
Faculdade de Tecnologia de Americana – Ministro Ralph Biasi
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

GUILBERT ADALSON DE OLIVEIRA
RICKI WARLEY VILLANUEVA MUJICA
ROMANTI-EZER GOMES DOS SANTOS
SILAS RODRIGUES GOUVEIA JUNIOR

**Desenvolvimento de um aplicativo para divulgar campanhas de doação de
sangue**

Faculdade de Tecnologia de Americana – Ministro Ralph Biasi
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

GUILBERT ADALSON DE OLIVEIRA
RICKI WARLEY VILLANUEVA MUJICA
ROMANTI-EZER GOMES DOS SANTOS
SILAS RODRIGUES GOUVEIA JUNIOR

**Desenvolvimento de um aplicativo para divulgar campanhas de doação de
sangue**

Projeto de Conclusão de Curso apresentado como requisito parcial para obtenção do grau Tecnólogo em Análise e Desenvolvimento de Sistemas, pela Faculdade de Tecnologia de Americana.

Orientador: Prof. Dr. Kleber de Oliveira Andrade

Área de concentração: Engenharia de Software

Americana

2019

FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS
Dados Internacionais de Catalogação-na-fonte

O47Dd OLIVEIRA, Guilbert Adalson de

Desenvolvimento de um aplicativo para divulgar campanhas de doação de sangue. / Guilbert Adalson de Oliveira, Ricki Warley Villanueva Mujica, Romanti-Ezer Gomes dos Santos, Silas Rodrigues Gouveia Junior. – Americana, 2019.

92f.

Monografia (Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza

Orientador: Prof. Dr. Kleber de Oliveira Andrade

1 Dispositivos móveis – aplicativos 2. Android – aplicativos 3. Inteligência artificial; I. MUJICA, Ricki Warley Villanueva II. SANTOS, Romanti-Ezer Gomes dos III. GOUVEIA JUNIOR, Silas Rodrigues IV. ANDRADE, Kleber de Oliveira. V. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana

CDU: 681.519

007.52

Faculdade de Tecnologia de Americana

Guilbert Adalson de Oliveira
Ricki Warley Villanueva Mujica
Romanti-ezer Gomes dos Santos
Silas Rodrigues Gouveia Junior

Desenvolvimento de um aplicativo para divulgar campanhas de doação de sangue

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas pelo Centro Paula Souza – FATEC Faculdade de Tecnologia de Americana. Área de concentração: Laboratório de Engenharia de Software.

Americana, 09 de dezembro de 2019.

Banca Examinadora:



Kleber de Oliveira Andrade
Doutor
Fatec Americana



Benedito Aparecido Cruz
Mestre
Fatec Americana



Eduardo Antonio Vicentini
Mestre
Fatec Americana

RESUMO

O número de doadores de sangue no Brasil está abaixo do percentual esperado pelo Ministério da Saúde, atualmente, o número de doadores de sangue corresponde a 1,6% da população o que está dentro dos parâmetros aceitáveis pela Organização Mundial de Saúde, mas ainda há espaço para alcançar a marca de 3% de doadores, o que seria ideal de acordo com estudos do Ministério da Saúde. Este trabalho tem como principal objetivo a criação de um aplicativo para aumentar o número de doadores no Brasil, tornando as doações de sangue mais simples tanto para quem precisa do sangue quanto para o doador, possibilitando que a própria comunidade publique campanhas de doação e que os doadores consigam filtrar estas campanhas por estado, cidade e tipo de sangue. Além disso, foi feito um *chatbot* preparado para responder dúvidas sobre doações de sangue de forma clara, eliminando o processo de pesquisa que o usuário faria. Para a criação do aplicativo foi aplicado a metodologia Scrum, realizando pequenas entregas ao longo de seis meses sendo divididos em sete *sprints*. Foram utilizadas as mais recentes tecnologias para o desenvolvimento como o Ionic, Firebase, Node JS, Dialogflow, entre outros.

Palavras-chave: Doações; Aplicativo; *Chatbot*; Ministério da Saúde; Organização Mundial de Saúde; Sangue; Ionic; Firebase; Dialogflow.

ABSTRACT

The number of blood donors in Brazil is below the percentage expected by the Ministry of Health, currently, the number of blood donors corresponds to 1.6% of the population which is within the acceptable parameters by the World Health Organization but there is still room to reach the 3% donors mark, which would be ideal according to studies by the Ministry of Health. The main objective of this work is to create an application to increase the number of donors in Brazil, making blood donations simpler for both those who need blood and for the blood donor, enabling the community to publish donation campaigns and blood donors to filter these campaigns by state, city and blood type. In addition, we provide a chatbot prepared to answer questions about blood donations in a clear way, eliminating the research process that the user would do. For the creation of the application was applied the Scrum methodology, making small deliveries over six months and divided into seven sprints. The latest development technologies were used such as Ionic, Firebase, Node JS, Dialogflow, among others.

Key words: Donations; Application; Chatbot; Ministry of Health; World Health Organization; Blood; Ionic; Firebase; Dialogflow.

AGRADECIMENTOS

Nossos agradecimentos a todos que contribuíram, direta ou indiretamente para a realização deste trabalho, em especial:

Ao nosso coordenador Kleber de Oliveira Andrade que nos auxiliou desde o início, ao professor Benedito Antunes Luciano de França pelos ensinamentos de Metodologia Científica e aos demais professores do Curso de Análise e desenvolvimento de Sistemas, pela dedicação, competência e por todos os ensinamentos.

Aos nossos amigos e familiares, por sempre confiarem, nos incentivando e apoiando durante esse período de elaboração do trabalho.

Por fim, gostaríamos de agradecer a Deus, que vem nos guiando e amparando durante todos os momentos difíceis de nossas vidas.

SUMÁRIO

1.	INTRODUÇÃO	14
2.	PROJETO DO MYBLOOD	17
2.1	Pesquisa de campo	17
2.2	Levantamento de Requisitos	19
2.2.1	Requisitos Funcionais	19
2.2.2	Requisitos Não Funcionais.....	20
2.3	Recursos e Ferramentas	21
3.	MODELAGEM.....	25
3.1	Diagramas de Caso de Uso.....	25
3.1.1	Documentação dos casos de uso	26
3.1.2	Casos de teste	30
3.2	Diagrama de Classes.....	34
3.3	Banco de Dados	36
3.3.1	Diagrama de Entidade e Relacionamento (DER)	36
3.3.2	Dicionário de Dados.....	37
4.	DESENVOLVIMENTO	40
4.1	Arquitetura e organização.....	40
4.2	API.....	42
4.2.1	API do Firebase Authentication	42
4.2.2	API do Cloud Firestore	44
4.2.3	API do ViaCEP	47
4.2.4	Maps JavaScript API e Geocoding API	48
4.2.5	API do Dialogflow	49
4.3	Metodologia Ágil.....	51
4.3.1	Metodologia ágil Scrum.....	51
4.4	Etapas de desenvolvimento.....	52

4.4.1	Primeira entrega.....	53
4.4.2	Segunda entrega.....	55
4.4.3	Terceira entrega.....	57
4.4.4	Quarta entrega	58
4.4.5	Quinta entrega	60
4.4.6	Sexta entrega.....	61
4.4.7	Sétima entrega.....	63
4.5	Interfaces de Usuário.....	64
5.	CONSIDERAÇÕES FINAIS	84
	REFERÊNCIAS.....	86

LISTA DE FIGURAS

Figura 1 - Entrada do Hospital da Região Metropolitana de Campinas.....	18
Figura 2 - Entrada do banco de sangue	18
Figura 3 - Sala de espera para voluntários	19
Figura 4 - Diagrama de caso de uso do aplicativo.	26
Figura 5 - Diagrama de classes do aplicativo.....	34
Figura 6 - DER do banco de dados armazenado no Cloud Firestore.....	36
Figura 7 - API do Firebase Authentication.....	43
Figura 8 - API do Cloud Firestore.....	44
Figura 9 - API do Cloud Firestore parte 2.....	45
Figura 10 - API do Cloud Firestore parte 3.....	46
Figura 11 - API do ViaCEP.....	47
Figura 12 - API do ViaCEP exemplo	47
Figura 13 - Maps JavaScript API e Geocoding API.....	48
Figura 14 - Maps JavaScript API e Geocoding API parte 2.....	49
Figura 15 - API do Dialogflow.....	49
Figura 16 - API do Dialogflow parte 2.....	50
Figura 17 - Processo de desenvolvimento utilizando Scrum.....	51
Figura 18 - Gráfico de Burndown da primeira etapa de desenvolvimento.....	54
Figura 19 - Gráfico de Burndown da segunda etapa de desenvolvimento.	56
Figura 20 - Gráfico de Burndown da terceira etapa de desenvolvimento.....	58
Figura 21 - Gráfico de Burndown da quarta etapa de desenvolvimento.....	59
Figura 22 - Gráfico de Burndown da quinta etapa de desenvolvimento.	61
Figura 23 - Gráfico de Burndown da sexta etapa de desenvolvimento	62
Figura 24 - Gráfico de Burndown da sétima etapa de desenvolvimento	64
Figura 25 - Tela de splash screen	65
Figura 26 - Tela de cadastro	66
Figura 27 - Tela de cadastro com botões do alerta	66
Figura 28 - Tela de login	67
Figura 29 - Esqueci minha senha.....	68
Figura 30 - Menu lateral	69
Figura 31 - Tela de início.....	70
Figura 32 - Publicar campanha exibição dos dados da doação	71

