhone_OS_Overview/index.html//apple_ref/doc/uid/TP40007592>. Citado na página 15.
- BATISTA, D. V. **Desenvolvimento front-end de aplicativo para busca de serviços de saúde**. Campina Grande: [s.n.], 2022. 80f p. Trabalho de Conclusão de Curso (Graduação em Computação). Citado na página 26.
- BITTENCOURT, J.; MURILO, C. **O que é um framework?** 2021. Acesso em: 08 out. 2022. Disponível em: <<https://www.alura.com.br/artigos/framework-o-que-e-pra-que-serve-essaferramenta/>>. Citado na página 16.
- DE, R. **Teletriagem: Há Benefícios Para Os Sistemas De Saúde E Seus Usuários?** 2017. Citado na página 11.
- DEITEL, P. et al. **Android para Programadores: uma abordagem baseada em aplicativos, 1ª Edição**. [S.l.]: Editora Bookman, 2013. Citado na página 14.
- FERNANDES, D. **TypeScript: Vantagens, mitos, dicas e conceitos fundamentais**. 2019. Acesso em: 13 out. 2022. Disponível em: <<https://blog.rocketseat.com.br/typescript-vantagens-mitosconceitos/>>. Citado na página 19.
- FGV. **Uso de TI no Brasil: País tem mais dois dispositivos digitais por habitante, revela pesquisa**. 2023. Disponível em: <<https://portal.fgv.br/noticias/uso-ti-brasil-pais-tem-mais-dois-dispositivos-digitais-habitante-revela-pesquisa>>. Citado na página 10.
- GOADRICH, M. H.; ROGERS, M. P. Smart smartphone development: ios versus android. In: **Proceedings of the 42nd ACM technical symposium on Computer science education**. [S.l.: s.n.], 2011. p. 607–612. Citado na página 13.
- HEMOBRÁS. **HEMOBRÁS**. 2023. Acesso em: 03 out. 2023. Disponível em: <<https://hemobras.gov.br/>>. Citado na página 21.

Hemogram. **Hemogram**. 2023. Acessado em: 16 outubro, 2023. Disponível em: <<https://hemogram.com.br/>>. Citado na página 23.

HEMOREDE. **HEMOREDE**. 2023. Acesso em: 03 out. 2023. Disponível em: <<https://www.hemocentro.unicamp.br/hemorrede/>>. Citado na página 20.

HEMORREDE. **Mapa do HemorRede do Rio de Janeiro**. 2023. Acessado em: 14 fevereiro de 2025. Disponível em: <https://www.hemorio.rj.gov.br/html/Hemorrede_mapa.htm>. Citado na página 20.

HEMOVIDA. **Sistema de Gerenciamento em Serviços de Hemoterapia**. 2023. Acessado em: 05 novembro de 2023. Disponível em: <<http://siab.datasus.gov.br/DATASUS/index.php?area=060503>>. Citado na página 22.

Hostgator. **Framework: o que é, quais utilizar e como eles funcionam!** 2021. Acesso em: 03 out. 2022. Disponível em: <<https://www.hostgator.com.br/blog/frameworks-na-programacao/>>. Citado na página 16.

Ionic. **Introdução ao Ionic**. 2023. Acesso em: 18 set. 2023a. Disponível em: <<https://ionicframework.com/docs/>>. Citado na página 18.

Kenzie. **Bootstrap**. 2022. Acesso em: 16 out. 2022. Disponível em: <<https://kenzie.com.br/blog/bootstrap/>>. Citado na página 16.

LECHETA, R. R. **Android Essencial com Kotlin**. [S.l.]: Novatec Editora, 2017. Citado na página 14.

LEITE, C. R. M.; ROSA, S. d. S. R. F. **Novas tecnologias aplicadas à saúde: integração de áreas transformando a sociedade**. UERN, 2017. Citado na página 10.

LIMA, D. W. M. d. et al. **Ciência da computação: análise comparativa de frameworks javascript para front-end**. Cruzeiro do Sul Educacional, 2023. Citado na página 10.

MENDONÇA, V. R. L. D.; BITTAR, T. J.; DIAS, M. D. S. **Um estudo dos Sistemas Operacionais Android e iOS para o desenvolvimento de aplicativos**. 2011. 2016. Citado na página 15.

Ministério da Saúde. **Ministério da Saúde lança campanha para incentivar doação de sangue**. 2023. Acesso em: 22 out. 2023. Disponível em: <<https://www.gov.br/saude/pt-br/assuntos/noticias/2023/junho/ministerio-da-saude-lanca-campanha-para-incentivar-doacao-de-sangue>>. Citado na página 11.

ROVEDA, U. **O QUE É BOOTSTRAP, COMO USAR E SUAS FUNCIONALIDADES**. 2022. Acesso em: 16 out. 2022. Disponível em: <<https://kenzie.com.br/blog/bootstrap/>>. Citado na página 18.

SARDI, L. et al. **Blood4life: a mobile solution to recruit and retain blood donors through gamification and trans-theoretical model**. In: SPRINGER. **New Knowledge in Information Systems and Technologies: Volume 3**. [S.l.], 2019. p. 3–12. Citado na página 11.

Stack Overflow. **Frameworks Web e Tecnologias**. 2023. Acesso em: 23 nov. 2023. Disponível em: <<https://survey.stackoverflow.co/2023/>>. Citado nas páginas 16 e 17.

Statcounter. **Statcounter**. 2023. Acesso em: 29 nov. 2023. Disponível em: <<https://gs.statcounter.com/>>. Citado na página 14.

