UNIVERSIDADE FEDERAL DE ALFENAS

FILLIPE TEIXEIRA SANTOS CESARINI

UM AMBIENTE DE APOIO AO PROCESSO DE GERENCIAMENTO DE DOAÇÕES DE SANGUE DA ASSOCIAÇÃO ESPERANÇAR

Alfenas/MG
2021

FILLIPE TEIXEIRA SANTOS CESARINI

UM AMBIENTE DE APOIO AO PROCESSO DE GERENCIAMENTO DE DOAÇÕES DE SANGUE DA ASSOCIAÇÃO ESPERANÇAR

Trabalho de conclusão de curso apresentado ao curso de Ciência da Computação, da Universidade Federal de Alfenas como requisito parcial à obtenção do título de Bacharelado em Ciência da Computação.

Orientadora: Mariane Moreira de Souza.

Alfenas/MG
2021

FILLIPE TEIXEIRA SANTOS CESARINI

UM AMBIENTE DE APOIO AO PROCESSO DE GERENCIAMENTO DE DOAÇÕES DE SANGUE DA ASSOCIAÇÃO ESPERANÇAR

A Banca examinadora abaixo-assinada, aprova o Trabalho apresentado como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação pela Universidade Federal de Alfenas.

_____

Profa. Mariane Moreira de Souza

Universidade Federal de Alfenas

_____

Prof. Rodrigo Martins Pagliares

Universidade Federal de Alfenas

_____

Prof. Flávio Barbieri Gonzaga

Universidade Federal de Alfenas

Alfenas/MG
2021

AGRADECIMENTOS

A mim, por sempre batalhar pelos meus sonhos, por sempre querer ser melhor a cada dia e sempre querer chegar o mais longe possível.

A minha família, meus pais Fabiana e Domênico e meus irmãos André e Anna Gabrielle, por sempre terem me apoiado para o meu desenvolvimento profissional, e por terem me dado todo o suporte necessário para poder chegar até aqui, principalmente à minha mãe por ser minha base e minha inspiração de vida, a quem eu devo tudo.

Ao meu namorado Lucas, por sempre ter acreditado em mim e me apoiado.

A Deus por ter me abençoado durante toda minha trajetória, me dando sabedoria e discernimento para lidar com as adversidades.

RESUMO

A crescente evolução e popularização da tecnologia têm possibilitado que tarefas do dia a dia possam ser feitas com mais agilidade, automatizando processos e facilitando sua execução. No que se refere à doação de sangue, em que se lida com muitas informações, tais como dados dos doadores, doações recebidas e gerenciamento do estoque de sangue, o uso de aplicativos para dispositivos móveis se torna muito útil, seja para facilitar a organização dessas informações ou permitir um contato mais prático e direto entre instituições de doação de sangue e os doadores. Atualmente, existem no mercado algumas soluções que auxiliam nesse processo. Porém, a maioria é voltada para grandes instituições de doação de sangue ou somente para o doador poder organizar doações realizadas. O Esperançar foi desenvolvido juntamente com os membros da Associação Esperançar, instituição responsável por gerenciar doações de sangue da cidade de Alfenas e região. A solução proposta é um ambiente que apoia o gerenciamento dos processos relacionados às doações de sangue, contando com um aplicativo para o doador poder gerenciar suas próprias doações, além de permitir a comunicação entre a associação e o doador. A solução desenvolvida facilita o processo de doação, tanto para a associação quanto para o doador, deixando com que o foco seja somente salvar vidas.

Palavras-chave: Doação de Sangue. Aplicação *web*. Dispositivos móveis. *API web*. Aplicativos. Desenvolvimento de software.

ABSTRACT

The growing evolution and popularization of technology has made possible that day-to-day tasks could be done quickly, automating processes and facilitating their execution. Regarding blood donation, which deals with a lot of information, such as donor data, donations received and blood stock management, the use of applications for mobile devices becomes very useful, whether facilitating the organization of this information or allowing more practical and direct contact between blood donation institutions and donors. Currently, some solutions on the market assist in this process. However, most are aimed at large blood donation institutions or built only for the donor to be able to organize donations they made. Esperançar was developed along with the members of the Associação Esperançar, an institution responsible for managing blood donations in the city of Alfenas and its region. The proposed solution is an environment that supports the management of processes related to blood donations, with an application for the donor to manage his own donations, besides allowing communication between the association and the donor. The developed solution facilitates the donation process, both for the association and the donor, leaving the focus only on saving lives.

Keywords: Blood donations. Web application, Mobile devices. Web API. Mobile applications. Software development.

LISTA DE FIGURAS

Figura 3.1 – Tela do aplicativo “Doadores de Sangue”.....14
Figura 3.2 – Tela do aplicativo “My Blood”.....15
Figura 3.3 – Tela do aplicativo “Hemovida”.....15
Figura 3.4 – Tela do aplicativo “Partiu doar sangue”.....16
Figura 4.1 – Diagrama de implantação UML da arquitetura do Esperançar.....23
Figura 4.2 – Diagrama de entidade-relacionamento do Esperançar.....24
Figura B.1 – Tela de *login* da aplicação *web*.....34
Figura B.2 – Tela de *login* do aplicativo móvel.....35
Figura B.3 – Tela de criação de conta do aplicativo móvel.....35
Figura B.4 – *Dashboard* aplicação *web*.....36
Figura B.5 – *Dashboard* do aplicativo móvel.....36
Figura B.6 – Tela de doadores da aplicação *web* administrativa.....37
Figura B.7 – Tela de listagem de ações da aplicação *web* administrativa.....37
Figura B.8 – Tela de criação de ação da aplicação *web* administrativa.....38
Figura B.9 – Tela de visualização de ação da aplicação *web* administrativa.....38
Figura B.10 – Tela de listagem de informes da aplicação *web* administrativa.....39
Figura B.11 – Tela de criação/visualização de informe da aplicação *web* administrativa.....39
Figura B.12 – Tela de relatórios da aplicação *web* administrativa.....40
Figura B.13 – Tela de relatório exemplo da aplicação *web* administrativa.....40
Figura B.14 – Tela da *dashboard* do aplicativo para dispositivos móveis, com menu expandido.....41
Figura B.15 – Tela de informar nova doação do aplicativo para dispositivos móveis.....41
Figura B.16 – Tela de doações do aplicativo para dispositivos móveis.....42
Figura B.17 – Tela de compartilhamento de doação do aplicativo para dispositivos móveis.....42
Figura B.18 – Tela de informações do usuário no aplicativo para dispositivos móveis.....43
Figura B.19 – Tela de notificações do aplicativo para dispositivos móveis.....43
Figura B.20 – Tela de ajustes do aplicativo para dispositivos móveis.....44

Figura B.21 – Tela de enviar convite do aplicativo para dispositivos móveis..... 44
Figura C.1 – *Style Guide* do Esperançar.....45

LISTA DE TABELAS

Tabela 3.1 – Análise comparativa considerando o Esperançar e outras aplicação.....17
Tabela 4.1 – Atores do Esperançar e suas principais interações no ambiente.....19
Tabela 4.2 – Principais requisitos funcionais do Esperançar.....20
Tabela 4.3 – Requisitos não funcionais do Esperançar.....21
Tabela 5.1 – Código de chamada do método *getAll* do *front-end* da aplicação *web* do Esperançar.....27
Tabela 5.2 – Código do método *getAll* do serviço de requisições de doações do *front-end* da aplicação *web* do Esperançar.....27
Tabela 5.3 – Código do método *checkData* que realiza a chamada do método *getDonations* do serviço *DonationService* do módulo aplicativo Esperançar.....28
Tabela 5.4 – Código do método *getDonations* do serviço *DonationService* do módulo aplicativo Esperançar.....28
Tabela 5.5 – Código do método *index* do controlador da API.....28
Tabela 5.6 – Código do método *index* da camada de serviço.....29
Tabela A.1 – Requisitos funcionais do Esperançar.....32

LISTA DE ABREVIATURAS E SIGLAS

HTTP	Hypertext Transfer Protocol
API	Application Programming Interface
REST	Representational State Transfer
MVC	Model-View-Controller
SGBD	Sistema Gerenciador de Banco de Dados
APP	Aplicativo
RF	Requisito Funcional
RNF	Requisito Não Funcional
JSON	JavaScript Object Notation
EER	Enhanced Entity-Relationship
ID	Identificador

SUMÁRIO

1. INTRODUÇÃO **9**
1.1 Problema 9
1.2 Objetivo 9
1.3 Resultados 10
1.4 Metodologia 10
1.5 Organização 10
2. TECNOLOGIAS USADAS **12**
2.1. Tecnologias para desenvolvimento de aplicações *web* 12
2.2. Tecnologias para desenvolvimento de aplicativo para dispositivos móveis 13
3. TRABALHOS RELACIONADOS **14**
4. ESPERANÇAR **18**
4.1. Introdução 18
4.2. Principais atores e interações 18
4.3. Requisitos funcionais 19
4.4. Requisitos não funcionais 21
4.5. Arquitetura 22
4.6. Diagrama Entidade-Relacionamento Estendido (EER) 23
5. IMPLEMENTAÇÃO **26**
5.1 Requisições da API via Aplicação *Web* 26
5.2 Requisições da API via Aplicação Móvel 27
5.3 Controlador 28
5.4 Serviço 29
6. DISCUSSÃO, CONCLUSÃO E TRABALHOS FUTUROS **30**
7. REFERÊNCIAS BIBLIOGRÁFICAS **31**
Anexo A - Requisitos Funcionais **32**
Anexo B - Prototipação **34**
Anexo C – *Style Guide* **45**

1. INTRODUÇÃO

1.1 Problema

A Esperançar é uma associação de doadores de sangue da cidade de Alfenas e região, que auxilia na manutenção dos bancos de sangue para essas localidades. Suas tarefas são variadas, envolvendo desde o controle manual dos níveis dos bancos de sangue até a divulgação de campanhas de doação.