Figura 33 - Publicar campanha exibição do campo de data.....	71
Figura 34 - Publicar campanha exibição dos dados do hemocentro	72
Figura 35 - Publicar campanha exibição dos dados de contato da campanha	72
Figura 36 - Publicar campanha exibição da validação de campos.....	73
Figura 37 - Visualizar campanhas tipos de filtro.....	74
Figura 38 - Visualizar campanhas filtro por UF	74
Figura 39 - Visualizar campanhas filtro	75
Figura 40 - Visualizar campanhas exibição do mapa	76
Figura 41 - Visualizar campanhas exibição do mapa com os dados da campanha ..	76
Figura 42 - Visualizar campanhas exibição da foto com a localidade do hemocentro publicado.....	77
Figura 43 - Visualizar campanhas exibição da foto com a localidade do hemocentro publicado e dados da campanha.....	77
Figura 44 - Visualizar campanhas visualização do mapa expandido	78
Figura 45 - Minha conta dados pessoais do usuário	79
Figura 46 - Minha conta campo de gênero selecionado.....	79
Figura 47 - Minha conta tipo sanguíneo do usuário	80
Figura 48 - Minhas doações	80
Figura 49 - Minhas doações cadastradas	81
Figura 50 - Guia informativo tela de boas vindas	82
Figura 51 - Guia informativo exemplo de pergunta	82
Figura 52 - Cadastro visualização na versão web.....	83
Figura 53 - Login visualização na versão web.....	83

LISTA DE TABELAS

Tabela 1 - Comparativo de funcionalidades da aplicação MyBlood em relação aos concorrentes.....	15
Tabela 2 - Requisitos funcionais do projeto.	20
Tabela 3 - Requisitos não funcionais do projeto.....	20
Tabela 4 - Caso de uso “Publicar campanha”	26
Tabela 5 - Caso de uso “Visualizar campanha”.....	27
Tabela 6 - Caso de uso “Minhas Doações”	27
Tabela 7 - Caso de uso “Guia informativo”.....	28
Tabela 8 - Caso de uso “Minha Conta”	29
Tabela 9 - Caso de uso “Efetuar Cadastro”.....	29
Tabela 10 - Caso de uso “Entrar no Aplicativo”.....	30
Tabela 11 - Caso de Teste do login	31
Tabela 12 - Caso de Teste do cadastro	31
Tabela 13 - Caso de Teste do publicar campanha.....	32
Tabela 14 - Caso de Teste do visualizar campanha	32
Tabela 15 - Caso de Teste do guia informativo.....	32
Tabela 16 - Caso de Teste do minhas doações	33
Tabela 17 - Caso de Teste da minha conta	33
Tabela 18 - Requisitos não funcionais do projeto.....	34
Tabela 19 - Requisitos não funcionais do projeto.....	35
Tabela 20 - Requisitos não funcionais do projeto.....	35
Tabela 21 - Entidade Divulgador	37
Tabela 22 - Entidade Campanha.....	38
Tabela 23 - Entidade Doador	38
Tabela 24 - Entidade Hemocentro.....	38
Tabela 25 - Entidade Usuário.....	39
Tabela 26 - Primeira etapa de desenvolvimento	53
Tabela 27 - Segunda etapa de desenvolvimento	55
Tabela 28 - Terceira etapa de desenvolvimento	57
Tabela 29 - Quarta etapa de desenvolvimento.....	59
Tabela 30 - Quinta etapa de desenvolvimento	60
Tabela 31 - Sexta etapa de desenvolvimento	62

Tabela 32 - Sétima etapa de desenvolvimento	63
---	----

LISTA DE SIGLAS E ABREVIATURAS

AI	Artificial Intelligence
API	Application Programming Interface
APK	Android Application Pack
BAAS	Back-end as a Service
CPF	Cadastro de pessoa física
CQ	Controle da Qualidade
CSS	Cascading Style Sheets
DER	Diagrama de Entidade e Relacionamento
DOM	Document Object Model
HTML	HyperText Markup Language
IBGE	Instituto Brasileiro de Geografia e Estatística
ID	Identifier
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
MIB	MebiByte
NPM	Node Package Manager
OMS	Organização Mundial de Saúde
RE	Requirements Engineering
SASS	Syntactically Awesome Stylesheets
SSL	Secure Socket Layer
TI	Tecnologia da informação
UF	Unidade da Federação
UI	User Interface
UML	Unified Modeling Language
WEB	Redução de World Wide Web

1. INTRODUÇÃO

No Brasil o número de doadores de sangue está de dezesseis a cada mil habitantes, isso equivale a 1,6% da população brasileira apesar de esse número estar dentro dos parâmetros aceitáveis da OMS (Organização Mundial de Saúde), a OMS recomenda chegar aos 3% dos habitantes que seria considerado o ideal. Os maiores desafios para o crescimento dessa porcentagem estão relacionados com a falta de conscientização, estigmas (mitos), normas e proibições, deficiência estrutural e herança cultural (cultura do país).

Tendo em vista os problemas apresentados, foi pensado no desenvolvimento de um aplicativo comunitário com a finalidade de aumentar o número de doadores no Brasil, conscientizar a comunidade sobre os benefícios de uma doação de sangue e apresentar os pré-requisitos e os cuidados pós-doação. A solução se demonstrou viável para desenvolvimento e com grandes potenciais de impacto porque será publicada em lojas de aplicativos e pela internet, meios acessíveis, pois, como demonstra o IBGE (Instituto Brasileiro de Geografia e Estatística), 69% dos brasileiros tem acesso à internet pelo celular e 92,7% dos lares contam com pelo menos uma pessoa dona de uma linha de telefonia móvel. Considerando esses dados, campanhas poderão ser publicadas no aplicativo e promovidas pela própria comunidade com o propósito de divulgar hemocentros que necessitam de doações e, conseqüentemente aumentar o número de doadores regulares.

Atualmente existem vários aplicativos voltados para doação de sangue cada um com diferentes funcionalidades, foram selecionadas às três aplicações mais populares e mais bem avaliadas pelos usuários da Play Store, são eles:

- **Hemoliga:** aplicação que contém informações sobre os hemocentros próximos e o nível de sangue destes. O aplicativo tem o objetivo de promover o engajamento social por meio da participação e colaboração de cada indivíduo visando aumentar o número de doadores ativos e da quantidade de doações e conseqüentemente o de vidas salvas. Aplicação com nota 4,1 na Google Play (GOOGLE PLAY, 2019a).
- **Partiu doar sangue:** aplicativo com o objetivo de ajudar a aumentar o número de doadores de sangue. Algumas das funcionalidades são: toda vez que uma nova solicitação de doação é feita os doadores são notificados, os usuários

podem indicar que vão atender os pedidos de doações, e, caso eles efetuem as doações, quem solicitou a doação é notificado, o que possibilita um maior controle do número de doações conseguidas. Aplicação com nota 4,6 na Google Play (GOOGLE PLAY, 2019b).

- **Time do sangue:** aplicação como o foco voltado para o agendamento de doações e exames, possui uma grande base de hemocentros como diferencial. O aplicativo auxilia a encontrar os hemocentros mais próximo do usuário, os hemocentros com maior falta do seu tipo sanguíneo, realizar agendamento para doação, e organizar todos os exames e doações realizadas pelo usuário. Aplicação com nota 4,0 na Google Play (GOOGLE PLAY, 2019c).

Levando estes aspectos em consideração, foi elaborada a Tabela 1 mostrando as principais diferenças do MyBlood, em relação aos aplicativos citados anteriormente:

Tabela 1 - Comparativo de funcionalidades da aplicação MyBlood em relação aos concorrentes.

Funcionalidades	Partiu doar sangue	Time do sangue	Hemoliga	MyBlood
Gerenciar entradas de hospital/banco de sangue	X	X	X	X
Cadastro de hemocentro pela comunidade	-	-	-	X
Gerenciar entradas de doações de sangue	X	X	X	X
Gerenciar entradas de doações de plaqueta	X	-	-	X
Gerenciar campanhas	X	-	X	X
Publicação de campanhas pela comunidade	-	-	-	X
Gerar relatório de doações	-	-	-	X
Gerenciar dados do usuário	X	X	X	X
Informar doador sobre restrições de doação	X	-	X	X
Notificar para doar, após um período da última doação	-	-	-	X
Compartilhar solicitações nas redes sociais	-	-	-	X
Ser multiplataforma	X	-	-	X

Fonte: Elaborado pelo autor (2019).

Quanto aos objetivos específicos deste trabalho, são eles:

- Possibilitar que a própria comunidade movimente o aplicativo e ajude diretamente no crescimento do número de doadores.
- Facilitar a busca de hemocentros próximos com o sangue compatível do doador.
- Conscientizar a comunidade sobre os benefícios de uma doação de sangue.
- Facilitar o controle das doações de sangue e lembrar o doador quando ele pode doar novamente.