Statcounter. **Statcounter**. 2024. Acesso em: 03 out. 2024. Disponível em: <<https://gs.statcounter.com/>>. Citado na página 30.

TANENBAUM, A. S. **Sistemas Operacionais Modernos**. [S.l.]: Pearson Prentice Hall, 2008. 1-15 p. Tradução de: Modern Operating Systems. Revisão técnica por Regina Borges de Araujo. Citado na página 13.

Um Só Sangue. **Um Só Sangue**. 2023. Acessado em: 27 de novembro de 2023. Disponível em: <<https://umsosangue.com.br/>>. Citado na página 24.

WASSERMAN, A. I. Software engineering issues for mobile application development. In: **Proceedings of the FSE/SDP workshop on Future of software engineering research**. [S.l.: s.n.], 2010. p. 397–400. Citado na página 13.

WATTANAPISIT, A. et al. Can mobile health apps replace gps? a scoping review of comparisons between mobile apps and gp tasks. **BMC medical informatics and decision making**, BioMed Central, v. 20, n. 1, p. 1–11, 2020. Citado na página 10.

APÊNDICE A – CÓDIGOS

O código pode ser encontrado na seguinte página do github, **clique aqui** ou acesse pelo link <https://github.com/raquelsilveiraa/Frontend-Vida.git>.

login.page.html

```

    <ion-content>
  <div class="container">

    <div class="esquerda">

      <p>Faça o login para acessar a área do administrador</p>
    </div>

    <div class="direita">
      <div class="bloco-login">
        <h1>Bem-vindo(a)!</h1>
        <form [formGroup]="formulario" (ngSubmit)="login()">
          <ion-item lines="none">
            <ion-label position="stacked"
              ⇨ color="light">Login</ion-label>
            <ion-input
              formControlName="login"
              type="text"
              required
            ></ion-input>
          </ion-item>

          <ion-item lines="none">
            <ion-label position="stacked"
              ⇨ color="light">Senha</ion-label>
            <ion-input
              formControlName="senha"
              type="password"

```

```
        required
      ></ion-input>
    </ion-item>

    <a class="esqueceu" (click)="esqueceuSenha()">Esqueceu
    ↪ a senha?</a>

    <ion-button expand="block"
    ↪ type="submit">ENTRAR</ion-button>
  </form>
</div>
</div>
</div>
</ion-content>
```

login.page.scss

```
:host {
  ion-content {
    --background: transparent;
  }

  .container {
    display: flex;
    height: 100vh;
  }

  .esquerda {
    flex: 1;
    background: #eaf4ff;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
```

```
img {
  max-width: 100%;
  margin-bottom: -0.1rem;
}

p {
  color: #7a9cb9;
  font-size: 1.5rem;
}

.direita {
  flex: 1;
  background: #006aff;
  display: flex;
  align-items: center;
  justify-content: center;

  .bloco-login {
    width: 80%;
    max-width: 380px;
    background: rgba(255, 255, 255, 0.15);
    border-radius: 16px;
    padding: 2rem;
    color: #fff;

    h1 {
      text-align: center;
      margin-bottom: 1.5rem;
      font-weight: 600;
    }

    ion-item {
      margin-bottom: 0.75rem;
    }

    a.esqueceu {
```

```
        display: block;
        text-align: center;
        font-size: 0.85rem;
        text-decoration: underline;
        color: #fff;
        margin-bottom: 1rem;
    }

    ion-button {
        margin-top: 0.5rem;
    }
}
}
```

login.page.ts

```
import { Component } from '@angular/core';
import { IonicModule, AlertController } from '@ionic/angular';
import { ReactiveFormsModule, FormBuilder, FormGroup, Validators
↵ } from '@angular/forms';
import { RouterModule, Router } from '@angular/router';
import { CommonModule } from '@angular/common';

@Component({
    selector: 'app-login',
    standalone: true,
    imports: [CommonModule, IonicModule, ReactiveFormsModule],
    templateUrl: './login.page.html',
    styleUrls: ['./login.page.scss'],
})
export class LoginPage {
    formulario: FormGroup;

    constructor(
        private constructor: FormBuilder,
```

```
private navegacao: Router,
private sinalAlerta: AlertController
) {
  this.formulario = this.construtor.group({
    login: ['', Validators.required],
    senha: ['', Validators.required],
  });
}

async login() {
  const { login, senha } = this.formulario.value;
  if (login === 'admin' && senha === '12345') {

    this.navegacao.navigateByUrl('/home', { replaceUrl: true
    ↪ });
  } else {
    const alert = await this.sinalAlerta.create({
      header: 'Erro',
      message: 'Usuário ou senha inválidos',
      buttons: ['OK']
    });
    await alert.present();
  }
}

async esqueceuSenha(): Promise<void> {
  const alert = await this.sinalAlerta.create({
    header: 'Recuperar Senha',
    message: 'Digite o seu e-mail para recuperar senha:',
    inputs: [
      {
        name: 'email',
        type: 'email',
        placeholder: 'admin@gmail.com'
      }
    ],
    buttons: [
      {
```

```
        text: 'Cancelar',
        role: 'cancel'
      },
      {
        text: 'Enviar',
        handler: (data) => {

          console.log('E-mail para recuperação:', data.email);

        }
      }
    ]
  });

  await alert.present();
}
```