Esse processo acaba sendo muito trabalhoso, além da grande possibilidade de perda de informações. Isso porque o controle do número de doações recebidas, a verificação dos tipos de sangue que foram doados, a verificação dos tipos de sangue que estão em baixa, dentre outras tarefas, são feitas manualmente, por meio do uso de planilhas e documentos. Além disso, algumas regras relacionadas ao processo de doação, como o intervalo de tempo entre uma doação e outra, que deve ser observado antes de se realizar uma doação, também é contabilizado de forma manual. Considerando que o aumento da quantidade de doadores tende a acontecer com o tempo (Agência Brasil, 2017), a taxa de erros dos processos manuais pode acabar se elevando muito, tornando o gerenciamento das doações mais lento e ineficiente.

Outro problema se refere à comunicação da associação com os doadores, que depende exclusivamente do uso de redes sociais, o que gera atrasos no processo de doação, por não permitir uma comunicação direta e rápida com os doadores para repasse de informações e avisos.

1.2 Objetivo

O trabalho tem como objetivo principal melhorar as condições de trabalho dos responsáveis pelo processo de doação de sangue da Associação Esperançar. Essa melhoria se dá por meio do desenvolvimento de plataformas digitais de apoio a esse processo, tanto para administradores quanto para doadores, promovendo automação e auxiliando na disseminação de informações a respeito de doações.

1.3 Resultados

Como resultado deste trabalho, criamos um aplicativo para dispositivos móveis com sistemas operacionais Android e iOS e uma aplicação *web* administrativa que se comunicam por meio de um serviço *web*. O aplicativo funciona como uma central para o doador poder controlar suas doações realizadas e ter acesso a outras informações úteis, enquanto a aplicação *web* possui função administrativa, sendo operado pelas pessoas responsáveis da Associação Esperançar, com o objetivo de realizar tarefas, tais como enviar comunicados, controlar doações recebidas e emitir relatórios.

1.4 Metodologia

O trabalho se iniciou pelo levantamento dos requisitos juntamente com o cliente. Realizamos algumas reuniões iniciais para discutirmos esses requisitos, com base nas funcionalidades fundamentais que a aplicação *web* e o aplicativo móvel deveriam possuir.

Depois disso, o processo de desenvolvimento foi baseado na definição de *sprints*[1], que continham os requisitos, elencados de acordo com prioridades pré-definidas, iniciando o processo. Tivemos como ponto de partida, por meio da ferramenta Justinmind[2], uma etapa de prototipação da aplicação *web* e do aplicativo móvel, levando em consideração os requisitos discutidos em reunião.

Após a validação dos protótipos com o cliente, iniciamos o processo de desenvolvimento, fazendo entregas quinzenais ao final de cada *sprint*, até a apresentação da aplicação *web* e aplicativo móvel, concluídos, para o cliente.

1.5 Organização

A organização deste trabalho foi feita da seguinte forma: no Capítulo 2 são apresentadas as tecnologias utilizadas para o desenvolvimento do Esperançar. No Capítulo 3 são apresentados os trabalhos relacionados ao Esperançar, além de uma

[1] Sprints são períodos iterativos com limite de tempo em que a equipe trabalha para entregar um conjunto de itens de valor do produto (Tal, 2015).

[2] Ferramenta de prototipação para *web* e aplicativos para dispositivos móveis. Disponível em: <https://www.justinmind.com/>.

tabela comparativa das funcionalidades apresentadas nestes trabalhos com as funcionalidades do Esperançar. No Capítulo 4 são apresentados os atores e seus objetivos, os requisitos funcionais e não funcionais, a arquitetura e um diagrama EER (entidade-relacionamento estendido) que ilustra os elementos que compõem a estrutura do banco de dados do Esperançar. No Capítulo 5 são apresentados detalhes da implementação do aplicativo e da aplicação *web*, com trechos de código para exemplificar seu funcionamento. No Capítulo 6, é apresentada a conclusão do trabalho, com discussões e sugestões para trabalhos futuros que deem continuidade ao Esperançar.

2. TECNOLOGIAS USADAS

Neste capítulo são descritas as tecnologias utilizadas para o desenvolvimento do ambiente Esperançar, sendo essas classificadas como tecnologias para desenvolvimento de aplicações *web* e tecnologias para desenvolvimento de aplicações para dispositivos móveis.

2.1. Tecnologias para desenvolvimento de aplicações *web*

O *front-end* é a estrutura da aplicação responsável pela parte que o usuário de fato enxerga e interage. O *front-end* da aplicação *web* foi desenvolvido com React.JS, uma biblioteca JavaScript para criar *interfaces* de usuário. O React.JS foi desenvolvido para exibir dados em uma *interface* de usuário em larga escala, com dados que mudam com o tempo (Gackenheimer, 2015).

A estrutura responsável pela lógica de uma aplicação é o *back-end*, e é por meio dele que são efetuadas as rotinas de leitura e escrita de dados, o que é feito principalmente por meio de uma API. De um modo geral, MASSE (2011) define a API de uma aplicação como a responsável por expor um conjunto de dados e funções para facilitar interações entre programas de computador e permitir que eles troquem informações por meio de requisições HTTP (Hypertext Transfer Protocol). Todo o processamento de informações da aplicação, tanto *web* quanto para dispositivos móveis, foi feito por meio da API REST. A nomenclatura REST ou RESTful, se refere ao estilo da arquitetura que é comumente aplicada em APIs para *web services* modernos (Masse, 2011). Além da tecnologia REST, a API do Esperançar foi construída utilizando-se o *framework* [3] PHP de nome Laravel, que trabalha com uma arquitetura MVC, que facilita o desenvolvimento, oferecendo consistência e flexibilidade nos projetos (Stauffer, 2019).

O SGBD escolhido para o armazenamento de dados da aplicação foi o MySQL, que é um banco de dados completo e *open source* com capacidade para atender aplicações de pequeno a grande porte, satisfatoriamente (Milani, 2007).

[3] Um framework compreende um conjunto de classes ou funções implementadas em uma linguagem de programação específica, usadas para auxiliar o desenvolvimento de software (Gabardo, 2017).

2.2. Tecnologias para desenvolvimento de aplicativo para dispositivos móveis

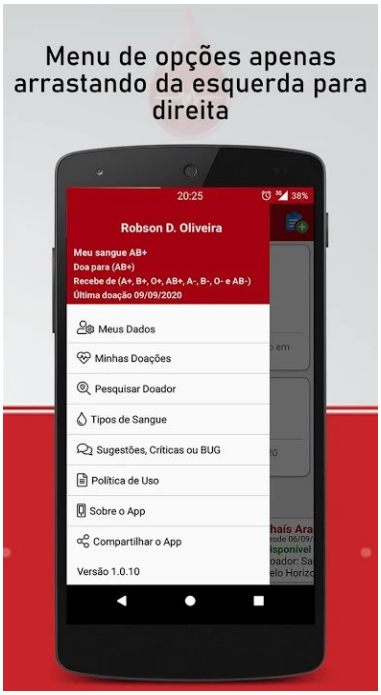
O desenvolvimento do aplicativo para dispositivos móveis foi feito utilizando React Native, que é um *framework* JavaScript para escrever aplicativos móveis nativamente renderizados para iOS e Android (Eisenman, 2016). Baseado em React.JS, o React Native facilita o desenvolvimento de aplicações móveis por permitir acesso a inúmeras bibliotecas desenvolvidas pela comunidade, além de permitir a comunicação com uma API de forma simples, devido a funções nativas do JavaScript.

3. TRABALHOS RELACIONADOS

Este capítulo apresenta alguns trabalhos relacionados que propõem soluções similares às encontradas para este trabalho e que contribuíram para o seu desenvolvimento, todos eles sendo gratuitos, assim como a solução desenvolvida para este trabalho.

O “Doadores de Sangue” é um aplicativo para Android desenvolvido por Robson D. Oliveira, que tem como principal objetivo conectar doadores de sangue com pessoas em busca de doadores. Por meio dele os usuários podem criar solicitações de doação de determinado tipo sanguíneo e doadores do tipo sanguíneo procurado podem informar doações para essas solicitações.

Figura 3.1 - Tela do aplicativo “Doadores de Sangue”

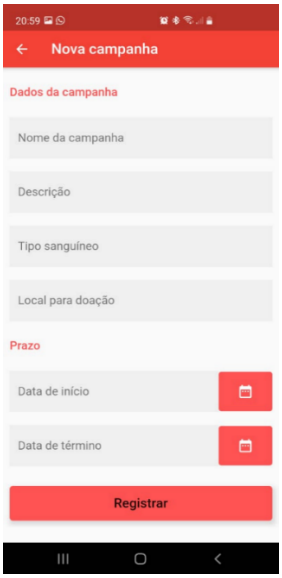


Fonte: Google Play, Doadores de Sangue

O aplicativo “My Blood”, desenvolvido por Kleber de Oliveira Andrade, assim como o aplicativo “Doadores de Sangue”, busca criar a conexão de doadores de sangue com solicitantes de doação. Nele, o usuário pode controlar suas doações realizadas, e o próprio aplicativo informa se você está apto para doar, de acordo com o prazo da sua última doação. Outras funcionalidades incluem a possibilidade de usuários criarem campanhas de doação, receber alertas de pessoas que necessitem

de doação, buscar locais de doação e também saber a melhor data para poderem doar.