As demais partes do trabalho estão organizadas em 5 sessões conforme descrição: Capítulo 2 apresenta a pesquisa de campo realizada no hemocentro de Americana, o levantamento de requisitos, os requisitos funcionais e não funcionais da aplicação e as ferramentas utilizadas, o Capítulo 3 mostra os diagramas tanto do aplicativo como a do banco de dados, o Capítulo 4 descreve a arquitetura e a organização do código da aplicação, são mencionadas e explicadas as APIs usadas no projeto, o conceito de metodologia ágil e as etapas de entrega descrevendo o desenvolvimento do projeto por meio das *sprints* utilizando a metodologia Scrum e também é apresentada a interface do aplicativo, e por fim, as considerações finais com as diversas possibilidades de trabalhos futuros são apresentadas no Capítulo 5.

2. PROJETO DO MYBLOOD

O aplicativo foi desenvolvido utilizando o *framework* Ionic que possibilita a criação de aplicativos híbridos, ou seja, aplicativos compatíveis com Android, IOS e Windows Phone. Foram realizadas pesquisas de opinião da comunidade de usuários de outras plataformas com boas avaliações na Play Store com a finalidade de usar o comentário desses aplicativos para agregar ao aplicativo MyBlood.

Sabendo o que é solicitado em melhorias pelos usuários desses aplicativos já existentes no mercado, foi possível concentrar o tempo para desenvolver com prioridade aquilo que é esperado pelo usuário. Durante a avaliação de aplicativos similares, percebeu-se uma funcionalidade diferenciada para o MyBlood, o *chatbot* introduzido no guia informativo, que busca esclarecer as dúvidas mais frequentes do usuário, enquanto aperfeiçoa sua capacidade de responder diversas perguntas a respeito de doação de sangue, na medida que surgem perguntas por ele não compreendidas. Essa tela é demonstrada na figura 38.

2.1 Pesquisa de campo

Para construção de conhecimento sobre doação de sangue, realizou-se uma pesquisa de campo. Segundo Gonsalves (2001, p.67)

“A pesquisa de campo é o tipo de pesquisa que pretende buscar a informação diretamente com a população pesquisada. Ela exige do pesquisador um encontro mais direto. Nesse caso, o pesquisador precisa ir ao espaço onde o fenômeno ocorre, ou ocorreu e reunir um conjunto de informações a serem documentadas [...]. “

Conforme GIL (2002), Pesquisas exploratórias têm como finalidade permitir maior domínio com o problema estudado.

A pesquisa de campo escolhida foi a exploratória, por ela ser a mais adequada para trabalhar quando se possui menor conhecimento no assunto abordado. Dessa forma, foi utilizada para alavancar o domínio sobre o tema estudado, tornando-o mais claro.

Realizada em um Hospital na Região Metropolitana de Campinas, foram efetuadas doações de sangue e uma reunião com a enfermeira chefe para tirar dúvidas. Durante a visita também foi recolhido informações com os enfermeiros e com os doadores voluntários presentes. As figuras 1, 2 e 3 mostram o ambiente onde ocorreu o estudo.

Figura 1 - Entrada do Hospital da Região Metropolitana de Campinas.



Fonte: Elaborado pelo autor (2019).

Figura 2 - Entrada do banco de sangue



Fonte: Elaborado pelo autor (2019).

Figura 3 - Sala de espera para voluntários



Fonte: Elaborado pelo autor (2019).

Com essa visita, informações sobre o documento do Ministério da Saúde referente à doação de sangue (PORTARIA Nº 158, DE 4 DE FEVEREIRO DE 2016) foram descobertas. Esse documento favoreceu o aplicativo agregando informações úteis e foi estudado pelo grupo para melhoria das funcionalidades existentes.

2.2 Levantamento de Requisitos

É o processo de compreensão e definição dos serviços requisitados do sistema e identificação de restrições relativas à operação e ao desenvolvimento do sistema. A engenharia de requisitos é um estágio particularmente crítico do processo de software, pois erros nessa fase inevitavelmente geram problemas no projeto e na implementação do sistema (SOMMERVILLE, 2011 p.24).

Tradicionalmente, os requisitos são divididos em dois tipos: requisitos funcionais e requisitos não funcionais.

2.2.1 Requisitos Funcionais

São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais

também podem explicitar o que o sistema não deve fazer (SOMMERVILLE, 2011 p.59).

A Tabela 2 apresenta os requisitos funcionais deste projeto.

Tabela 2 - Requisitos funcionais do projeto.

Identificação	Requisito Funcional	Prioridade
RF001	Cadastrar hospital/banco de sangue.	Essencial
RF002	Visualizar campanhas publicadas.	Importante
RF003	Minhas doações.	Importante
RF004	Publicar campanha.	Essencial
RF005	Autenticação do usuário com criptografia.	Essencial
RF006	Cadastro do usuário.	Essencial
RF007	Avisar doador sobre restrições de doação.	Importante
RF008	Atualizar dados de cadastro.	Importante
RF009	Guia Informativa com <i>chatbot</i> .	Desejável

Fonte: Elaborado pelo autor (2019).

2.2.2 Requisitos Não Funcionais

São restrições aos serviços ou funções oferecidas pelo sistema. Incluem restrições de timing, restrições no processo de desenvolvimento e restrições impostas pelas normas. Ao contrário das características individuais ou serviços do sistema, os requisitos não funcionais, muitas vezes, aplicam-se ao sistema como um todo (SOMMERVILLE, 2011 p.59).

A Tabela 3 apresenta os requisitos não funcionais deste projeto.

Tabela 3 - Requisitos não funcionais do projeto.

Identificação	Requisito não funcional	Categoria	Prioridade
RNF001	Experiência do usuário.	Usabilidade	Essencial
RNF002	Ser intuitivo o uso do aplicativo.	Usabilidade	Essencial
RNF003	Interface limpa.	Usabilidade	Essencial
RNF004	Segurança e privacidade.	Segurança	Essencial
RNF005	Autenticação do usuário com criptografia.	Segurança	Essencial
RNF006	Disponibilização de todos os dados coletados do usuário.	Confiabilidade	Essencial

RNF007	Suportar versão do Android superior ao 4.4.	Hardware e Software	Essencial
RNF008	Suporta IOS superior ao 10.	Hardware e Software	Essencial
RNF009	Possibilitar uso parcial quando <i>offline</i> .	Hardware e Software	Desejável

Fonte: Elaborado pelo autor (2019).

2.3 Recursos e Ferramentas

Esta seção contempla as ferramentas de programação e os conceitos necessários para o desenvolvimento do sistema:

- **Adobe Photoshop:** é um aplicativo de software para edição de imagens e retoque de fotos para uso em computadores Windows ou MacOS. O Photoshop oferece aos usuários a capacidade de criar, aprimorar ou editar imagens e ilustrações (ADOBE, 2019).
- **Adobe XD:** o Adobe Experience Design ou Adobe XD, integra design e prototipagem em uma aplicação desktop que é rápida e fluida. A solução ajuda designers a alternarem entre *wireframe*, design, protótipo e preview, para conseguirem desenvolver apps e sites que mantêm os usuários engajados (ADOBE, 2019).
- **Android Studio:** ambiente de desenvolvimento integrado (IDE) oficial para o desenvolvimento de aplicativos Android, baseado no IntelliJ IDEA, que oferece um ambiente unificado para o desenvolvimento de aplicativos. No qual é possível desenvolver, fazer *debugs*, testes e interfaces para smartphones e tablets Android e dispositivos Android Wear, Android TV e Android Auto (ANDROID STUDIO, 2019).
- **Apache Cordova:** é uma plataforma de desenvolvimento móvel com APIs (*Application Programming Interface*) que permitem que o desenvolvedor acesse funções nativas do dispositivo, como a câmera ou o acelerômetro (APACHE CORDOVA, 2019).
- **Astah UML:** é uma ferramenta que suporta os requerimentos da UML 2.x para construção de diagramas de classe, caso de uso, sequência, máquina de estado, atividade, componente, dentre outros (ASTAH, 2019). É a ferramenta utilizada para a construção dos diagramas de classe e caso de uso.

- **Br Modelo:** é uma ferramenta criada para o auxílio na criação de modelos relacionais de um banco de dados (BERNARDI, 2019).
- **Codacy:** ferramenta que identifica automaticamente problemas através da análise estática e revisão do código em todos os *commits* (conjunto de alterações enviadas para o versionamento de código) realizados durante o fluxo de desenvolvimento. Quando há alguma irregularidade, é enviado automaticamente uma notificação avisando a equipe. (CODACY, 2019).
- **Firebase:** é uma plataforma do Google que contém várias ferramentas e uma excelente infraestrutura para ajudar desenvolvedores web e mobile a criar aplicações de alta qualidade e performance, a plataforma possui vários serviços (GOOGLE DEVELOPERS, 2019), as utilizadas no desenvolvimento foram:
 - **Autenticação:** é possível gerenciar de forma simples e segura os usuários, o Firebase oferece vários métodos de autenticação como e-mail e senha, conta da Google e Facebook.
 - **Cloud Firestore:** é possível armazenar e sincronizar dados entre o usuário e o dispositivo em escala global através de um banco de dados NoSQL que está hospedado na nuvem.
 - **Dialogflow:** desenvolvido pela Google, é uma plataforma que utiliza de tecnologias de interação humano-computador baseadas em conversas de *natural language*¹.
- **Git:** é um sistema de controle de versão distribuído de código aberto e gratuito, projetado para lidar com tudo, desde projetos pequenos a grandes, com velocidade e eficiência (GIT, 2019).
- **GitGuardian:** é uma ferramenta de segurança destinada a proteger os desenvolvedores de liberar segredos potencialmente prejudiciais no GitHub. O serviço faz uma varredura pelo repositório em busca de chaves de API, conexões com bancos de dados, certificados SSL e outros tipos de acessos (GITGUARDIAN, 2019).
- **GitHub:** é uma plataforma de desenvolvimento inspirada na maneira como você trabalha. Do código aberto aos negócios, você pode hospedar e revisar

¹ *Natural language* ou Linguagem natural é um ramo da inteligência artificial que ajuda os computadores a entender, interpretar e manipular a linguagem humana.

códigos, gerenciar projetos e criar software junto a 40 milhões de desenvolvedores (GITHUB, 2019).