home.page.html

```
<ion-split-pane contentId="main-content">

  <ion-menu contentId="meio-menu" type="overlay">
    <ion-content class="fundo-menu">
      <div class="icone-menu">
        
      </div>

      <ion-list>

        <ion-menu-toggle auto-hide="false">
          <ion-item routerLink="/home" lines="none"
            ↪ detail="false" class="funcionalidades">
```

```
        
        <ion-label>HOME</ion-label>
    </ion-item>
</ion-menu-toggle>

<ion-menu-toggle auto-hide="false">
    <ion-item routerLink="/instituicoes" lines="none"
    ↪ detail="false" class="funcionalidades">
        
        <ion-label> INSTITUIÇÃO</ion-label>
    </ion-item>
</ion-menu-toggle>

<ion-menu-toggle auto-hide="false">
    <ion-item routerLink="/agendamento" lines="none"
    ↪ detail="false" class="funcionalidades">
        
        <ion-label>AGENDAMENTO</ion-label>
    </ion-item>
</ion-menu-toggle>

<ion-menu-toggle auto-hide="false">
    <ion-item routerLink="/notificacoes" lines="none"
    ↪ detail="false" class="funcionalidades">
        
        <ion-label>EMITIR ALERTA</ion-label>
    </ion-item>
</ion-menu-toggle>

</ion-list>
</ion-content>
```

```
</ion-menu>

<div class="ion-page" id="main-content">
  <ion-header>
    <ion-toolbar color="light">
      <ion-buttons slot="start">
        <ion-menu-button></ion-menu-button>
      </ion-buttons>
      <ion-title>HOME</ion-title>
      <ion-buttons slot="end">
        <ion-button (click)="logout()" fill="clear">
          
        </ion-button>
      </ion-buttons>
    </ion-toolbar>
  </ion-header>

  <!-- Exibe lista de Alertas Recentes -->
  <ion-list *ngIf="alertas.length > 0">
    <ion-list-header>
      <ion-label>Alertas Recentes</ion-label>
    </ion-list-header>
    <ion-item *ngFor="let n of alertas">
      <ion-label>
        <h2>{{ n.tipoSanguineo }} {{ n.rh }} <input type="checkbox" /> {{
          ↪ n.hemocentro }}</h2>
        <p>{{ n.timestamp | date:'short' }}</p>
      </ion-label>
    </ion-item>
  </ion-list>

  <ion-content>

  <!-- ===== CARD "Quem pode doar?" ===== -->
  <ion-card class="quem-pode-doar-card">
```

```
<!-- Título Azul -->
<ion-card-header>
  <ion-card-title>
    REQUISITOS PARA SER UM DOADOR
  </ion-card-title>
</ion-card-header>

<!-- Conteúdo com fundo claro -->
<ion-card-content>
  <h2>Quem pode doar?</h2>
  <ul>
    <li>Estar em boas condições de saúde.</li>
    <li>Ter entre 18 e 69 anos, desde que a primeira
      ↳ doação tenha sido feita até 60 anos.</li>
    <li>Pesar no mínimo 50kg.</li>
    <li>Estar descansado (ter dormido pelo menos 6 horas
      ↳ nas últimas 24 horas).</li>
    <li>Estar alimentado (evitar alimentação gordurosa
      ↳ nas 4 horas que antecedem a doação).</li>
    <li>
      Apresentar documento original com foto recente,
      ↳ que permita a identificação do candidato,
      ↳ emitido por órgão oficial:
      <ul class="sub-list">
        <li>RG [ ] Carteira de Identidade física ou cópia
          ↳ autenticada; RG [ ] Carteira de Identidade
          ↳ digital;</li>
        <li>CNH [ ] Carteira Nacional de Habilitação
          ↳ física, com foto e filiação; CNH [ ] Carteira
          ↳ Nacional de Habilitação digital, com a
          ↳ presença do QR Code.</li>
      </ul>
    </li>
  </ul>
</ion-card-content>
</ion-card>
</ion-content>
</div>
```

```
</ion-split-pane>
```

```
home.page.scss
```

```
    :host {

    ion-menu {

        width: 50px;

    ion-menu .meio-menu {
        --background: #006aff;
    }

    ion-content.meio-menu{
        --background: #006aff;
    }

    ion-content.fundo-menu {
        --background: #006aff;
        background: #006aff;
        background-color:#006aff;
    }

    .icone-menu {
        border-bottom: 3px solid rgba(255, 255, 255, 0.3);
        display: flex;
        justify-content: flex-start;
        align-items: center;
        height: 54px;
        padding: 0.5rem;
    }

    .icone-menu .menu{
```

```
    width: 30px;
    height: 30px;
    object-fit: contain;
}

ion-list {
    margin-top: 7rem;
    --background: transparent;
    background-color: #006aff ;

    .funcionalidades{

        --background: transparent;
        --border-radius: 0;
        padding-left: 1rem;
        height: 50px;
        margin-bottom: 1cm;
        display: flex;
        align-items: center;

        gap: 0.75rem;

        ion-icon{
            --background: #006aff;
            background-color: #006aff;
            color: #006aff;
        }

        ion-label {

            color: #ffffff;
            font-weight: 400;
            font-size: 0.95rem;
        }
    }
}
```

```
    }

    &:hover {
        background: rgba(255, 255, 255, 0.1);
    }

    &.activated {
        background: rgba(255, 255, 255, 0.2);
    }

    .icones {
        width: 30px;
        height: 30px;
        object-fit: contain;
        margin-right: 1cm;
    }
}

}

}

//Conteudo da Pagina Home//

.card-frame {
    margin: 1rem;
    border: 1px solid #e0e0e0;
    box-shadow: none;
    padding-bottom: 1rem;
}

.card-title {
    background: #006aff;
    color: #fff;
    font-weight: 600;
    padding: 0.8rem 1rem;
    border-radius: 4px 4px 0 0;
    margin: -1px -1px 1rem -1px;
```

```
}

#main-content {
  ion-header {
    --background: #ffffff;
    ion-toolbar {
      --background: #006aff;
      --color: #333333;
    }
  }
}

ion-content {
  --background: #f5f9ff;
  padding: 1rem;

  .page-header {
    margin-bottom: 1.5rem;
    display: flex;
    align-items: baseline;
    gap: 0.5rem;

    h1 {
      font-size: 1rem;
      font-weight: 100;
      margin: 25px;
      color: #333;
    }
  }

  .requisitos-card {
    margin: 5rem;
    border-radius: 10px;
    background-color: #ffffff;
    box-shadow: 0 2px 6px rgba(0, 0, 0, 0.1);
    overflow: visible;
  }
}
```

```
ion-card-header {
  background-color: #006aff;
  padding: 1rem;
  --background: #006aff;

  ion-card-title {
    color: #006aff;
    font-size: 1.125rem;
    font-weight: 600;
    margin: 0;
  }

  border-bottom: 1px solid rgba(0, 0, 0, 0.1);
}

ion-card-content {
  background-color: #f5f9ff;
  padding: 1rem;

  h2 {
    color: #333333;
    font-size: 1rem;
    margin-top: 0;
    margin-bottom: 0.75rem;
    font-weight: 500;
  }

  ul {
    list-style: none;
    padding-left: 0;

    li {
      color: #555555;
      font-size: 0.9rem;
      margin-bottom: 0.5rem;
      line-height: 1.4;
    }
  }
}
```

```
    & .sub-list {
      margin-top: 0.25rem;
      margin-left: 1rem;
    }

    & .sub-list li {
      list-style: disc;
      margin-bottom: 0.25rem;
    }
  }
}
}
}

/* ===== ESTILO DO BOTÃO DE LOGOUT NO TOOLBAR ===== */
.toolbar-icon {
  width: 24px;
  height: 24px;
  object-fit: contain;
}
}
}
}
}
```