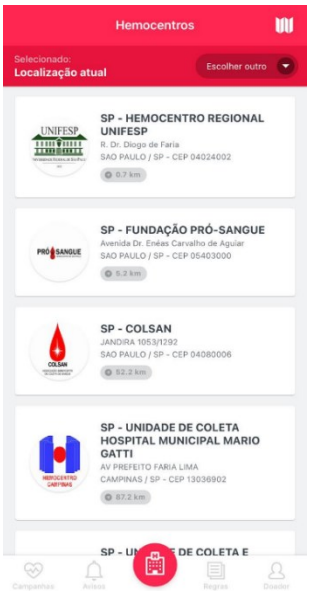
Figura 3.2 - Tela do aplicativo “My Blood”



Fonte: Google Play, My Blood

O Hemovida é um aplicativo disponível para Android e iOS desenvolvido pelo Ministério da Saúde do Governo do Brasil. Com ele, o doador pode encontrar os hemocentros mais próximos, agendar doação nos hemocentros, ter acesso ao histórico de doações, receber informativos de campanhas e ter controle de quando é sua próxima doação.

Figura 3.2 - Tela do aplicativo “Hemovida”



Fonte: Governo do Brasil, Hemovida

O aplicativo “Partiu doar sangue” tem como finalidade permitir ao usuário registrar e solicitar doações de sangue e medula. Ele permite que o doador controle suas doações realizadas, tanto de sangue quanto de medula óssea, permite a consulta de locais de doação, possui informações de campanhas de doação e possibilita a solicitação de doação de sangue e medula. O aplicativo é integrado com uma aplicação *web*, que possui a mesma funcionalidade do aplicativo.

Figura 3.2 - Tela do aplicativo “Partiu doar sangue”

Fonte: Partiu doar sangue, 2020

O Esperançar se diferencia dos demais aplicativos por ter uma aplicação *web* completamente dedicado para os administradores, possibilitando, por meio dele, enviar informativos e convites de doação para os doadores, além de possibilitar uma análise dos dados de doações. A Tabela 3.1 mostra uma análise comparativa, considerando o Esperançar e os demais aplicativos, de acordo com as principais funcionalidades oferecidas por eles. Pode-se notar que o Esperançar possui algumas das funcionalidades dos outros aplicativos, porém com o diferencial de ter um módulo *web* complementar. Funcionalidades como agendamento de doações, solicitar doação pelo aplicativo e verificar doações pelo aplicativo não entraram no escopo do trabalho pois não foram solicitadas pelo cliente, por não se encaixarem com o modelo de negócio dele.

Tabela 3.1 - Análise comparativa considerando o Esperançar e outras aplicações.

Funcionalidades	Esperançar	Doadores de Sangue	My Blood	Partiu Doar Sangue	Hemovida
Informe de doações	X	X	X	X	X
Compartilhamento de doações com foto	X				
Recebimento de informativos e alertas	X				X
Existência de aplicação *web*	X				
Agendamento de doação					X
Criação de ação de doação	X				X
Verificação de locais de doação			X	X	X
Existência de histórico de doações	X	X	X	X	X
Geração de relatórios administrativos	X				
Solicitação de doação pelo aplicativo		X	X	X	
Envio de conte para amigo(a)	X	X	X		X
Existência de painel administrativo para controle das doações recebidas	X				

Fonte: Elaborado pelo autor.

4. ESPERANÇAR

Descrevemos neste capítulo as características da aplicação *web* e do aplicativo que formam o ambiente Esperançar, incluindo seus objetivos, sua composição, atores, requisitos funcionais e não funcionais, sua arquitetura, além do resultado da etapa de prototipação, realizada antes do desenvolvimento do ambiente em si.

4.1. Introdução

O Esperançar é um ambiente hospedado em um servidor, que possui uma *interface* de aplicação *web* e um aplicativo para dispositivos móveis. O ambiente possui também o módulo servidor, uma aplicação que funciona como *back-end*, responsável pela parte lógica do sistema.

Toda a experiência do usuário de interatividade com o ambiente é realizada por meio do *front-end*, com uma versão administrativa para a *web* e também para dispositivos móveis, para usuários em geral. Além de ser responsável pela parte lógica, o módulo servidor é responsável pelo armazenamento dos dados.

Como o ambiente surgiu de uma demanda de um cliente, a Associação Esperançar, o desenvolvimento da aplicação *web* foi feito baseando-se nas funcionalidades requisitadas por eles, com o objetivo de automatizar e facilitar o trabalho manual, realizado pelos responsáveis pelo processo de doação de sangue. Além disso, outro objetivo é auxiliar os doadores, por meio de um aplicativo para dispositivos móveis, a terem uma maior autonomia para o controle de suas doações.

4.2. Principais atores e interações

O Esperançar possui dois atores, o Administrador, que acessa o sistema por meio da aplicação *web*, e o Doador, que acessa o sistema por meio do aplicativo para dispositivos móveis. A Tabela 4.1 mostra as principais interações dos atores no ambiente Esperançar.

Tabela 4.1 - Atores do Esperançar e suas principais interações no ambiente.

Código	Ator	Objetivos
INT-01	Administrador/Doador	Acessar a aplicação *web*/aplicativo por meio de *email* e senha
INT-02	Administrador	Gerenciar doadores cadastrados
INT-03	Administrador	Gerenciar ações de doação
INT-04	Administrador	Enviar informativos para os doadores
INT-05	Administrador	Gerenciar doações recebidas
INT-06	Doador	Consultar doações realizadas
INT-07	Doador	Informar e compartilhar nova doação
INT-08	Doador	Receber informativos enviados pelo Administrador
INT-09	Doador	Ser informado se já está apto a doar sangue novamente
INT-10	Doador	Enviar convite para outro usuário se tornar doador
INT-11	Administrador	Gerar relatórios
INT-12	Doador	Receber convite de doação pelo administrador

Fonte: Elaborado pelo autor.

4.3. Requisitos funcionais

De acordo com Vazquez e Simões (2016, pág. 96), os Requisitos Funcionais (RFs), descrevem o que o *software* faz, considerando uma perspectiva de tarefas e serviços de seus usuários em específico. Eles são as exigências mínimas que devem existir para que o *software* seja funcional, da forma como foi planejado.

A Tabela 4.2 apresenta uma lista dos principais RFs implementados no Esperançar, que são mencionados em outras seções deste trabalho, como no anexo B, que mostra os resultados da prototipação e na seção 5, que mostra detalhes de implementação do ambiente. Na primeira coluna da tabela temos o código do RF, na segunda coluna temos a interação à qual ele se refere (de acordo com a Tabela 4.1);

Na coluna “Requisito” é exibida a informação do RF em si e na última coluna, a sua descrição.

Tabela 4.2 - Principais requisitos funcionais do Esperançar

Código	**Objetivo**	**Requisito**	**Descrição**
RF07	INT-03	Criar ação	Deve ser possível que os administradores possam criar uma ação de doação.
RF09	INT-05	Visualizar doadores da ação	Deve ser possível que os administradores possam visualizar os doadores que fizeram doações para uma ação.
RF10	INT-03	Convidar usuário para ação	Deve ser possível que os administradores possam enviar convite para doadores ao criarem uma ação.
RF11	INT-04	Criar informativo	Deve ser possível que os administradores possam criar informativos.
RF14	INT-07	Informar doação	Deve ser possível que os doadores possam informar que fizeram uma doação.
RF16	INT-06 e INT-07	Compartilhar doação	Deve ser possível que os doadores possam compartilhar suas doações realizadas.
RF17	INT-12	Receber convite para fazer doação	Deve ser possível que os doadores possam receber convite dos administradores para fazerem doação.
RF19	INT-09	Verificar aptidão para doar	Deve ser possível que os doadores possam saber se estão aptos para doar sangue.
RF20	INT-10	Enviar convite para usuário	Deve ser possível que os doadores possam convidar outros usuários para se tornarem doadores.

Fonte: Elaborado pelo autor.

4.4. Requisitos não funcionais

De acordo com Chung (2000), a complexidade de um *software* é determinada em parte por sua funcionalidade, ou seja, o que o sistema faz, e em parte por requisitos gerais que fazem parte do desenvolvimento do *software* como custo, performance, confiabilidade, manutenção, portabilidade, custos operacionais, dentre outros. Estes requisitos são chamados de Requisitos Não Funcionais (RNFs).

Enquanto um Requisito Funcional descreve o que deve ser feito, um Requisito Não Funcional indica como deve ser feito, ou seja, a forma que o *software* tornará realidade o que está sendo planejado. Por exemplo, para qual(is) sistema(s) operacional(is) o sistema será desenvolvido, qual tecnologia de armazenamento será utilizada, como será feita a conexão entre as partes da plataforma, dentre outras.

A tabela 4.3 apresenta os principais requisitos não funcionais do Esperançar, sendo a primeira coluna o seu código, a segunda coluna referente à sua categoria e a terceira coluna uma descrição do requisito.

Tabela 4.3 - Requisitos não funcionais do Esperançar.

Código	Código	Descrição
RNF01	Usabilidade	As *interfaces* da aplicação *web* e do aplicativo móvel devem ser agradáveis e fáceis de serem utilizadas.
RNF02	Portabilidade	A aplicação *web* poderá ser acessado de qualquer plataforma e o aplicativo deverá ser utilizado em dispositivos Android ou iOS.
RNF03	Disponibilidade	A aplicação *web* e o aplicativo móvel não deverão armazenar dados localmente, sendo estes direcionados a um servidor.
RNF04		A aplicação *web* e o aplicativo móvel deverão estar conectados à internet para serem utilizados.
RNF05	Desempenho	As informações não podem levar mais do que 15 segundos para serem carregadas na aplicação *web* e no aplicativo móvel.
RNF06	Segurança	A autenticação na API deve ser feita via *token*.
RNF07		As senhas devem ser criptografadas.