- **HTML:** é uma das linguagens mais utilizadas pelos desenvolvedores de *websites*. O acrônimo HTML vem do inglês e significa *Hypertext Markup Language* ou em português Linguagem de Marcação de Hipertexto. As marcações são feitas através de *tags* que exibem o conteúdo do site.

As *tags* tem o propósito de dizer ao navegador o que é cada informação, assim ele consegue diferenciar o que é título, parágrafo, botão entre outros. (MARQUES, 2019).

- **Ionic:** *framework* para a criação de aplicativos móveis e de desktop usando a tecnologia Web (HTML, CSS e JavaScript), tem como principal objetivo criar aplicações que funcionem em diversas plataformas (IOS, Android, desktop e Web) com apenas um único código (Ionic Framework, 2019).
- **NPM:** é o nome reduzido de *Node Package Manager* (Gerenciador de Pacotes do Node). O NPM é duas coisas: primeiro, e mais importante, é um repositório *online* para publicação de projetos de código aberto para o Node.js; segundo, é um utilitário de linha de comando que interage com este repositório *online*, que ajuda na instalação de pacotes, gerenciamento de versão e gerenciamento de dependências. (NODE BR, 2019).
- **Sass:** é uma extensão de CSS que adiciona poder e elegância ao CSS básico, essa que é uma Gem do Ruby, ou seja, uma biblioteca escrita em linguagem Ruby. Esse pré-processador, como dito anteriormente, permite que você use variáveis, regras aninhadas, importações e mais, tudo com uma sintaxe totalmente compatível com CSS (GONÇALVES, 2017).
- **Trello:** é uma ferramenta de gerenciamento de projetos em listas extremamente versátil e que pode ser ajustada de acordo com as necessidades do usuário. Você pode utilizá-lo para organizar as suas tarefas do trabalho, os seus planos de viagens, as prioridades de seus estudos, entre muitos outros. (CASTELLI, 2019).
- **TypeScript:** é um super conjunto de Java Script, o que significa que ele contém toda a funcionalidade do Java Script e mais algumas. Portanto, qualquer programa escrito em JavaScript válido também será executado conforme o esperado no TypeScript (LUIS, 2019).

- **Visual Studio Code:** é um editor de código leve, multiplataforma, gratuito e *open source*. Mantido pela Microsoft, essa ferramenta conta com suporte a várias linguagens, extensões, integração com Git, debug, terminal integrado, entre outros recursos (DEV MEDIA, 2019).
- **WhatsApp:** é um aplicativo de troca de mensagens e comunicação em áudio e vídeo pela internet, disponível para smartphones Android, iOS, Windows Phone, Nokia e computadores Mac e Windows (CARVALHO, 2019).

3. MODELAGEM

*Unified Modeling Language*² (UML) ou Linguagem de Modelagem Unificada (em português) é uma notação padrão para modelagem de objetos do mundo real. Trata-se de uma linguagem de modelagem de desenvolvimento de software que se destina a fornecer uma maneira padrão de visualizar o "desenho" de um sistema (PORTAL GSTI, 2019).

3.1 Diagramas de Caso de Uso

O Diagrama de Casos de Uso auxilia no levantamento dos requisitos funcionais do sistema, descrevendo um conjunto de funcionalidades do sistema e suas interações com elementos externos e entre si (VIEIRA, 2015). Dentro do diagrama são retratados os atores (representado pelos bonecos), as funcionalidades (representadas pelos balões com a ação escrita por dentro) e as relações (representadas pelas linhas).

Os atores que interagem com o sistema são: o Usuário, o Firebase e as API do Dialogflow, Google Maps e do ViaCEP. O sistema é um caso de uso explícito e se trata do sistema em si em que os casos de uso acontecem.

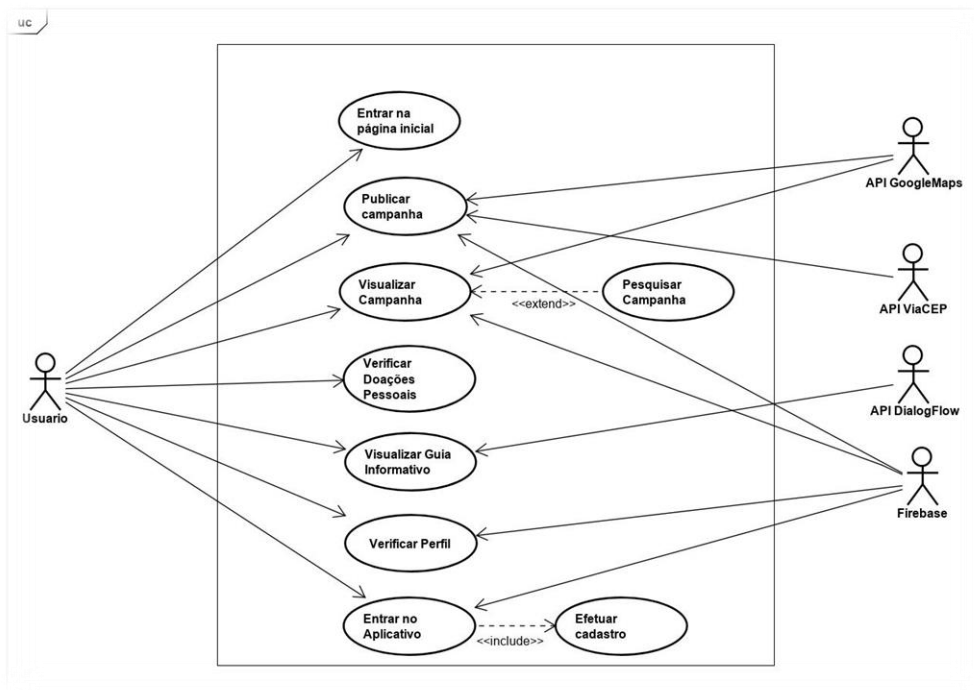
- **Usuário** é o ator que representa os clientes deste aplicativo. Ele pode publicar, visualizar e filtrar campanhas, controlar doações pessoais, acessar guia informativa e verificar seu perfil.
- **Firestore** representa o banco de dados em tempo real, onde o aplicativo armazena as informações de usuários (doador e divulgador) e as campanhas publicadas.
- **API Dialogflow** representa a API do campo de aprendizado de máquina utilizando modelos de compreensão de *natural language* para identificar padrões em mensagens de texto diante de um vocabulário conhecido, interpretar contextos no diálogo humano-computador e reagir com respostas predeterminadas.

² *Unified Modeling Language* ou Linguagem Unificada de Modelagem (UML) é uma linguagem padrão para modelagem e documentar os sistemas orientados a objetos.

- **API Google Maps** representa a API de conexão do mapa com o aplicativo, onde é exibido cada localização apontada pelo mapa.
- **API ViaCEP** representa a API que fornece dados relacionados ao CEP informado pelo usuário, como logradouro, cidade e estado.

A figura 4 ilustra o diagrama de caso de uso que demonstra como usar o aplicativo.

Figura 4 - Diagrama de caso de uso do aplicativo.



Fonte: Elaborado pelo autor (2019).

3.1.1 Documentação dos casos de uso

Cada funcionalidade dos diagramas de casos de uso será descrita da Tabela 4 à Tabela 10.

Tabela 4 - Caso de uso “Publicar campanha”.

Nome do caso de uso	Publicar campanha.
Atores envolvidos	Usuário, Firebase, API Google Maps, API ViaCEP.
Objetivo	Este caso de uso descreve as ações de campanhas que podem ser feitas e publicadas.

Prioridade de desenvolvimento	Imprescindível.
Ações do ator	Ações do Sistema
1. O usuário clica em um campo para digitar as informações da campanha, como: local, tipo de campanha e data e sem seguida clica para publicar.	1. O aplicativo realiza uma requisição ao Firebase para armazenar e salvar as informações.
Validações	Haverá um moderador para que filtre mensagens que não dizem respeito as campanhas.

Fonte: Elaborado pelo autor (2019).

Tabela 5 - Caso de uso “Visualizar campanha”.

Nome do caso de uso	Visualizar campanha.
Atores envolvidos	Usuário, Firebase, API Google Maps.
Objetivo	Este caso de uso mostra ao usuário as informações pertinentes a campanha publicada.
Prioridade de desenvolvimento	Essencial.
Ações do ator	Ações do Sistema
1. O usuário clica na campanha X e visualiza as informações relacionadas a ela, tais como: tipo de campanha, local e data.	1. O aplicativo realiza uma requisição ao Firebase, traz a lista de campanhas que foram armazenadas na base de dados e as exibe em uma página de visualização ao usuário. 2. O aplicativo faz uma requisição a API do Google Maps que exibe o mapa da respectiva localização da campanha.
Validações	Não há validações para esse caso de uso.