home.page.ts

```
import { Component, OnInit } from '@angular/core';
import { CommonModule } from '@angular/common';
import { IonicModule } from '@ionic/angular';
import { RouterModule, Router } from '@angular/router';
import { AlertaService, Alerta } from '../alerta.service';
import { Subscription } from 'rxjs';
```

```
@Component({
  selector: 'app-home',
  standalone: true,
  imports: [CommonModule, IonicModule, RouterModule],
  templateUrl: './home.page.html',
  styleUrls: ['./home.page.scss'],
})

export class HomePage implements OnInit {
  alertas: Alerta[] = [];
  private sub!: Subscription;

  constructor(private router: Router, private alertaService:
    → AlertaService) {}

  ngOnInit() {

    this.sub = this.alertaService.alertas$.subscribe(lista => {
      this.alertas = lista;
    });
  }

  logout() {

    localStorage.removeItem('isLoggedIn');

    this.router.navigateByUrl('/login', { replaceUrl: true });
  }
}
```

```
instituicoes.page.html
```

```
<ion-split-pane contentId="main-content">

<ion-menu contentId="meio-menu" type="reveal">
  <ion-content class="fundo-menu">
    <div class="icone-menu">
      
    </div>

    <ion-list>

      <ion-menu-toggle auto-hide="false">
        <ion-item routerLink="/home" lines="none"
          ↪ detail="false" class="funcionalidades">

          
          <ion-label>HOME</ion-label>
        </ion-item>
      </ion-menu-toggle>

      <ion-menu-toggle auto-hide="false">
        <ion-item routerLink="/instituicoes" lines="none"
          ↪ detail="false" class="funcionalidades">
          
          <ion-label>INSTITUIÇÕES</ion-label>
        </ion-item>
      </ion-menu-toggle>

      <ion-menu-toggle auto-hide="false">
        <ion-item routerLink="/agendamento" lines="none"
          ↪ detail="false" class="funcionalidades">
          
          <ion-label>AGENDAMENTO</ion-label>
        </ion-item>
      </ion-menu-toggle>
    </ion-list>
  </ion-content>
</ion-menu>
</ion-split-pane>
```

```
        </ion-item>
    </ion-menu-toggle>

    <ion-menu-toggle auto-hide="false">
        <ion-item routerLink="/notificacoes" lines="none"
            ↳ detail="false" class="funcionalidades">
            
            <ion-label>EMITIR ALERTA</ion-label>
        </ion-item>
    </ion-menu-toggle>

    </ion-list>
</ion-content>
</ion-menu>

<div class="ion-page" id="main-content">

    <ion-header>
        <ion-toolbar>

            <ion-buttons slot="start">
                <ion-menu-button></ion-menu-button>
            </ion-buttons>

            <ion-title>INSTITUIÇÕES DE DOAÇÃO DE SANGUE</ion-title>

            <ion-buttons slot="end" class="sair">

                <ion-button fill="clear" size="small"
                    ↳ (click)="logout()">
                    Sair
                </ion-button>
            </ion-buttons>