RNF08	Confiabilidade	A aplicação *web* e o aplicativo móvel deverão apresentar informações de erro para o usuário em caso de problemas.
RNF09		A aplicação *web* e o aplicativo móvel deverão estar disponíveis nos momentos em que o usuário tentar acessar.
RNF10	Manutenibilidade	A aplicação *web* e o aplicativo móvel deverão permitir incluir novas funcionalidades.

Fonte: Elaborado pelo autor.

4.5. Arquitetura

O Esperançar possui uma arquitetura composta de um módulo aplicativo móvel e um módulo de servidor. O servidor possui uma aplicação *web*, por meio da qual sua lógica de negócio e seu banco de dados podem ser acessados por requisições HTTP via API. Além disso, o servidor também hospeda o *front-end* da aplicação *web*, que é acessada pelos administradores do Esperançar.

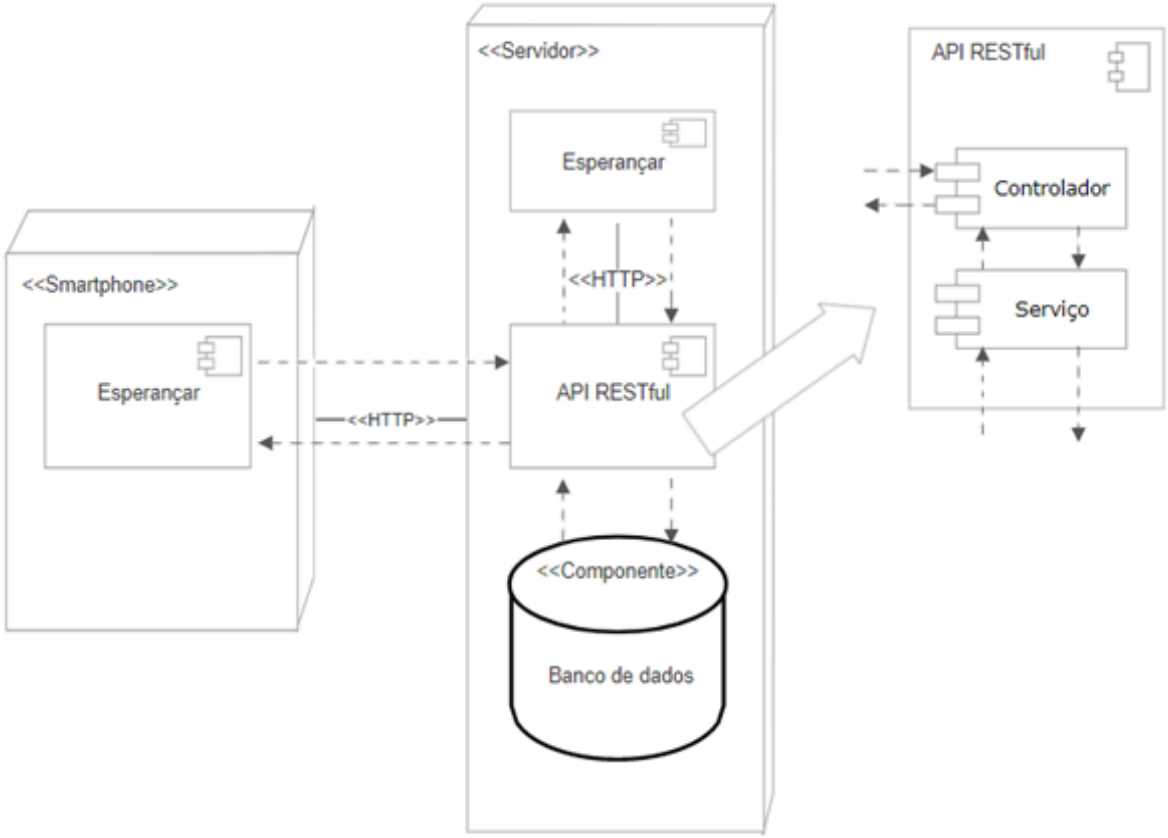
O *front-end* da aplicação *web* pode ser acessado por meio de qualquer navegador *web*, enquanto o aplicativo móvel pode ser acessado em dispositivos Android, com versão mínima API *level* 21, ou iOS com versão mínima 10.0 e com conexão à internet.

O servidor hospeda o banco de dados da aplicação, que possui os dados de doadores e doações do Esperançar, além de hospedar a API REST, que é responsável por receber as requisições HTTP, fazendo a integração entre o *front-end* da aplicação *web*, o aplicativo móvel, a lógica de negócio do Esperançar e o banco de dados.

De acordo com a Figura 4.1, supondo que um doador queira acessar suas doações de sangue, uma requisição HTTP usando o método GET é enviada para a API, passando parâmetros como token de autenticação e informações do doador. A camada que recebe a requisição é o controlador, que é o responsável pelo direcionamento daquela para a camada de serviço, de acordo com o método HTTP recebido, seja ele GET, POST, PUT ou DELETE. Em seguida, a camada de serviço acessa o banco de dados e obtém as doações do doador, retornando a lista para o controlador. Nessa lista, a informação é serializada e retornada em forma de um JSON

para o aplicativo, sendo exibida para o doador. A seção 5 aborda mais detalhes da implementação, não ilustrados na Figura 4.1.

Figura 4.1 - Diagrama de implantação UML da arquitetura do Esperançar.



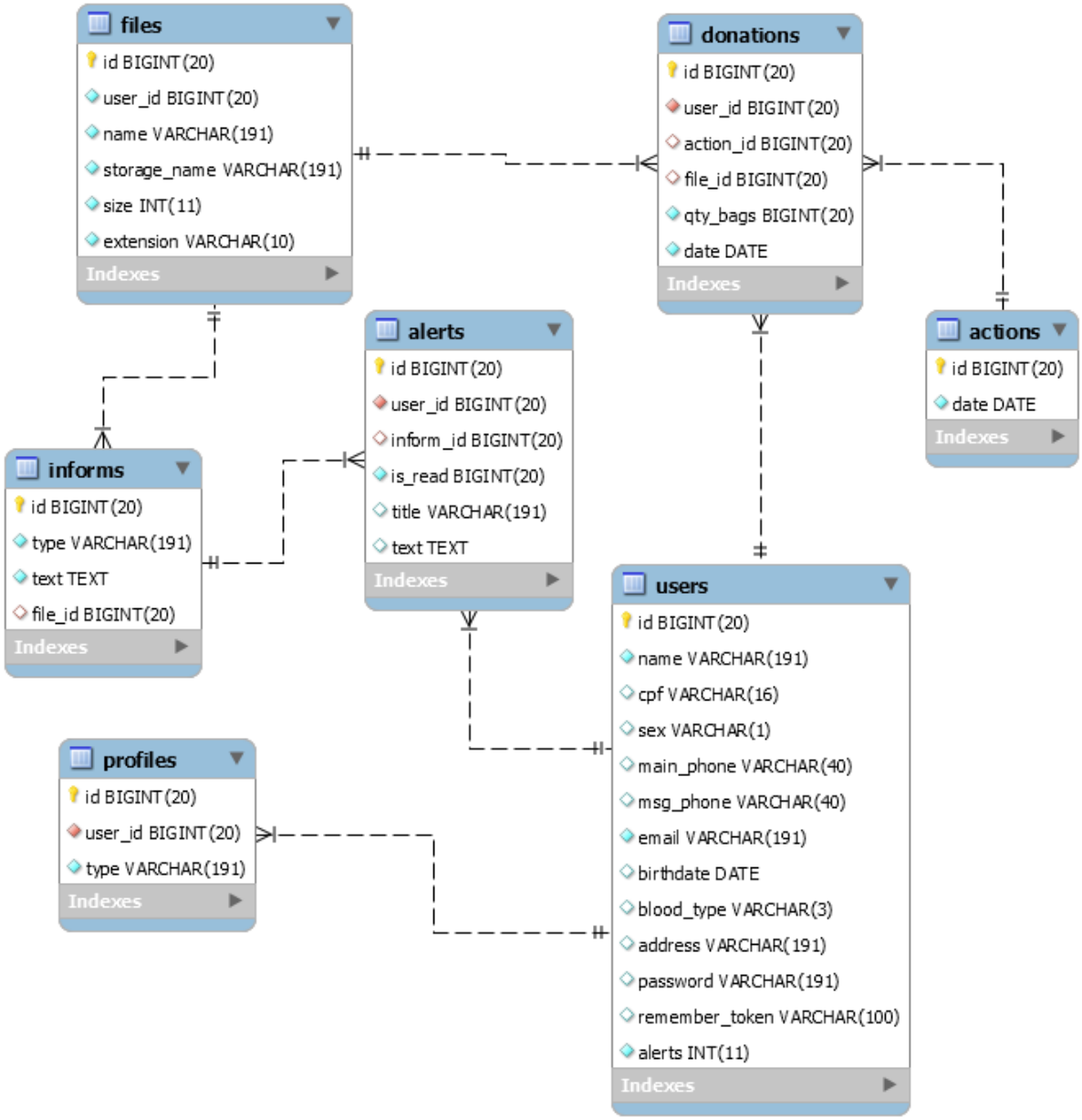
Fonte: Elaborado pelo autor.

4.6. Diagrama Entidade-Relacionamento Estendido (EER)

A Figura 4.2 apresenta o diagrama EER (entidade-relacionamento estendido) do Esperançar que, como o próprio nome diz, é uma extensão do modelo ER (entidade-relacionamento). De acordo com Coronel (2011), esse diagrama pode ser definido como o primeiro passo na jornada do desenvolvimento de um banco de dados, servindo como uma ponte entre os objetos do mundo real e o modelo de banco de dados que é implementado pelo computador. Os diagramas EER são modelos de alto nível, que representam os requisitos e complexidades dos bancos de dados e são úteis para sua modelagem por usarem conceitos relacionados à programação

orientada a objetos e apresentarem mais possibilidades de representação de propriedades e restrições do que os modelos ER.