Fonte: Elaborado pelo autor (2019).

Tabela 6 - Caso de uso “Minhas Doações”.

Nome do caso de uso	Minhas Doações.
Atores envolvidos	Usuário.

Objetivo	Neste caso de uso o doador adiciona os locais onde ele doou sangue e a data para saber quando será possível doar novamente.
Prioridade de desenvolvimento	Imprescindível.
Ações do ator	Ações do Sistema
1. O doador pode consultar e adicionar suas doações.	
1. Através do histórico o doador também saberá quando poderá doar de novo.	2. O aplicativo com a data da última doação realiza um cálculo de acordo com o gênero do doador e notifica quando é possível realizar outra doação.
Validações	Não há validações para esse caso de uso.

Fonte: Elaborado pelo autor (2019).

Tabela 7 - Caso de uso “Guia informativo”.

Nome do caso de uso	Guia informativo
Atores envolvidos	Usuário, API Dialogflow.
Objetivo	Informar o usuário a respeito de informações pertinentes ao processo que antecede e que sucede o período de doação e esclarecer dúvidas.
Prioridade de desenvolvimento	Essencial.
Ações do ator	Ações do Sistema
1.1.7 O usuário ao clicar na página poderá fazer perguntas no chat para obter informações no mesmo momento.	1. O aplicativo realiza uma requisição para a API do Dialogflow que busca as informações e identifica padrões em mensagens de texto diante de um vocabulário conhecido, interpreta contextos no diálogo com o usuário e reage com respostas predeterminadas para esclarecer dúvidas.
Validações	Não há validações para esse caso de uso.

Fonte: Elaborado pelo autor (2019).

Tabela 8 - Caso de uso “Minha Conta”.

Nome do caso de uso	Minha Conta.
Atores envolvidos	Usuário, Firebase.
Objetivo	Este caso de uso exibe o perfil com as informações pessoais do usuário
Prioridade de desenvolvimento	Essencial.
Ações do ator	Ações do Sistema
1. O doador ao clicar sobre o botão no menu se redireciona para página onde visualiza seus dados.	1. O aplicativo realiza uma requisição ao Firebase para trazer a lista de informações.
2. Opcionalmente o usuário pode alterar os dados pessoais e clicar no botão para salvar.	2. O aplicativo realiza uma requisição ao Firebase para armazenar e salvar as informações.
Validações	Não há validações para esse caso de uso.

Fonte: Elaborado pelo autor (2019).

Tabela 9 - Caso de uso “Efetuar Cadastro”.

Nome do caso de uso	Efetuar Cadastro.
Atores envolvidos	Usuário, Firebase.
Objetivo	Este caso de uso descreve os passos do cadastro de um usuário no aplicativo.
Prioridade de desenvolvimento	Essencial.
Ações do ator	Ações do Sistema
1. O usuário digita seu e-mail, cria uma senha de seis dígitos e clica no botão para se cadastrar.	1. Uma requisição ao Firebase é realizada pelo aplicativo para registrar os dados do usuário na base de dados para que possam ser autenticados no login.
	2. Após registrado na base de dados pelo Firebase, o aplicativo troca a página exibindo a tela de login.
Validações	Para que o cadastro seja efetuado, é necessário informar um e-mail com padrão válido (x@x.x), e a senha deve ter seis dígitos no mínimo.

Fonte: Elaborado pelo autor (2019).

Tabela 10 - Caso de uso “Entrar no Aplicativo”.

Nome do caso de uso	Entrar no Aplicativo.
Atores envolvidos	Usuário, Firebase.
Objetivo	Este caso de uso descreve os passos do login de um usuário no aplicativo.
Prioridade de desenvolvimento	Essencial.
Ações do ator	Ações do Sistema
1. O usuário digita seu e-mail e a senha de 6 dígitos e clica no botão de login.	1. As informações digitadas são autenticadas através de uma requisição feita que consulta na base de dados.
	2. Após essa autenticação, o aplicativo troca a página exibindo a tela inicial.
Validações	Para que o login seja efetuado, o usuário deve inserir seu e-mail e sua senha de pelo menos seis dígitos. Caso o e-mail ou senha estejam errados ou o cadastro tenha sido desabilitado na base de dados, uma mensagem de erro é exibida.

Fonte: Elaborado pelo autor (2019).

3.1.2 Casos de teste

Um Caso de Teste mostra os caminhos percorridos por um módulo, Caso de Uso ou funcionalidade dentro do projeto. Serve como base para que os testadores possam executar os testes manualmente, mas pode ser criado, também, com o intuito de automatizar os testes. Além disso, os Casos de Teste devem cobrir o máximo de situações possíveis. As tabelas 11 a 17 mostram de forma detalhada os Casos de Testes realizados.

Tabela 11 - Caso de Teste do login

	Caso de Teste	
	Login do usuário	Não informar o e-mail ou digitar um e-mail fora do padrão x@x.x.
Não informar a senha ou informar uma senha com menos de 6 caracteres.		É exibido a mensagem “Precisa ter ao menos 6 caracteres”.
Estar na tela de login.		O menu não fica acessível.
Atender os padrões de e-mail e senha.		O botão para fazer login fica em uma cor mais escura e habilitado para ser clicado.
Usuário fez a solicitação de login e foi retornado que seu e-mail ou senha estão incorretos.		É exibido a mensagem “A senha é inválida ou o usuário não tem uma senha”.
Usuário fez a solicitação de login e foi retornado que o usuário está desabilitado.		É exibido a mensagem “Usuário desabilitado. Contate os administradores para mais detalhes”.

Fonte: Elaborado pelo autor (2019).

Tabela 12 - Caso de Teste do cadastro

	Caso de Teste	
	Cadastro do usuário	Não informar o e-mail ou digitar um e-mail fora do padrão x@x.x.
Não informar a senha ou informar uma senha com menos de 6 caracteres.		É exibido a mensagem “Precisa ter ao menos 6 caracteres”.
Estar na tela de cadastro.		O menu não fica acessível.
Atender os padrões de e-mail e senha.		O botão para realizar cadastro fica em uma cor mais escura e habilitado para ser clicado.
Usuário fez a solicitação e foi retornado que o e-mail está em uso.		É exibido a mensagem “O e-mail informado já está sendo utilizado”.

Fonte: Elaborado pelo autor (2019).

Tabela 13 - Caso de Teste do publicar campanha

Publicar Campanha	Casos de Teste	
	Não informar ou digitar mais que 60 caracteres para o nome do hemocentro.	É exibido um alerta “Digite um nome de até 60 caracteres”.
	Não informar um CEP com o padrão correto de 8 dígitos.	É exibido um alerta “Informe um CEP válido, com 8 caracteres”.
	Não informar o endereço.	É exibido um alerta “É necessário informar o endereço”.
	Não informar o número com o padrão de 1 a 5 caracteres numéricos.	É exibido um alerta “É necessário informar o número”.
	Não informar a cidade.	É exibido um alerta “É necessário informar a cidade”.
	Não informar a sigla de 2 dígitos que correspondem ao estado (UF).	É exibido um alerta “É necessário informar a sigla do estado”.
	Não informar os campos “tipo de doação” ou “tipo de sangue” ou “eu sou o paciente”.	É exibido a mensagem no final da tela “Campos que ainda precisam da sua atenção” e a lista de campos inválidos.
	Atender os padrões de validação.	O botão fica em uma cor mais escura e habilitado para ser clicado.

Fonte: Elaborado pelo autor (2019).

Tabela 14 - Caso de Teste do visualizar campanha

Visualizar Campanha	Casos de Teste	
	Não selecionar um estado ou tipo de sangue ou não digitar a cidade desejada para pesquisar.	O botão para realizar pesquisa fica desabilitado e visualmente fica com menos opacidade e sem animação de clique.

Fonte: Elaborado pelo autor (2019).

Tabela 15 - Caso de Teste do guia informativo

Guia Informativo	Casos de Teste	
	Não informar corretamente o que deseja perguntar, possuindo algum erro de ortografia.	O <i>chatbot</i> vai tentar entender a pergunta realizada e responder de acordo, caso o que foi digitado esteja com poucos erros

		ortográficos e dentro dos padrões de diálogos definidos.
--	--	--

Fonte: Elaborado pelo autor (2019).

Tabela 16 - Caso de Teste do minhas doações

	Casos de Teste	
	Cadastrar Doação	Não informar o tipo de doação.
	Não informar a data da doação.	É exibido a mensagem “É necessário informar a data da doação”.
	Não informar o nome do hemocentro.	É exibido a mensagem “É necessário informar o nome do hemocentro”.
	Atender os padrões de preenchimento da tela.	O botão para realizar o cadastro fica em uma cor mais escura e habilitado para ser clicado.

Fonte: Elaborado pelo autor (2019).