```

```
    </ion-toolbar>
  </ion-header>

  <ion-content>

    <ion-card class="tabela">

      <ion-card-header>
        <ion-card-title>
          {{ unidades.length }} Unidades de saúde cadastradas
        </ion-card-title>
      </ion-card-header>

      <div class="crud">

        <ion-buttons>

          <ion-button (click)="onAdicionar()" slot="end"
            ↪ color="primary">
            <ion-icon slot="icon-only"
              ↪ name="add-circle-outline"></ion-icon>
          </ion-button>

          <ion-select
            interface="popover"
            placeholder="ID"
            [(ngModel)]="filtrar"
            size="small">
            <ion-select-option
              ↪ value="id">ID</ion-select-option>
            <ion-select-option
              ↪ value="nome">Nome</ion-select-option>
```

```
        <ion-select-option
          ↪ value="cidade">Cidade</ion-select-option>
      </ion-select>

</ion-buttons>

<ion-searchbar
  [(ngModel)]="buscar"
  placeholder="Pesquisar"
  debounce="400"
  showCancelButton="focus">
</ion-searchbar>

</div>

<div class="rolar-tabela">
  <table>
    <thead>
      <tr>
        <th>ID</th>
        <th>CNES</th>
        <th>Nome</th>
        <th>Instituição</th>
        <th>Categoria</th>
        <th>Cidade</th>
        <th>Bairro</th>
        <th>Telefone</th>
      </tr>
    </thead>

    <tbody>
      <tr *ngFor="let u of unidadesFiltradas">
        <td>{{ u.id }}</td>
        <td>{{ u.cnes }}</td>
        <td>{{ u.nome }}</td>
        <td>{{ u.instituicao }}</td>
        <td>{{ u.categoria }}</td>
      </tr>
    </tbody>
  </table>
</div>
```

```
        <td>{{ u.municipio }}</td>
        <td>{{ u.bairro }}</td>
        <td>{{ u.telefone }}</td>
    </tr>
</tbody>

</table>
</div>

</ion-card>

</ion-content>
</div>
</ion-split-pane>
```

instituicoes.page.scss

```
:host {

    ion-menu {
        width: 50px;
    }

    ion-menu .meio-menu {
        --background: #006aff;
    }

    ion-content.meio-menu{
        --background: #006aff;
    }

    ion-content.fundo-menu {
        --background: #006aff;
        background: #006aff;
    }
}
```

```
    background-color:#006aff;
}

.icone-menu {
  border-bottom: 3px solid rgba(255, 255, 255, 0.3);
  display: flex;
  justify-content: flex-start;
  align-items: center;
  height: 50px;
  padding: 0.5rem;
}

.icone-menu .menu{
  width: 30px;
  height: 30px;
  object-fit: contain;
}

ion-list {
  margin-top: 7rem;
  --background: transparent;
  background-color:#006aff ;

  .funcionalidades {
    --background: transparent;
    --border-radius: 0;
    color: #ffffff;
    height: 50px;
    margin-bottom: 1cm;
    display: flex;
    align-items: center;
    padding-left: 1rem;
    gap: 0.75rem;

    ion-label {
      color: #ffffff;
```

```
        font-weight: 400;
        font-size: 0.95rem;
    }

    &:hover {
        background: rgba(255, 255, 255, 0.1);
    }

    &.activated {
        background: rgba(255, 255, 255, 0.2);
    }

    .icones{
        width: 30px;
        height: 30px;
        object-fit: contain;
        margin-right: 1cm;
    }
}

ion-toolbar {
    --background: #ffffff;
    --color: #000;
    --min-height: 54px;
    font-size: 0.5rem;
}

.sair{
    align-items: center;
}

.tabela{
```

```
margin: 1rem;
border: 1px solid #e0e0e0;
box-shadow: none;

ion-card-header {
  padding-bottom: .5rem;

  ion-card-title {
    font-size: 1rem;
    font-weight: 600;
  }
}

.crud {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 0 1rem 1rem;

  ion-searchbar {
    --background: transparent;
    --border-radius: 30px;
    width: 200px;
  }

  ion-select {
    min-width: 110px;
  }

  ion-button {
    --padding-start: 0;
    --padding-end: 0;
  }
}

.rolar-tabela {
  overflow-x: auto;
```

```
}

table {
  width: 100%;
  border-collapse: collapse;
  font-size: .85rem;
  min-width: 600px;

  thead {
    background: #f0f4ff;

    th {
      text-align: left;
      padding: .75rem;
      white-space: nowrap;
    }
  }

  tbody {
    tr {
      border-bottom: 1px solid #e0e0e0;

      &:hover {
        background: #f9fbff;
      }

      td {
        padding: .65rem .75rem;
        white-space: nowrap;

        &:first-child {
          color: #006aff;
          font-weight: 600;
        }
      }
    }
  }
}
```

```
    }  
  }  
}
```