Figura 4.2 - Diagrama entidade-relacionamento do Esperançar.



Fonte: Elaborado pelo autor.

Nos diagramas EER conseguimos representar os objetos com seus atributos. Por exemplo, na entidade ‘users’, ilustrada na figura, representamos os usuários com nomes, tipos sanguíneos e seu identificador único. A entidade ‘profiles’ é a responsável pelo perfil dos usuários (doador ou administrador) e por isso é relacionada com os usuários. Na entidade ‘donations’ temos informações sobre as doações dos usuários, como data de doação e quantidade de bolsas de sangue doadas. A entidade

‘donations’, além de se relacionar com usuários, por cada usuário ter várias doações ligadas a ela, também se relaciona com a entidade ‘actions’, que é a responsável pelas ações criadas pelos administradores, fazendo com que cada doação pertença a uma ação. Além disso, a entidade ‘donations’ também está relacionada com a entidade ‘files’, que é a responsável por armazenar arquivos, pois é ela quem armazena dados sobre a foto de cada doação. Existe ainda o relacionamento da entidade ‘files’ com a entidade ‘informs’, que guarda as informações dos informes criados pelos administradores do Esperançar, a saber: tipo, texto, além do arquivo. Por fim, a tabela ‘informs’, por disparar um alerta para cada usuário quando é criado, se relaciona com a entidade ‘alerts’, responsável pelos alertas que o usuário recebe.

5. IMPLEMENTAÇÃO

As requisições para a API, tanto na aplicação móvel quanto na aplicação *web*, ocorrem de forma similar, por meio dos métodos GET, POST, PUT e DELETE. Na aplicação móvel essas requisições são feitas utilizando a API *FETCH*, que fornece uma *interface* JavaScript para acessar e manipular partes do *pipeline* HTTP, tais como pedidos e respostas. Já na aplicação *web*, as requisições são feitas por meio do AXIOS, que é um cliente HTTP semelhante à API *FETCH* nativa do JavaScript. Neste capítulo damos como exemplo prático de uso das tecnologias discutidas, a funcionalidade de realizar chamadas das doações de sangue, tanto para a aplicação *web*, quanto para a aplicação móvel. Neste exemplo são desconsiderados alguns trechos de códigos que se relacionam com a *interface* ou manipulação dos dados, visando facilitar o entendimento.

5.1 Requisições da API via Aplicação *Web*

A Tabela 5.1 apresenta um trecho de código do *front-end* da aplicação *web*, que realiza as requisições de doações de sangue da API. Como filtro para a API, na Linha 3, passamos o parâmetro '*embed*', indicando que, para cada registro de doação do banco de dados, queremos também receber as informações de ação e doador, uma vez que existe um relacionamento entre esses conceitos: cada doação é parte de uma ação e possui um doador. Em seguida, realizamos a chamada do serviço de requisição de doação, que é responsável por requisitar as doações existentes, passando o filtro citado acima como parâmetro. Isso é feito por meio da chamada do serviço de doações *DonationRequestService* e do método *getAll* que, como pode ser observado na Tabela 5.2, realiza uma requisição GET, na Linha 3, para a API passando os parâmetros mencionados.

Tabela 5.1 – Código de chamada do método *getAll* do serviço *DonationRequestService* do *front-end* da aplicação *web* do Esperançar.

```
1  useEffect(() => {
2      let filters = {
3          embed: 'acao,doador'
4       };
5      let request = DonationRequestService.getAll({params:{...filters}});
6  }, []);
```

Fonte: Elaborado pelo autor.

Tabela 5.2 - Código do método *getAll* do serviço *DonationRequestService* do *front-end* da aplicação *web* do Esperançar.

```
1  const DonationRequestService = {
2      getAll(params) {
3          return api.get('donations', params);
4      }
5  };
```

Fonte: Elaborado pelo autor.

5.2 Requisições da API via Aplicação Móvel

O exemplo da solicitação realizada pelo aplicativo, para requisitar as doações do usuário logado, funciona de maneira similar à da aplicação *web*. Primeiramente, na Linha 2 da Tabela 5.3, verificamos se já temos a identificação do usuário (ID) logado. Caso não tenhamos, buscamos no *Storage* do aplicativo, local onde é armazenada toda informação do usuário quando ele faz *login*. Em seguida, utilizamos essa identificação na Linha 3, para poder requisitar as doações ao serviço *DonationService* por meio do método *getDonations*. Considerando a Tabela 5.4, na Linha 2 definimos a URL da nossa API e na Linha 3 buscamos o *token* de autenticação do usuário logado, para fazer *login* na API. Com isso, na Linha 4 realizamos a requisição GET, por meio do API *FETCH* do JavaScript, passando alguns parâmetros para a API. Um deles é o parâmetro ID, juntamente com os cabeçalhos da chamada (Linha 6), que servem para dizer para a API o que precisamos que seja retornado, além de passar a autorização de autenticação. Na Linha 12 usamos o método *json*, que retorna o corpo da resposta da nossa requisição como um arquivo JSON, usado como retorno do método na Linha 13.

Tabela 5.3 - Código do método *checkData* que realizada a chamada do método *getDonations* do serviço *DonationService* do módulo aplicativo Esperançar.

```
1  const checkData = async () => {
2      if (userId == '') setUserId(await AsyncStorage.getItem('id'));
3      let response = await DonationService.getDonations(userId);
4  };
```

Fonte: Elaborado pelo autor.

Tabela 5.4 - Código do método *getDonations* do serviço *DonationService* do módulo aplicativo Esperançar.

```
01 getDonations: async (user_id) => {
02     let url = BASE_API;
03     const token = await AsyncStorage.getItem('token');
04     const req = await fetch(`$(url)?embed=acao,arquivo&user_id=${user_id}`, {
05         method: 'GET',
06         headers: {
07             Accept: 'application/json',
08             'Content-Type': 'application/json',
09             'Authorization': 'Bearer ' + token,
10         }
11     });
12     const json = await req.json();
13     return json;
14 };
```

Fonte: Elaborado pelo autor.

5.3 Controlador

Todas as requisições feitas para a API são recebidas pelo controlador. Todos os controladores herdam da classe *ApiController* alguns métodos básicos, como *index*, *store*, *update* e *delete*. Requisições GET são manipuladas pelo método *index*, que realiza a chamada do componente de serviço. Este componente é responsável por construir a *query* de consulta, utilizando os parâmetros recebidos pela requisição e exibindo seu resultado formatado por meio do método *showAll*. Este método faz a organização dos dados recebidos do banco de dados e retorna como resposta um arquivo JSON, como ilustrado na Linha 2 da Tabela 5.5.

Tabela 5.5 - Código do método index do controlador da API.

```
1  public function index(){
2      return $this->showAll($this->service->index());
3  }
```

Fonte: Elaborado pelo autor.

5.4 Serviço

Assim como no controlador, todos os serviços herdam de uma classe os seus métodos. Neste caso, os serviços herdam os métodos da classe *BaseService*, que instancia os mesmos métodos básicos citados no controlador. Na Tabela 5.6, ao receber a chamada do método *index*, realizada pelo controlador, na Linha 2 o serviço executa o método *query*, que realiza a consulta a partir do objeto *model*, responsável por indicar qual tabela do banco de dados está sendo manipulada (neste exemplo é a tabela de doações) e atribuir os dados à variável *list*. A Linha 3 retorna os dados para o controlador.

Tabela 5.6 - Código do método *index* da camada de serviço.

1	`public function index(){`
2	`    $list = $this->model->query();`
3	`    return $list`
4	`}`

Fonte: Elaborado pelo autor.

6. DISCUSSÃO, CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho apresenta o ambiente Esperançar, que consiste em uma aplicação *web* e um aplicativo móvel para o gerenciamento e automatização das atividades relacionadas ao processo de doação de sangue, realizadas pela Associação Esperançar e pelos doadores de sangue.

A utilização das aplicações *web* e móvel ajudará tanto os doadores, que poderão informar suas doações, ter controle de doações efetuadas anteriormente e ficar por dentro das campanhas de doação, quanto os responsáveis pela Associação Esperançar, que poderão obter informações de todas as doações que forem recebidas e informar os doadores sobre ações em andamento.

Tanto a versão *web* como a versão para dispositivos móveis foram construídas com o propósito de serem simples e intuitivas para os usuários, facilitando suas atividades, previamente manuais, e permitindo que doadores e a associação se comuniquem de forma mais eficiente.

Para dar sequência a este trabalho, realizaremos testes no ambiente do cliente, com as aplicações já publicadas. Com isso, os usuários poderão utilizar o ambiente e as possíveis correções ou melhorias poderão ser identificadas. Além disso, o código desenvolvido para este trabalho é livre, de forma que a comunidade possa dar sequência e/ou sugerir melhorias. O código-fonte pode ser acessado em: https://bitbucket.org/marianemoreirasouza/esperancar/src/master/.

Como trabalhos futuros, novas funcionalidades poderão ser implementadas, facilitando ainda mais o trabalho dos funcionários da associação Esperançar, bem como auxiliando os doadores. Dentre essas funcionalidades, podemos citar a possibilidade de agendar doações pelo aplicativo, trazendo mais comodidade e organização, e a listagem de todos os locais de doação parceiros da Associação Esperançar, facilitando a realização de doações por parte do doador.