Tabela 17 - Caso de Teste da minha conta

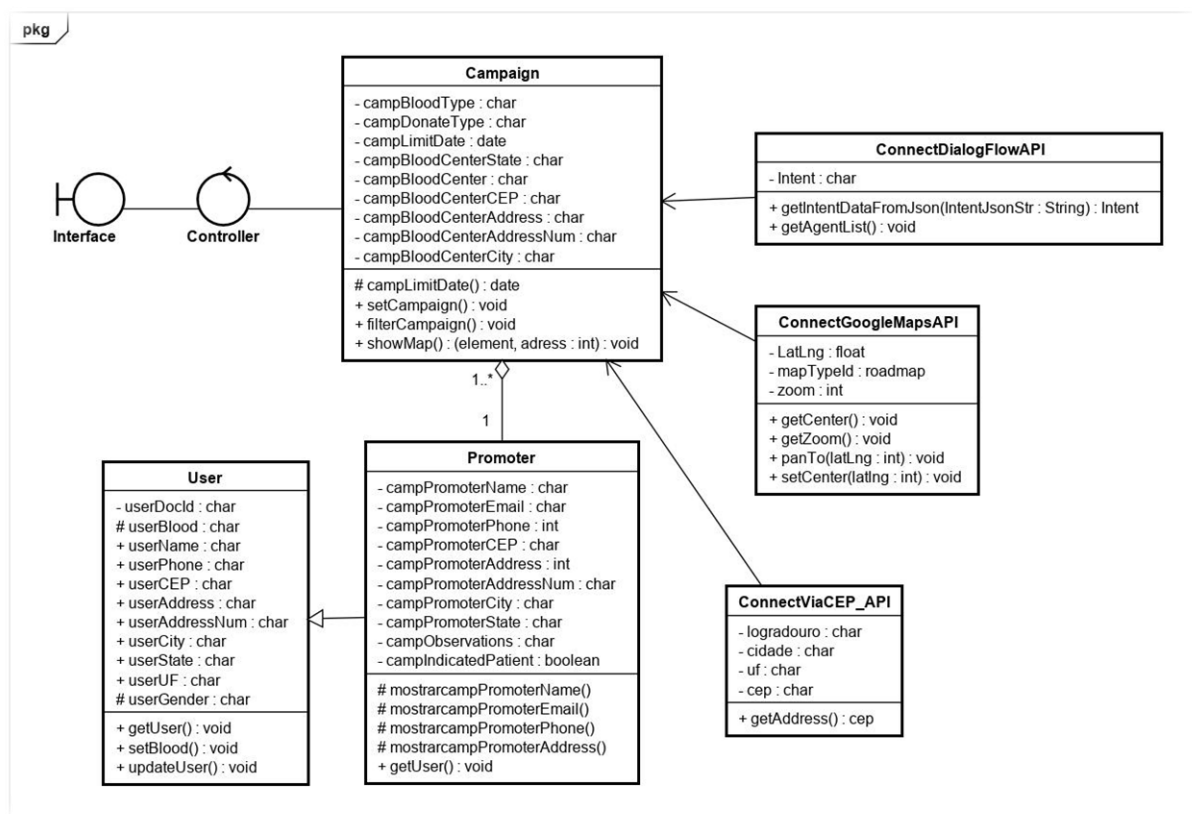
	Casos de Teste	
	Minha Conta	Não informar um CEP com o padrão correto de 8 dígitos.
	Não informar o endereço.	É exibido um alerta “É necessário informar o endereço”.
	Não informar o número com o padrão de 1 a 5 caracteres numéricos.	É exibido um alerta “É necessário informar o número”.
	Não informar a cidade.	É exibido um alerta “É necessário informar a cidade”.
	Não informar a sigla de 2 dígitos que correspondem ao estado (UF).	É exibido um alerta “É necessário informar a sigla do estado”.
	Não informar o tipo de sangue.	É exibido um alerta “É necessário informar o tipo de sangue”.
	Atender os padrões de preenchimento da tela.	O botão para atualizar cadastro fica em uma cor mais escura e habilitado para ser clicado.

Fonte: Elaborado pelo autor (2019).

3.2 Diagrama de Classes

O diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos. Uma classe é representada por um retângulo com três divisões, são elas: O nome da classe, seus atributos e por fim os métodos (DEV MEDIA, 2019). Para entender melhor a estrutura funcional do projeto, a Figura 5 apresenta as principais classes do aplicativo MyBlood.

Figura 5 - Diagrama de classes do aplicativo



Fonte: Elaborado pelo autor (2019).

A classe *ConnectDialogFlowAPI* é responsável por criar e gerenciar a conexão com a API do Dialogflow. Possuindo dois métodos que são apresentados na Tabela 18.

Tabela 18 - Requisitos não funcionais do projeto.

Método	Descrição
--------	-----------

<i>getIntentAndAgentDataFromJson</i>	Retorna um JSON com todas as intenções no agente especificado.
<i>getAgentList</i>	Processa uma consulta de <i>natural language</i> e retorna um JSON de dados estruturados e acionáveis como resultado.

Fonte: Elaborado pelo autor (2019).

A classe *ConnectGoogleMapsAPI* é responsável por criar e gerenciar a conexão do aplicativo com a API do Google Maps, possuindo quatro métodos que são apresentados na tabela 19.

Tabela 19 - Requisitos não funcionais do projeto.

Método	Descrição
<i>getCenter</i>	Retorna a posição exibida no centro do mapa com as coordenadas geográficas em latitude (varia entre -90 e 90 graus e longitude (varia entre -180 e 180 graus).
<i>getZoom</i>	Retorna um número e maximiza ou minimiza a visualização do mapa.
<i>panTo</i>	Muda o centro do mapa para o dado latitude e longitude, se a alteração for menor que a largura e a altura do mapa, a transição será animada sem problemas.
<i>setCenter</i>	Retorna como parâmetros os valores de latitude e longitude.

Fonte: Elaborado pelo autor (2019).

A classe *ConnectCEP_API* é responsável por criar e gerenciar a conexão com a API do ViaCEP. Possuindo um método que é apresentado na Tabela 20.

Tabela 20 - Requisitos não funcionais do projeto

Método	Descrição
<i>getAddress</i>	Retorna o CEP a partir do nome do logradouro, localidade e unidade federativa. Esse método requer autenticação de usuário. Se autenticado, retorna as 20 primeiras linhas encontradas.

Fonte: Elaborado pelo autor (2019).

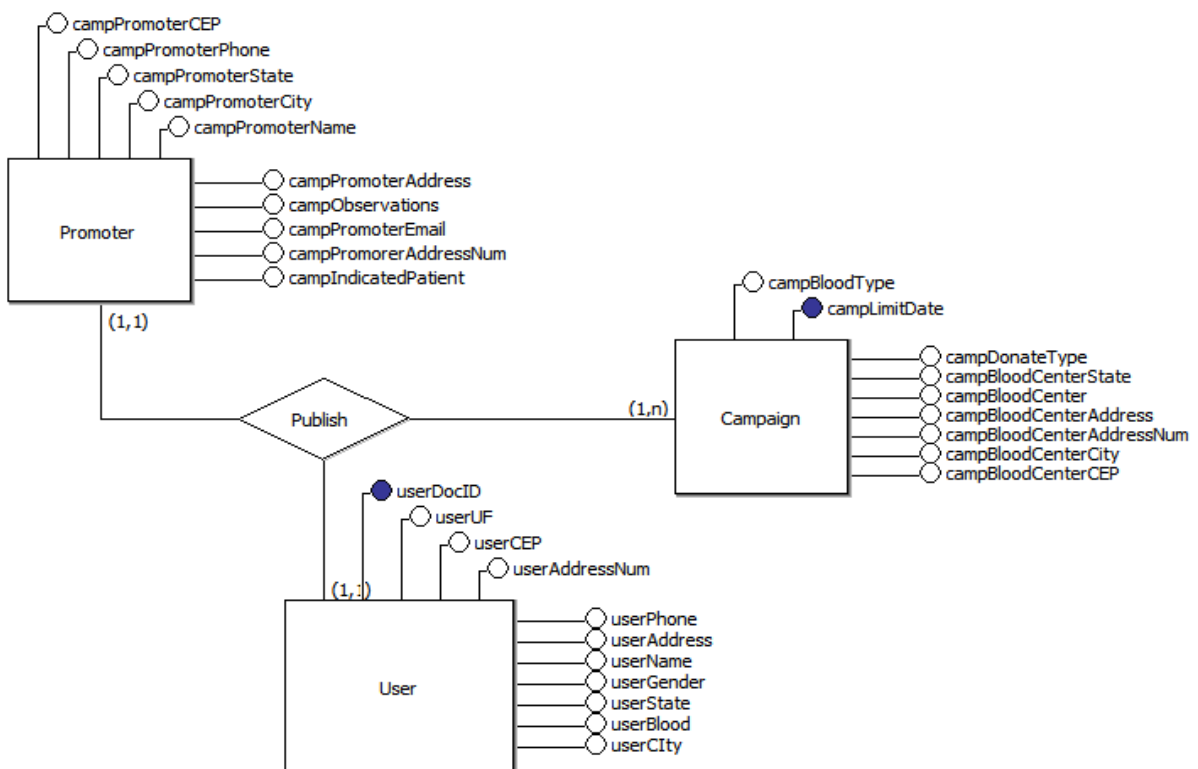
3.3 Banco de Dados

O Banco de Dados é um ambiente onde é possível armazenar dados de forma organizada, gerando conjuntos de arquivos. Ele permite o agrupamento de informações que, posteriormente, podem ser extraídas para outras funções.