instituicoes.page.ts

```
import { Component }      from '@angular/core';  
import { CommonModule }   from '@angular/common';  
import { FormsModule }    from '@angular/forms';  
import { IonicModule, AlertController } from  
↳ '@ionic/angular';  
import { RouterModule, Router } from '@angular/router';  
  
@Component({  
  selector: 'app-instituicoes',  
  templateUrl: './instituicoes.page.html',  
  styleUrls: ['./instituicoes.page.scss'],  
  standalone: true,  
  imports: [IonicModule, CommonModule, FormsModule,  
↳ RouterModule]  
})  
export class InstituicoesPage {  
  
  filtrar = 'id';  
  buscar = '';  
  
  unidades = [  
    {  
      id: 1,  
      cnes: '9466061',  
      nome: 'HEMATO',  
      instituicao: 'HEMOCENTRO',  
      categoria: 'CENTRO DE ATENÇÃO HEMATOLOGICA',  
      municipio: 'JOÃO PESSOA',
```

```
    bairro: 'TORRE',
    telefone: '(83) 3434-2999'
  },

  {
    id: 2,
    cnes: '2606879',
    nome: 'HEMOCENTRO DE SOUSA',
    instituicao: 'HEMOCENTRO',
    categoria: 'CENTRO DE ATENÇÃO HEMATOLOGICA',
    municipio: 'SOUSA',
    bairro: 'JOSE FACUNDO DE LIRA',
    telefone: '(83) 3522-6364'
  },

  {
    id: 3,
    cnes: '2399598',
    nome: 'HEMOCENTRO DO ESTADO DA PARAIBA',
    instituicao: 'HEMOCENTRO',
    categoria: 'CENTRO DE ATENÇÃO HEMATOLOGICA',
    municipio: 'JOÃO PESSOA',
    bairro: 'CENTENARIO',
    telefone: '(83) 3133-3473'
  },

  {
    id: 4,
    cnes: '2612828',
    nome: 'HEMOCENTRO REGIONAL DE CAMPINA GRANDE',
    instituicao: 'HEMOCENTRO',
    categoria: 'CENTRO DE ATENÇÃO HEMATOLOGICA',
    municipio: 'CAMPINA GRANDE',
    bairro: 'CATOLE',
    telefone: '(83) 3310-7173'
  },

  {
```

```
    id: 5,
    cnes: '2613204',
    nome: 'HEMONUCLEO',
    instituicao: 'HEMONUCLEO',
    categoria: 'CENTRO DE ATENÇÃO HEMATOLOGICA',
    municipio: 'GUARABIRA',
    bairro: 'CENTRO',
    telefone: '(83) 3271-3001'
  },

  {
    id: 6,
    cnes: '2606321',
    nome: 'HEMONUCLEO DE CATOLE DO ROCHA',
    instituicao: 'HEMONUCLEO',
    categoria: 'CENTRO DE ATENÇÃO HEMATOLOGICA',
    municipio: 'CATOLE DO ROCHA',
    bairro: 'BATALHÃO',
    telefone: '(83) 3441-2281'
  },

  {
    id: 7,
    cnes: '2794047',
    nome: 'HEMONUCLEO DE ITAPORANGA',
    instituicao: 'HEMONUCLEO',
    categoria: 'CENTRO DE ATENÇÃO HEMATOLOGICA',
    municipio: 'ITAPORANGA',
    bairro: 'CENTRO',
    telefone: '(83) 3451-3819'
  },

  {
    id: 8,
    cnes: '2363410',
    nome: 'HEMONUCLEO DE MONTEIRO',
    instituicao: 'HEMONUCLEO',
    categoria: 'CENTRO DE ATENÇÃO HEMATOLOGICA',
```

```
    municipio: 'MONTEIRO',
    bairro: 'CENTRO',
    telefone: '(83) 3351-2004'
  },

  {
    id: 9,
    cnes: '2794055',
    nome: 'HEMONUCLEO DE PICUI',
    instituicao: 'HEMONUCLEO',
    categoria: 'CENTRO DE ATENÇÃO HEMATOLOGICA',
    municipio: 'PICUI',
    bairro: 'MONTE SANTO',
    telefone: '(83) 3371-2554'
  },

  {
    id: 10,
    cnes: '2794063',
    nome: 'HEMONUCLEO DE PRINCESA IZABEL',
    instituicao: 'HEMONUCLEO',
    categoria: 'CENTRO DE ATENÇÃO HEMATOLOGICA',
    municipio: 'PRINCESA IZABEL ',
    bairro: 'ALTO DA CASCAVEL',
    telefone: '(83) 3457-2938'
  },

  {
    id: 11,
    cnes: '2399598',
    nome: 'HEMONUCLEO DE ITABAINA',
    instituicao: 'HEMONUCLEO',
    categoria: 'CENTRO DE ATENÇÃO HEMATOLOGICA',
    municipio: 'ITABAIANA',
    bairro: 'NOVA ITABAIANA',
    telefone: '(83) 3281-2004'
  },
}
```

```
{
  id: 12,
  cnes: '2794055',
  nome: 'HEMONUCLEO DE PIANCÓ',
  instituicao: 'HEMONUCLEO',
  categoria: 'CENTRO DE ATENÇÃO HEMATOLOGICA',
  municipio: 'PIANCÓ',
  bairro: 'OURO BRANCO',
  telefone: '(83) 3452-2733'
},

{
  id: 13,
  cnes: '2606321',
  nome: 'HEMONUCLEO DE CAJAZEIRAS',
  instituicao: 'HEMONUCLEO',
  categoria: 'CENTRO DE ATENÇÃO HEMATOLOGICA',
  municipio: 'CAJAZEIRAS',
  bairro: 'CENTRO',
  telefone: '(83) 3531-6687'
},

{
  id: 14,
  cnes: '2399598',
  nome: 'HEMONUCLEO DE PATOS',
  instituicao: 'HEMONUCLEO',
  categoria: 'CENTRO DE ATENÇÃO HEMATOLOGICA',
  municipio: 'PATOS',
  bairro: 'BELO HORIZONTE-BELA VISTA',
  telefone: '(83) 3423-2180'
},

];
```

```
constructor(private router: Router, private alertCtrl:
↳ AlertController) {}

get unidadesFiltradas() {

    if (!this.buscar || this.buscar.trim() === '') {
        return this.unidades;
    }

    const busca = this.buscar.trim().toLowerCase();

    return this.unidades.filter((uni) => {
        switch (this.filtrar) {
            case 'id':

                return uni.id.toString().includes(busca);

            case 'nome':
                return uni.nome.toLowerCase().includes(busca);

            case 'municipio':
                return uni.municipio.toLowerCase().includes(busca);

            default:
                return true;
        }
    });
}

logout() {

    localStorage.removeItem('isLoggedIn');

    this.router.navigateByUrl('/login', { replaceUrl: true });
}

onAdicionar() { }

}
```