7. REFERÊNCIAS BIBLIOGRÁFICAS

Agência do Brasil, **Doação de sangue: 1,8% da população brasileira doa sangue; meta da OMS é 3%.** Disponível em: <https://agenciabrasil.ebc.com.br/geral/noticia/2017-06/doacao-de-sangue-18-da-populacao-brasileira-doa-sangue-meta-da-oms-e-3>. Acesso em 29 de agosto de 2021.

CHUNG, L., NIXON, B. A., YU, E. e MYLOPOULOS, J.; ***Non-Functional Requirements in Software Engineering***, 1 ed. Springer US, 2000.

CORONEL, Carlos; MORRIS, Steven; ROB, Peter; ***Database Systems: Design, Implementation, and Management***, 9 ed. Cengage Learning, 2011.

EISENMAN, Bonnie; ***Learning React Native: Building Native Mobile Apps with JavaScript***, 2 ed. O'Reilly Media, 2017.

GABARDO, Ademir; **Laravel para Ninjas**, 1 ed. Novatec, 2017.

GACKENHEIMER, Cory; ***Introduction to React***, 1 ed. Apress, 2015.

GOOGLE PLAY. **Doadores de Sangue**. Disponível em: <https://play.google.com/store/apps/details?id=br.DoadoresDeSangue>. Acesso em: 01 de março de 2021.

GOOGLE PLAY. **My Blood**. Disponível em: <https://play.google.com/store/apps/details?id=com.kleberandrade.myblood>. Acesso em: 01 de março de 2021.

GOVERNO DO BRASIL. **Hemovida.** Disponível em: <https://www.gov.br/pt-br/apps/hemovida>. Acesso em: 01 de março de 2021.

PARTIU DOAR SANGUE. Disponível em: <https://partiudoarsangue.com.br/#aplicativo>. Acesso em: 01 de março de 2021.

MASSE, Mark; ***REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces***, 1 ed. O'Reilly Media, 2011.

MILANI, André; **MySQL - Guia do Programador**, 1 ed. Novatec, 2007.

STAUFFER, Matt; ***Laravel: Up & Running: A Framework for Building Modern PHP Apps***, 2 ed. O'Reilly Media, 2019.

TAL, Liran; ***Agile Software Development with HP Agile Manager***, 1 ed. Apress, 2015.

VAZQUEZ, Carlos Eduardoo. e SIMÕES, Guilherme Siqueira; **Engenharia de Requisitos: software orientado ao negócio**, 1 ed. Brasport, 2016.

Anexo A - Requisitos Funcionais

A Tabela A.1 lista os RFs do Esperançar. A coluna “Interação” se refere ao código das interações descritas na Tabela 4.2.

Tabela A.1 - Requisitos funcionais do Esperançar.

Código	Interação	Requisito	Descrição
RF01	INT-01 e INT-02	Criação de conta	Para conectar-se ao aplicativo, o doador deve criar uma conta usando um e-mail e criando uma senha
RF02	INT-01	*Login* no aplicativo e aplicação *web*	Para se conectar ao aplicativo e à aplicação *web*, deve ser informado e-mail e senha
RF03	INT-01	*Logout* no aplicativo e aplicação *web*	Para se desconectar do aplicativo e da aplicação *web*, o doador ou o administrador devem clicar em um botão no menu.
RF04	INT-02	Visualização de usuários cadastrados	Deve ser possível que os administradores possam visualizar usuários cadastrados
RF05	INT-02	Inativar usuários	Deve ser possível que os administradores possam inativar usuários
RF06	INT-02	Criar usuário	Deve ser possível que os administradores possam criar usuários
RF07	INT-03	Criar ação	Deve ser possível que os administradores possam criar uma ação de doação
RF08	INT-03	Visualizar ações de doação	Deve ser possível que os administradores possam visualizar as ações existentes
RF09	INT-05	Visualizar doadores da ação	Deve ser possível que os administradores possam visualizar os doadores que fizeram doações para uma ação
RF10	INT-03	Convidar usuário para ação	Deve ser possível que os administradores possam enviar convite para doadores ao criarem uma ação
RF11	INT-04	Criar informativo	Deve ser possível que os administradores possam criar informativos
RF12	INT-04	Visualizar informativos	Deve ser possível que os administradores possam visualizar informativos
RF13	INT-11	Visualizar relatórios	Deve ser possível que os administradores possam visualizar e gerar relatórios
RF14	INT-07	Informar doação	Deve ser possível que os doadores possam informar que fizeram uma doação

RF15	INT-06	Visualizar doações	Deve ser possível que os doadores possam visualizar suas doações informadas
RF16	INT-06 e INT-07	Compartilhar doação	Deve ser possível que os doadores possam compartilhar suas doações realizadas
RF17	INT-12	Receber convite para fazer doação	Deve ser possível que os doadores possam receber convite dos administradores para fazerem doação
RF18	INT-08	Receber alerta de novo informativo	Deve ser possível que os doadores possam receber informativos disparados pelos administradores
RF19	INT-09	Verificar aptidão para doar	Deve ser possível que os doadores possam saber se estão aptos para doar sangue
RF20	INT-10	Enviar convite para usuário	Deve ser possível que os doadores possam convidar outros usuários para se tornarem doadores

Fonte: Elaborado pelo autor.

Anexo B - Prototipação

Utilizando a ferramenta Justinmind, foram criados protótipos interativos de telas do Esperançar, tanto do aplicativo móvel, quanto da aplicação *web*, com o objetivo de propor uma primeira experiência com o futuro ambiente para o cliente. Com isso, a ideia pôde ser validada antes de começar o desenvolvimento do ambiente, poupando esforços desnecessários. O fluxo de navegação foi criado, buscando tornar a utilização simples e objetiva para os atores, levando em consideração padrões utilizados em sistemas e aplicativos conhecidos do mercado.

Tanto o aplicativo para dispositivos móveis quanto a aplicação *web* iniciam na tela de *login* (Figuras B.1 e B.2). No caso do aplicativo existe, além do botão de fazer *login*, a opção de criação de conta para o doador (Figura B.3).

Figura B.1 – Tela de *login* da aplicação *web*

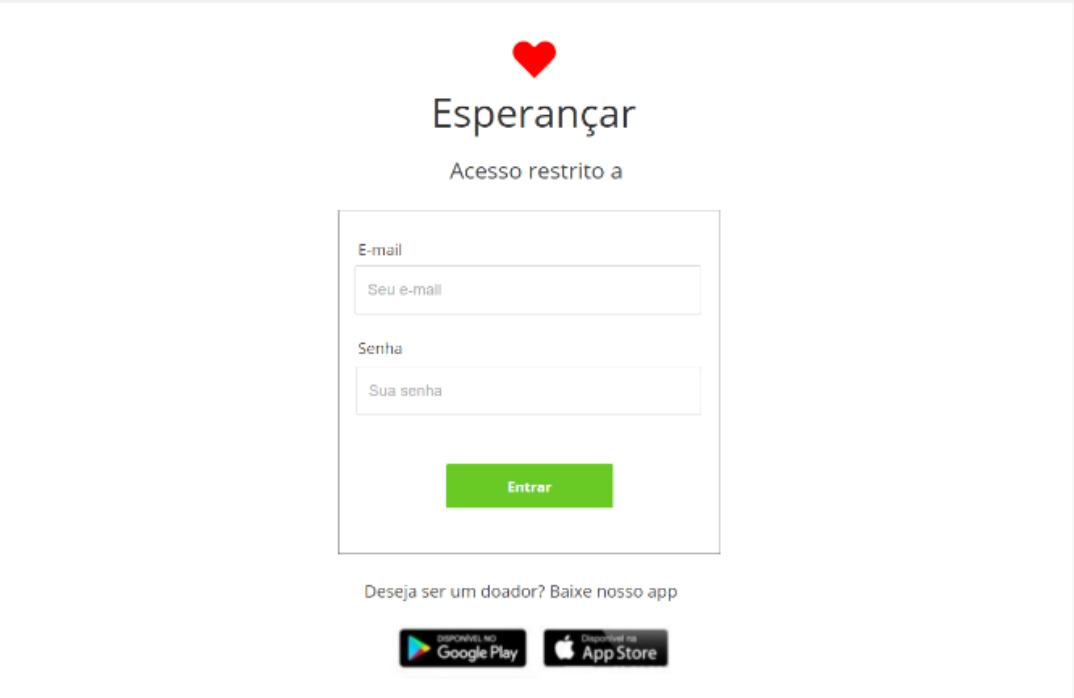


Figura B.2 - Tela de *login* do aplicativo móvel

Figura B.3 – Tela de criação de conta do aplicativo móvel

Após o *login*, o usuário é direcionado para a *dashboard* [4] (Figuras B.4 e B.5) que possui informações importantes e resumidas. Na *dashboard*, por meio do menu, o usuário consegue navegar para as demais telas

[4] A dashboard é um painel visual que apresenta, de maneira centralizada, um conjunto de informações: indicadores e suas métricas.