3.3.1 Diagrama de Entidade e Relacionamento (DER)

O Diagrama Entidade Relacionamento é um modelo diagramático que descreve o modelo de dados de um sistema com alto nível de abstração. Ele é a principal representação do Modelo de Entidades e Relacionamentos. Sua maior aplicação é para visualizar o relacionamento entre tabelas de um banco de dados, no qual as relações são construídas através da associação de um ou mais atributos destas tabelas. A figura 6 apresenta o DER do aplicativo MyBlood.

Figura 6 - DER do banco de dados armazenado no Cloud Firestore



Fonte: Elaborado pelo autor (2019).

3.3.2 Dicionário de Dados

Um dicionário de dados contém as características lógicas dos dados que serão utilizados no sistema, incluindo nome, descrição, conteúdo e organização.

Estes dicionários se desenvolvem durante a análise de fluxo de dados e ajuda aos analistas que participam na determinação dos requerimentos do sistema, seu conteúdo também se emprega durante o desenho do projeto (ALVAREZ, 2019). As tabelas devem conter os seguintes campos:

Entidade: é uma pessoa, objeto ou lugar que será considerado como objeto que há o interesse em guardar informações a seu respeito.

Atributo: os atributos são as características da entidade que devem ser guardados.

Classe: as classes podem ser: simples, composto, multivalorado e determinante. Simples indica um atributo normal. Composto indica que ele poderá ser dividido em outros atributos como, por exemplo, o endereço. Multivalorado é quando o valor do atributo poderá ou não ser único e determinante é um atributo que será usado como chave, CPF, ID, etc.

Domínio: podem ser numéricos, texto, data e *boolean*. Pode-se chamar também de tipo de valor que o atributo irá receber.

Tamanho: define a quantidade de caracteres que serão necessários para armazenar o seu conteúdo. Geralmente o tamanho só é definido para atributos de domínio texto.

Descrição: é opcional e pode ser usado para descrever o que é aquele atributo ou dar informações adicionais que possam ser usadas futuramente pelo analista, ou programador do sistema.

As tabelas 21 a 25 apresentam o dicionário de dados de cada entidade.

Tabela 21 - Entidade Divulgador

Entidade: Divulgador				
Atributo	Classe	Tipo	Tamanho	Descrição

Paciente Indicado	Multivalorado	Texto	Até 1.048.487 bytes (1 MiB — 89 bytes)	Poderá ou não haver indicação de um paciente
-------------------	---------------	-------	---	--

Fonte: Elaborado pelo autor (2019).

Tabela 22 - Entidade Campanha

Entidade: Campanha				
Atributo	Classe	Tipo	Tamanho	Descrição
Data Limite	Determinante	Data	de 1 de janeiro de 1753 a 31 de dezembro de 9999,	Indicador da data limite para a campanha
Observações	Simple	Texto	Até 1.048.487 bytes (1 MiB — 89 bytes)	Campo de observações com limite de 200 caracteres

Fonte: Elaborado pelo autor (2019).

Tabela 23 - Entidade Doador

Entidade: Doador				
Atributo	Classe	Tipo	Tamanho	Descrição
Tipo de doação	Multivalorado	Numérico	64 bits	Sangue ou Plaquetas
Tipo de sangue	Simple	Texto	Até 1.048.487 bytes (1 MiB — 89 bytes)	Selecionar apenas um tipo sanguíneo
Gênero	Simple	Texto	Até 1.048.487 bytes (1 MiB — 89 bytes)	Masculino, feminino ou outros.

Fonte: Elaborado pelo autor (2019).

Tabela 24 - Entidade Hemocentro

Entidade: Hemocentro				
Atributo	Classe	Tipo	Tamanho	Descrição
Endereço	Simple	Texto	Até 1.048.487 bytes (1 MiB — 89 bytes)	Pode ter até 100 caracteres
CEP	Determinante	Numérico	64 bits	Precisa ter exatamente 8 dígitos

Cidade	Simple	Texto	Até 1.048.487 bytes (1 MiB — 89 bytes)	Pode ter no máximo 35 caracteres
Número	Simple	Numérico	64 bits	Pode ter no máximo 5 caracteres
UF	Simple	Texto	Até 1.048.487 bytes (1 MiB — 89 bytes)	Precisa ter exatamente 2 dígitos
Nome	Simple	Texto	Até 1.048.487 bytes (1 MiB — 89 bytes)	Pode ter até 15 caracteres

Fonte: Elaborado pelo autor (2019).

Tabela 25 - Entidade Usuário

Entidade: Usuário				
Atributo	Classe	Tipo	Tamanho	Descrição
ID	Determinante	Numérico	64 bits	Identificador único de usuário, com criptografia
CEP	Simple	Numérico	64 bits	Precisa ter exatamente 8 dígitos
Cidade	Determinante	Texto	Até 1.048.487 bytes (1 MiB — 89 bytes)	Pode ter no máximo 35 caracteres
E-mail	Multivalorado	Texto	Até 1.048.487 bytes (1 MiB — 89 bytes)	Validação para que o usuário forneça um e-mail válido
Endereço	Multivalorado	Texto	Até 1.048.487 bytes (1 MiB — 89 bytes)	Pode ter até 100 caracteres
Número	Simple	Texto	Até 1.048.487 bytes (1 MiB — 89 bytes)	Pode ter até 5 caracteres
Nome	Simple	Texto	Até 1.048.487 bytes (1 MiB — 89 bytes)	Pode ter até 15 caracteres
Telefone	Multivalorado	Texto	Até 1.048.487 bytes (1 MiB — 89 bytes)	Precisa ter exatamente 11 dígitos
UF	Simple	Texto	Até 1.048.487 bytes (1 MiB — 89 bytes)	Precisa ter exatamente 2 dígitos

Fonte: Elaborado pelo autor (2019).

4. DESENVOLVIMENTO

Esse capítulo demonstra como foi dividido as atividades a serem realizadas, o prazo e pontuação atribuídos para cada tarefa e as etapas de desenvolvimento.

4.1 Arquitetura e organização

A aplicação está implementada através do *framework* Ionic de desenvolvimento de aplicativos híbridos. O Ionic é considerado uma junção e complementação de outros *frameworks*, o Angular e o Apache Cordova, que são incluídos por padrão ao criar um projeto pelo terminal de comandos do Ionic.

O Angular é uma plataforma para a construção de aplicações em HTML e TypeScript, que geralmente são projetadas para serem executadas em navegadores de internet. Os principais conceitos que o Ionic utiliza dessa plataforma são:

- **Módulos:** declaram um conjunto de componentes e funcionalidades que serão dedicados para a aplicação em um contexto único. Todo projeto em Angular tem um módulo root, que concentra as importações e declarações de componentes, serviços e outros códigos, e é responsável pela inicialização da aplicação.
- **Componentes:** Cada componente define uma classe que contém uma parte da lógica ou apresentação de dados da aplicação. Geralmente composto pelas seguintes partes:
 - Visualização (View): parte responsável pela apresentação das informações na interface com a linguagem de marcação HTML.
 - Estilo (Style): responsável pela estilização da interface que será apresentada, usando CSS, uma linguagem de folhas de estilos.
 - Classe (Class): tem a função de intermediar a comunicação entre serviços/funcionalidades e a View. Faz a preparação ou atualização dos dados recebidos pelos serviços para serem apresentados na View e recebe as interações feitas pelo usuário na interface e as processam, gerando um ciclo.

- Teste (Test): contém métodos para testar a funcionalidade do componente.
- Módulo (Module): declara importações, exportações, o domínio do componente e serviços para a aplicação.

Toda aplicação em Angular tem ao menos um componente root, que inicia a hierarquia de componentes e os conecta com a *Document Object Model* (DOM) para que as estruturas sejam geradas dinamicamente durante o fluxo de execução.

- **Serviços e injeção de dependências:** quando algum dado ou lógica de negócio não é associado apenas a uma única parte da aplicação e é necessário compartilhar o recurso, cria-se uma classe do tipo *service*. Essa classe deve ser declarada como injetável para que seja injetada como dependência no momento da construção das classes dos componentes que vão utilizá-la.

O Apache Cordova no que lhe concerne, é um *framework* de desenvolvimento para dispositivos móveis. Ele permite que os desenvolvedores utilizem tecnologias web com padrões como HTML, CSS e JavaScript para um desenvolvimento *cross-plataform*. As aplicações que utilizam do Apache Cordova são executadas dentro de pacotes destinados para cada plataforma (Android, iOS etc.) e podem contar com APIs padronizadas para acessar recursos nativos dos celulares, como sensores, status da conexão e outras informações do sistema.

Além de utilizar dos recursos dessas plataformas, o Ionic tem as próprias implementações e recursos padrões que aceleraram o desenvolvimento:

- **Ionic Command Line Interface:** facilita a criação de projetos e inclusão de páginas e serviços através de comandos simples executados no terminal.
- **Servidor local de desenvolvimento:** para verificar como a aplicação está sendo executada no navegador.
- **Ionic Lab:** para simular a execução da aplicação em dispositivos móveis das plataformas Android, iOS e Windows Phone.
- **UI Components e Theming:** possui uma série de componentes prontos para utilização, que se adaptam a cada plataforma para trazer melhor experiência para o usuário.