notificacoes.page.html

```
<ion-split-pane contentId="main-content">

  <ion-menu contentId="meio-menu" type="reveal">
    <ion-content class="fundo-menu">
      <div class="icone-menu">
        
      </div>

      <ion-list>

        <ion-menu-toggle auto-hide="false">
          <ion-item routerLink="/home" lines="none"
            ↳ detail="false" class="funcionalidades">
            
            <ion-label>HOME</ion-label>
          </ion-item>
        </ion-menu-toggle>

        <ion-menu-toggle auto-hide="false">
          <ion-item routerLink="/instituicoes" lines="none"
            ↳ detail="false" class="funcionalidades">
            
            <ion-label>INSTITUIÇÕES</ion-label>
          </ion-item>
        </ion-menu-toggle>

        <ion-menu-toggle auto-hide="false">
```

```
<ion-item routerLink="/agendamento" lines="none"
  ↳ detail="false" class="funcionalidades">
  
  <ion-label>AGENDAMENTO</ion-label>
</ion-item>
</ion-menu-toggle>

<ion-menu-toggle auto-hide="false">
  <ion-item routerLink="/notificacoes" lines="none"
    ↳ detail="false" class="funcionalidades">
    
    <ion-label>EMITIR ALERTA </ion-label>
  </ion-item>
</ion-menu-toggle>
</ion-list>
</ion-content>
</ion-menu>

<div class="ion-page" id="main-content">

  <ion-header>
    <ion-toolbar>
      <ion-buttons slot="start">
        <ion-menu-button></ion-menu-button>
      </ion-buttons>
      <ion-title>EMITIR ALERTA</ion-title>
      <ion-buttons slot="end">
        <ion-button fill="clear" size="small"
          ↳ (click)="logout()">SAIR</ion-button>
      </ion-buttons>
    </ion-toolbar>
  </ion-header>

  <ion-content>
```

```
<ion-card class="central-cartao">

  <div class="titulo-cartao">EMITIR ALERTA</div>

  <form [formGroup]="Alerta" (ngSubmit)="emitirAlerta()"
  ↪ class="formulario-alerta">

    <ion-item>
      <ion-label position="stacked">Hemocentro</ion-label>
      <ion-select interface="popover"
        FormControlName="hemocentro"
        placeholder="SELECIONE">
        <ion-select-option *ngFor="let h of hemocentros"
        ↪ [value]="h">
          {{ h }}
        </ion-select-option>
      </ion-select>
    </ion-item>

    <ion-item>
      <ion-label position="stacked">Tipo Sanguíneo em
      ↪ Carência</ion-label>
      <ion-select interface="popover"
        FormControlName="tipoSanguineo"
        placeholder="SELECIONE">
        <ion-select-option *ngFor="let tipo of
        ↪ tiposSanguineos" [value]="tipo">
          {{ tipo }}
        </ion-select-option>
      </ion-select>
    </ion-item>
```

```
<div class="aparencia-rh">
  <div class="titulo-rh">Fator RH:</div>
  <ion-item lines="none">
    <ion-checkbox slot="start"
      ↪ formControlName="rhPos"></ion-checkbox>
    <ion-label>+</ion-label>
  </ion-item>
  <ion-item lines="none">
    <ion-checkbox slot="start"
      ↪ formControlName="rhNeg"></ion-checkbox>
    <ion-label>-</ion-label>
  </ion-item>
</div>

<ion-button expand="block" type="submit"
  ↪ [disabled]="Alerta.invalid">
  CONFIRMAR
</ion-button>
</form>
</ion-card>
<ion-list *ngIf="notificacoes.length > 0">
  <ion-list-header>
    <ion-label>Alertas Recentes</ion-label>
  </ion-list-header>
  <ion-item *ngFor="let n of notificacoes">
    <ion-label>
      <h2>{{ n.tipoSanguineo }} {{ n.rh }}</h2>
      <p>{{ n.timestamp | date:'short' }}</p>
    </ion-label>
  </ion-item>
</ion-list>
</ion-content>
</div>
</ion-split-pane>
```