Figura B.4 – *Dashboard* aplicação *web*

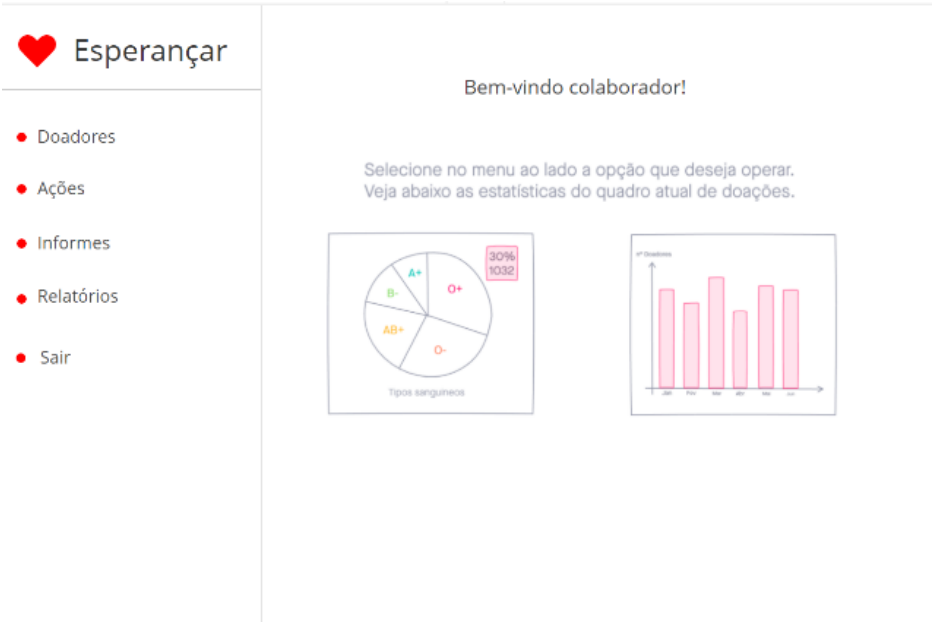


Figura B.5 - *Dashboard* do aplicativo móvel

Na aplicação *web*, o usuário pode acessar do menu lateral a tela de Doadores que, além de listar os doadores cadastrados (Figura B.6), permite também o cadastro de novos doadores. Na tela de Ações o usuário pode listar ações (Figura B.7), visualizar as ações já criadas (Figura B.8) ou criar uma ação de doação, enviando convite para os doadores (Figura B.9). Na tela de Informes, o usuário pode listar ou criar/visualizar informes (Figuras B.10 e B.11).

Por fim, a tela de relatórios (Figura B.12) permite que os usuários possam acessar relatórios dos dados de doações e doadores do Esperançar, como pode ser visto no relatório de exemplo da Figura B.13. O item ‘Sair’ do menu permite que o usuário faça *logout* e retorne para a tela de *login*.

Figura B.6 – Tela de doadores da aplicação *web* administrativa

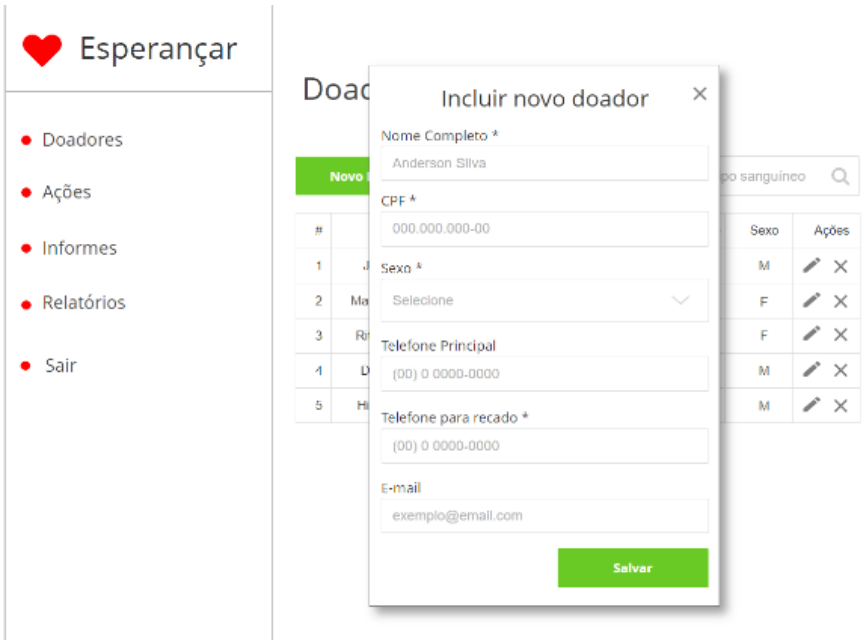


Figura B.7 - Tela de listagem de ações da aplicação *web* administrativa

Figura B.8 – Tela de criação de ação da aplicação *web* administrativa

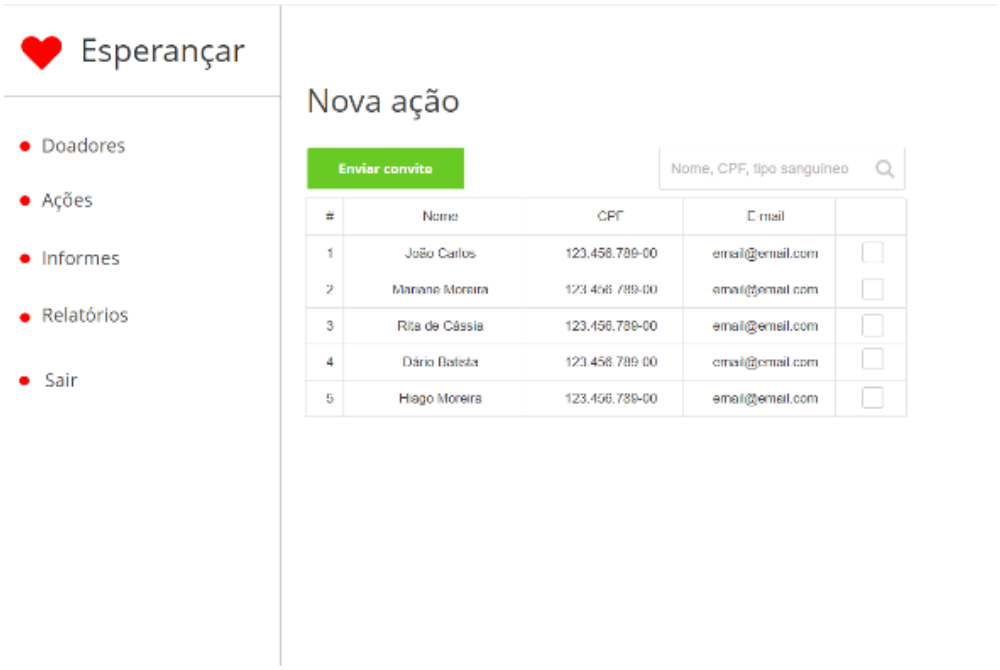


Figura B.9 – Tela de visualização de ação da aplicação *web* administrativa

Figura B.10 – Tela de listagem de informes da aplicação *web* administrativa

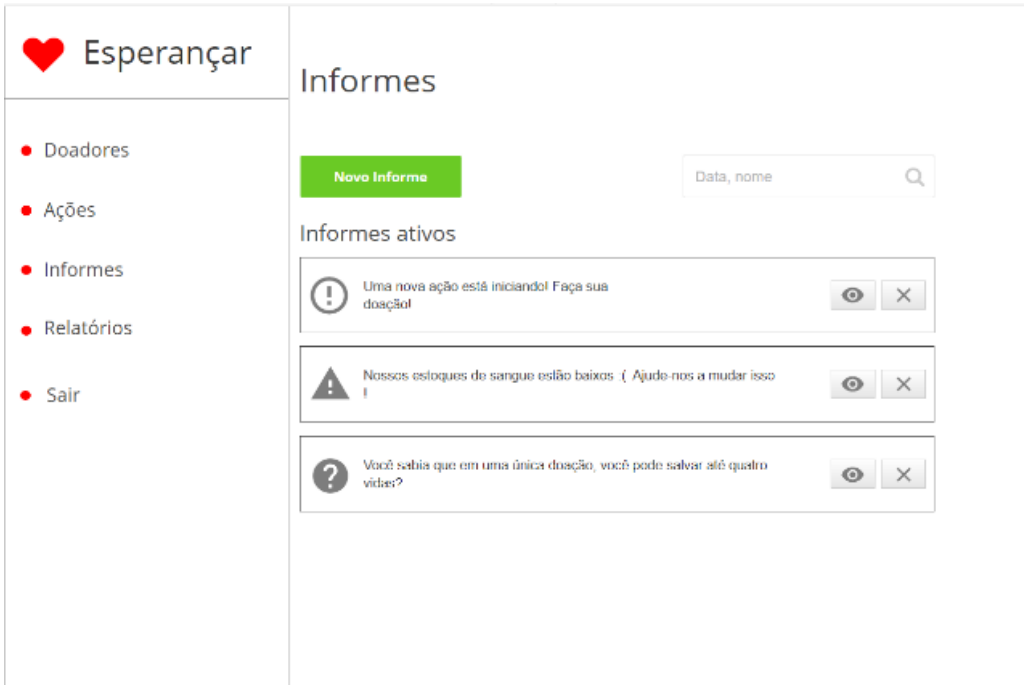


Figura B.11 – Tela de criação/visualização de informe da aplicação *web* administrativa

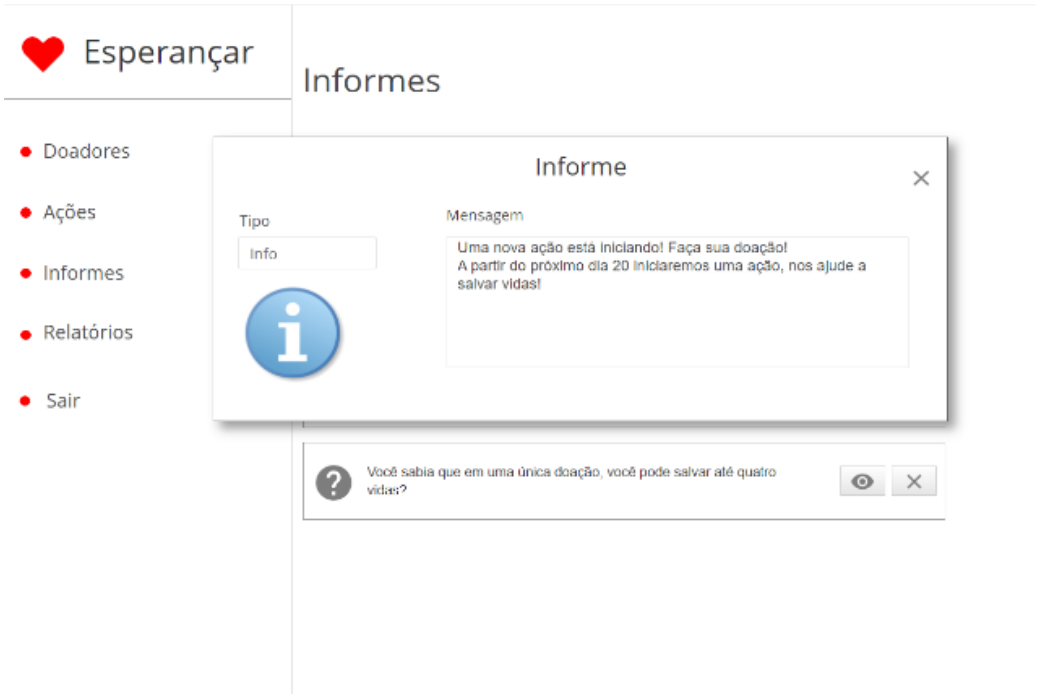


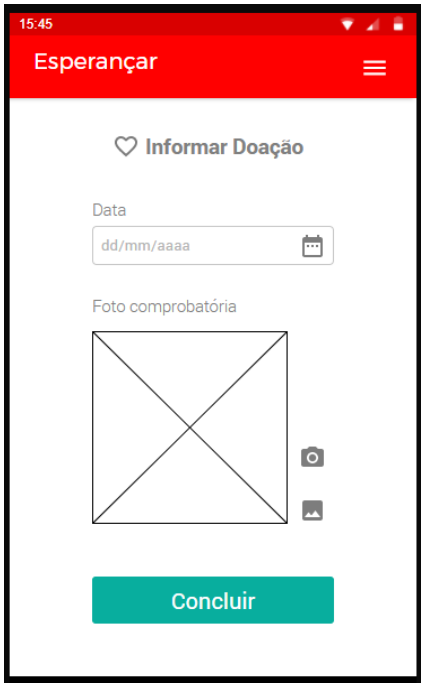
Figura B.12 - Tela de relatórios da aplicação *web* administrativa

Figura B.13 - Tela de relatório exemplo da aplicação *web* administrativa

No aplicativo para dispositivos móveis do Esperançar, o usuário, por meio do menu que pode ser acessado pelo ícone no canto superior, consegue navegar pelas outras telas do aplicativo, como pode ser visto na Figura B.14. Além disso, o usuário também pode acessar, a partir da dashboard, a tela de informar nova doação (Figura B.15).

Figura B.14 - Tela da *dashboard* do aplicativo para dispositivos móveis, com menu expandido

Figura B.15 - Tela de informar nova doação do aplicativo para dispositivos móveis



No menu doações, o usuário pode visualizar suas doações efetuadas anteriormente (Figura B.16), além de poder compartilhar a doação, como pode ser visto na Figura B.17.

Figura B.16 - Tela de doações do aplicativo para dispositivos móveis

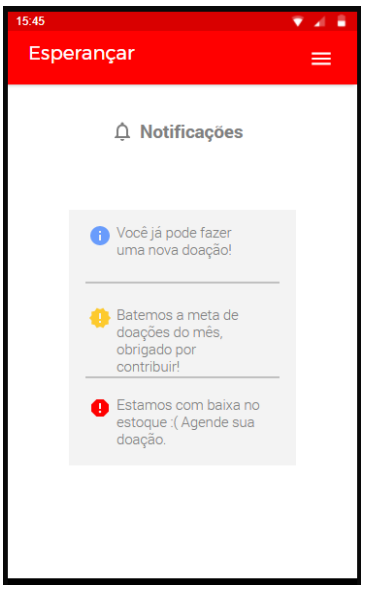
Figura B.17 - Tela de compartilhamento de doação do aplicativo para dispositivos móveis

Acessando o ‘Minha Conta’, por meio do menu, o usuário tem acesso à tela que exibe os detalhes do usuário logado (Figura B.18), na qual o usuário pode alterar alguns de seus dados.

Figura B.18 - Tela de informações do usuário no aplicativo para dispositivos móveis

Na Figura B.19 pode ser visto o protótipo de tela que é exibido acessando o item Notificações. Neste caso, o usuário pode visualizar suas notificações recebidas, que são enviadas pelos administradores, podendo ser um aviso de informe ou um convite para participar de uma nova ação.

Figura B.19 - Tela de notificações no aplicativo para dispositivos móveis



A Figura B.20 representa a tela de ajustes do aplicativo, em que o usuário pode escolher desativar o recebimento de notificações. O item de menu ‘Convidar amigo(a)’ permite que um *link* do aplicativo seja compartilhado com algum outro usuário (Figura B.21).

Figura B.20 - Tela de ajustes do aplicativo para dispositivos móveis

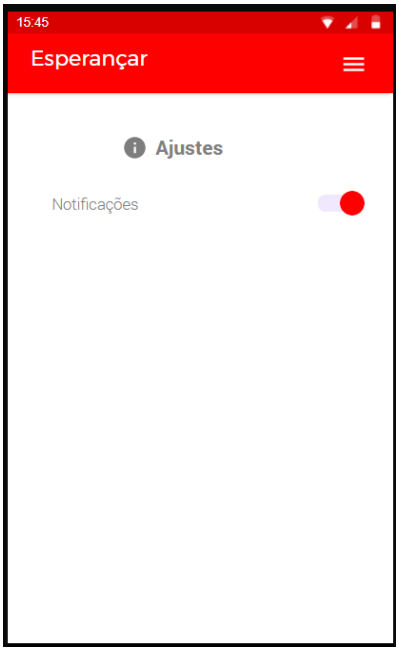
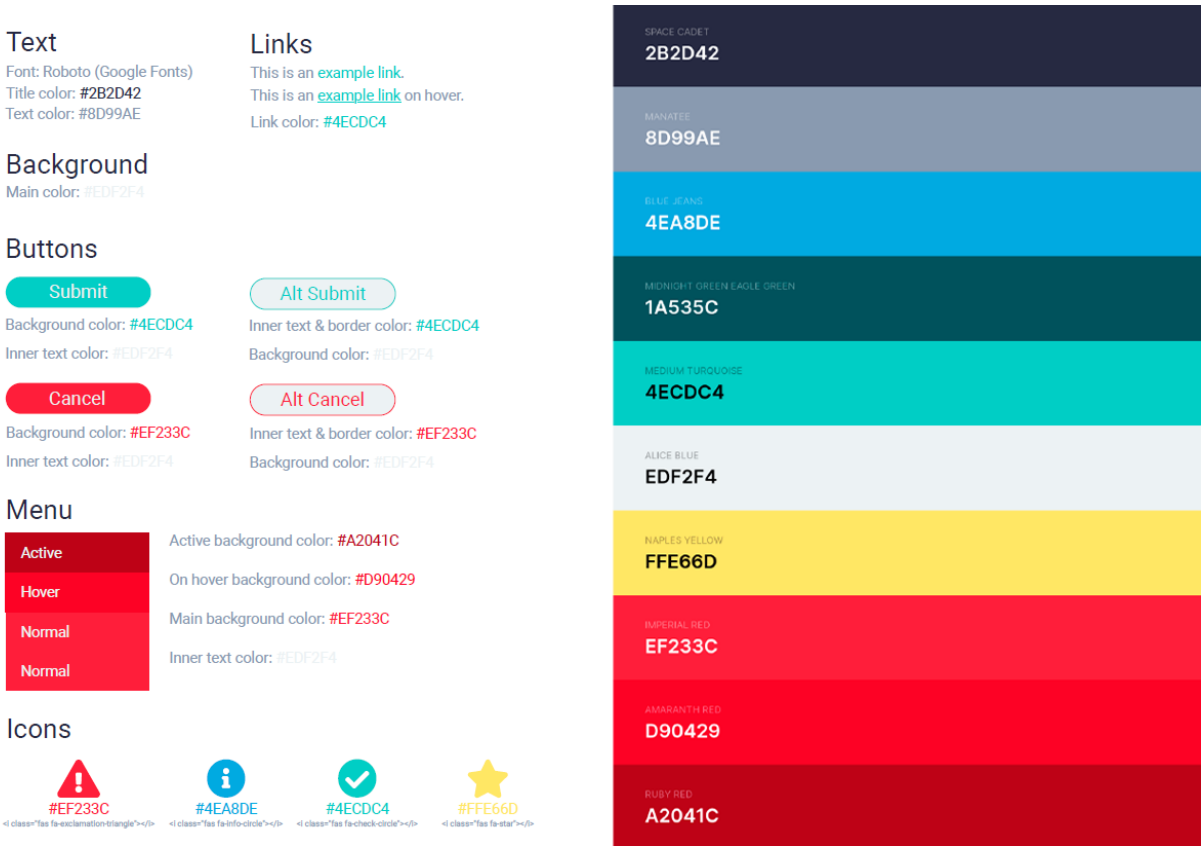


Figura B.21 - Tela de enviar convite do aplicativo para dispositivos móveis

Anexo C – *Style Guide*

O *style guide* é um guia para desenvolvimento de *interfaces* digitais. Sua função é padronizar elementos como tipografia, cores e botões durante o desenvolvimento de *interfaces* (Medium, 2017). A Figura B.22 exibe o *style guide* criado para o Esperançar utilizando-se a ferramenta Coolors [5].

Figura C.1 – *Style Guide* do Esperançar.



Fonte: Elaborado pelo autor

[5] Ferramenta de geração de paleta de cores. Disponível em: <https://coolors.co>.