notificacoes.page.scss

```
:host {  
  
  ion-menu {  
    width: 50px;  
  }  
  
  ion-menu .meio-menu{  
    --background: #006aff;  
  }  
  
  ion-content.meio-menu{  
    --background: #006aff;  
  }  
  
  ion-content.fundo-menu{  
    --background: #006aff;  
    background: #006aff;  
    background-color:#006aff;  
  }  
  
  .icone-menu {  
    border-bottom: 3px solid rgba(255, 255, 255, 0.3);  
    display: flex;  
    justify-content: flex-start;  
    align-items: center;  
    height: 54px;  
    padding: 0.5rem;  
  }  
  .icone-menu .menu{  
    width: 30px;  
    height: 30px;  
    object-fit: contain;
```

```
}

ion-list {
  margin-top: 7rem;
  --background: transparent;
  background-color:#006aff ;

  .funcionalidades {
    --background: transparent;
    --border-radius: 0;
    color: #ffffff;
    height: 50px;
    margin-bottom: 1cm;
    display: flex;
    align-items: center;
    padding-left: 1rem;
    gap: 0.75rem;

    ion-label {
      color: #ffffff;
      font-weight: 400;
      font-size: 0.95rem;
    }

    &:hover {
      background: rgba(255, 255, 255, 0.1);
    }

    &.activated {
      background: rgba(255, 255, 255, 0.2);
    }

    .icones{
      width: 30px;
    }
  }
}
```

```
        height: 30px;
        object-fit: contain;
        margin-right: 1cm;
    }
}
}

//Conteudo Principal de Emitir Alerta

ion-list-header{
    background-color: #e0e0e0;

}

ion-toolbar {
    --background: #ffffff;
    --color: #000;
    --min-height: 56px;
}

.central-cartao{
    margin: 1rem;
    border: 1px solid #e0e0e0;
    box-shadow: none;
    padding-bottom: 1rem;

}

.titulo-cartao{
    background: #006aff;
    color: #fff;
    font-weight: 600;
    padding: 0.8rem 1rem;
    border-radius: 4px 4px 0 0;
    margin: -1px -1px 1rem -1px;
}
```

```
}

.formulario-alerta{
  max-width: 380px;
  margin: 0 auto;
}

.formulario-alerta ion-item {
  --inner-padding-end: 0;
  margin-bottom: 1rem;
}

.aparencia-rh {
  display: flex;
  align-items: center;
  margin-bottom: 1.5rem;

  .titulo-rh {
    flex: 1;
    font-weight: 500;
    padding-top: 0.5rem;
  }
}

ion-item {
  --inner-padding-start: 0;
  --inner-padding-end: 0;
  width: 60px;
  border: 10px;
  margin-left: 1rem;

  ion-checkbox {
    margin-right: 0.5rem;
  }
}
}
```

```
notificacoes.page.ts
```

```
// src/app/notificacoes/notificacoes.page.ts
import { Component, OnInit }
  ↪ from '@angular/core';
import { CommonModule }
  ↪ from '@angular/common';
import { IonicModule }
  ↪ from '@ionic/angular';
import { ReactiveFormsModule, FormBuilder, FormGroup, Validators
  ↪ } from '@angular/forms';
import { RouterModule, Router }
  ↪ from '@angular/router';
import { AlertaService, Alerta }
  ↪ from '../alerta.service';

@Component({
  selector    : 'app-notificacoes',
  standalone : true,
  templateUrl: './notificacoes.page.html',
  styleUrls  : ['./notificacoes.page.scss'],
  imports    : [IonicModule, CommonModule, ReactiveFormsModule,
  ↪ RouterModule]
})
export class NotificacoesPage implements OnInit {

  Alerta!: FormGroup;

  notificacoes: Alerta[] = [];

  tiposSanguineos = ['A', 'B', 'AB', 'O'];

  hemocentros = [
    'HEMATO',
```

```
'HEMOCENTRO PARAIBA',
'HEMOCENTRO CAMPINA GRANDE',
'HEMOCENTRO PATOS',
'HEMOCENTRO SOUSA',
];

constructor(
  private fb: FormBuilder,
  private router: Router,
  private alertaService: AlertaService
) {}

ngOnInit(): void {

  this.Alerta = this.fb.group({
    tipoSanguineo: [null, Validators.required],
    rhPos          : [false],
    rhNeg          : [false],
    hemocentro    : [null, Validators.required]
  });

  this.alertaService.alertas$.subscribe(lista => {
    this.notificacoes = lista;
  });
}

logout(): void {
  localStorage.removeItem('isLoggedIn');
  this.router.navigateByUrl('/login', { replaceUrl: true });
}

emitirAlerta(): void {
  if (this.Alerta.invalid) {
    return;
  }

  const dados = this.Alerta.value;
  const fatoresRH: ('+' | '-')[] = [];
```

```
    if (dados.rhPos) fatoresRH.push('+');
    if (dados.rhNeg) fatoresRH.push('-');

    const rhSelecioneado: '+' | '-' = fatoresRH.length > 0 ?
    ↪ fatoresRH[0] : '+';

    const alerta: Alerta = {
      tipoSanguineo: dados.tipoSanguineo as string,
      rh            : rhSelecioneado,
      hemocentro: dados.hemocentro,
      timestamp    : new Date()
    };

    this.alertaService.enviarAlerta(alerta);
  }

  goToDetalhesInstituicao(id: string): void {
    this.router.navigateByUrl(`/instituicoes/${id}`);
  }
}
```

alerta.service.ts

```
    import { Injectable } from '@angular/core';
import { BehaviorSubject, Observable } from 'rxjs';

export interface Alerta {
  tipoSanguineo: string;
  rh: '+' | '-';
  hemocentro?: string;
  timestamp: Date;
}
```

```
@Injectable({
  providedIn: 'root'
})
export class AlertaService {

  private alertasSubject = new BehaviorSubject<Alerta[]>([]);

  get alertas$(): Observable<Alerta[]> {
    return this.alertasSubject.asObservable();
  }

  enviarAlerta(novoAlerta: Alerta) {
    const atuais = this.alertasSubject.value;

    this.alertasSubject.next([novoAlerta, ...atuais]);
  }

  limparAlertas() {
    this.alertasSubject.next([]);
  }
}
```