- **Ionicons:** conjunto de ícones mais utilizados que também se adaptam a cada plataforma e está disponível por padrão para utilização.

Neste projeto, esse *framework* está sendo utilizado em conjunto com serviços do Firebase, que exerce a função de *Back-end as a Service* (BaaS), mas outras possibilidades de *back-end* como em Laravel, NodeJS e Ruby on Rails também são suportadas e utilizadas, basta possibilitar o consumo de APIs.

4.2 API

A API (*Application Programming Interface* ou Interface de Programação de Aplicações) é um conjunto de definições e protocolos que possibilitam a construção e integração de aplicações. As APIs permitem que produtos e serviços se comuniquem entre si, sem precisarem se importar em como estão implementadas, o que pode simplificar o desenvolvimento de software e economizar tempo e dinheiro.

Por conta de as APIs simplificarem o modo em que novos componentes podem ser implementados em uma estrutura existente, elas ajudam organizações e times de TI a realizar colaborações. Os negócios precisam mudar cada vez mais rápido para responder às mudanças do mercado digital, então é importante conseguir desenvolver rapidamente e publicar produtos e serviços inovadores.

4.2.1 API do Firebase Authentication

Essa API fornece os métodos necessários para se conectar ao serviço de autenticação do Firebase, o Firebase Authentication e realizar as seguintes ações: criar conta utilizando o e-mail, realizar login com e-mail e senha, sair da conta, pedir redefinição de senha por e-mail, pegar dados do usuário logado e persistir dados da sessão no dispositivo entre outras ações disponíveis, a figura 7 exibe o código para consumir essa API.

Figura 7 - API do Firebase Authentication

```

import { Injectable } from '@angular/core';
import { AngularFireAuth } from 'angularfire2/auth';
import * as firebase from 'firebase/app';

@Injectable()
export class AuthService {
  private user: firebase.User;

  constructor(public afAuth: AngularFireAuth) {
    afAuth.authState.subscribe(user => {
      this.user = user;
    });
  }

  // Login with e-mail and password
  public signInWithEmail(credentials) {
    return this.afAuth.auth.signInWithEmailAndPassword(credentials.e
mail, credentials.password);
  }

  // Create an account with e-mail and password
  public signUp(credentials) {
    return this.afAuth.auth.createUserWithEmailAndPassword(credentia
ls.email, credentials.password);
  }

  // Create an account with e-mail and password
  public resetPassword(email) {
    return this.afAuth.auth.sendPasswordResetEmail(email);
  }

  // Sign out of the user
  public signOut(): Promise<void> {
    return this.afAuth.auth.signOut();
  }

  // Return data about the logged user
  public getUserData() {
    return this.user;
  }

  public setPersistence() {
    return this.afAuth.auth.setPersistence(firebase.auth.Auth.Persis
tence.LOCAL)
  }
}

```

Fonte: Elaborado pelo autor (2019).

4.2.2 API do Cloud Firestore

Essa Interface de Programação de Aplicação fornece um meio fácil de se comunicar com o serviço de gerenciamento de dados do Cloud Firestore, possibilitando que consultas e inserções na base de dados sejam feitas através de funções parametrizadas. Pode-se ver nas figuras 8, 9 e 10 o código que consome a API.

Figura 8 - API do Cloud Firestore

```
import { Injectable } from '@angular/core';
import { AngularFireAuth } from 'angularfire2/auth';
import { AngularFirestore } from 'angularfire2/firestore';
import { AngularFirestoreCollection } from 'angularfire2/firestore';
import { Observable } from 'rxjs/Observable';
import 'rxjs/add/operator/catch';
import 'rxjs/add/operator/map';

// (...)

@Injectable()
export class FirestoreService {

  // (...)

  // Returns logged user donations
  public getDonation(): Observable<Donation[]> {
    if (this.afAuth.auth.currentUser) {
      return this.donations
        .map(donations => {
          return donations.map((donation) => new Donation(dona
tion));
        });
    }
  }

  // Creates new donation
  public setDonation(donatUserID, donatBloodCenter, donatDate, donatTy
pe) : boolean {
    if (this.afAuth.auth.currentUser) {
      if (this.donationsCollectionRef.add({ donatUserID: donatUser
ID, donatBloodCenter: donatBloodCenter, donatDate: new Date(donatDate),
donatType: donatType })) {
        return true;
      } else {
        return false;
      }
    }
  }
}
```

Fonte: Elaborado pelo autor (2019)

Figura 9 - API do Cloud Firestore parte 2

```

    }
  }
  return false;
}

// Returns all campaigns
public getCampaign() {
  if (this.afAuth.auth.currentUser)
    return this.campaigns;
}

// Creates new campaign on campaigns collection
public setCampaign(campBloodType = '', campDonateType = '', campLimitDate = null, campBloodCenter = '', campBloodCenterCEP = '', campBloodCenterAddress = '', campBloodCenterAddressNum = '', campBloodCenterCity = '', campBloodCenterState = '', campPromoterName = '', campPromoterEmail = '', campPromoterPhone = '', campPromoterCEP = '', campPromoterAddress = '', campPromoterAddressNum = '', campPromoterCity = '', campPromoterState = '', campPromoterIsPatient = false, campIndicatedPatient = '', campObservations = '') {
  if (this.afAuth.auth.currentUser) {
    if (this.campaignsCollectionRef.add({
      campBloodType: campBloodType,
      campDonateType: campDonateType,
      campLimitDate: new Date(campLimitDate),
      campBloodCenter: campBloodCenter,
      campBloodCenterCEP: campBloodCenterCEP,
      campBloodCenterAddress: campBloodCenterAddress,
      campBloodCenterAddressNum: campBloodCenterAddressNum,
      campBloodCenterCity: campBloodCenterCity,
      campBloodCenterState: campBloodCenterState,
      campPromoterName: campPromoterName,
      campPromoterEmail: campPromoterEmail,
      campPromoterPhone: campPromoterPhone,
      campPromoterCEP: campPromoterCEP,
      campPromoterAddress: campPromoterAddress,
      campPromoterAddressNum: campPromoterAddressNum,
      campPromoterCity: campPromoterCity,
      campPromoterState: campPromoterState.toUpperCase(),
      campPromoterIsPatient: campPromoterIsPatient,
      campIndicatedPatient: campIndicatedPatient,
      campObservations: campObservations
    })) {
      return true;
    }
    return false;
  }
  return false;
}

```

Fonte: Elaborado pelo autor (2019).

Figura 10 - API do Cloud Firestore parte 3

```

    }

    // Filters campaigns by state and city
    public filterCampaign(campBloodCenterState = '', campBloodCenterCity
= '', campBloodType = '') {
        if (this.afAuth.auth.currentUser) {

            // If we state and city is provided
            if (campBloodCenterState && campBloodCenterCity && campBlood
Type) {
                return this.angularFirestore.collection<Campaign>('campa
igns', ref => ref.where('campBloodCenterState', '==', campBloodCenterSta
te).where('campBloodCenterCity', '==', campBloodCenterCity).where('campB
loodType', '==', campBloodType)).valueChanges();
            }

            // If only state is provided
            else if(campBloodCenterState) {
                return this.angularFirestore.collection<Campaign>('campa
igns', ref => ref.where('campBloodCenterState', '==', campBloodCenterSta
te)).valueChanges();
            }

            //if only state is provided
            else if(campBloodType) {
                return this.angularFirestore.collection<Campaign>('campa
igns', ref => ref.where('campBloodType', '==', campBloodType)).valueChan
ges();
            }

            // If only city is provided
            else if(campBloodCenterCity) {
                return this.angularFirestore.collection<Campaign>('campa
igns', ref => ref.where('campBloodCenterCity', '==', campBloodCenterCity
)).valueChanges();
            }

            // If no data is passed
            else {
                return this.angularFirestore.collection<Campaign>('campa
igns').valueChanges();
            }
        }
    }
}

```

Fonte: Elaborado pelo autor (2019).

4.2.3 API do ViaCEP

Esta é uma API onde é enviado apenas um CEP, e caso ele seja válido é retornado uma série de informações sobre ele, como logradouro, cidade e estado, informações bastante úteis para preencher automaticamente o cadastro do usuário, agilizando o processo. A figura 11 demonstra o uso da API.

Figura 11 - API do ViaCEP

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import 'rxjs/add/operator/toPromise';

@Injectable()
export class CepService {
  constructor(private http:HttpClient) { }

  public getAddress(cep:string):any {
    return this.http
      .get(`https://viacep.com.br/ws/${cep}/json/`)
  }
}
```

Fonte: Elaborado pelo autor (2019).

Exemplo de resposta para o CEP “13173-301”, exibido na figura 12:

Figura 12 - API do ViaCEP exemplo

```
{
  "cep": "13173-301",
  "logradouro": "Rua Crenac",
  "complemento": "",
  "bairro": "Jardim Basilicata",
  "localidade": "Sumaré",
  "uf": "SP",
  "unidade": "",
  "ibge": "3552403",
  "gia": "6713"
}
```

Fonte: Elaborado pelo autor (2019).

4.2.4 Maps JavaScript API e Geocoding API

Dentre as várias APIs disponibilizadas pelo Google, neste projeto foi utilizada a Maps JavaScript API, para criar pequenos mapas navegáveis e a Geocoding API, para buscar as coordenadas de endereços em texto, que posteriormente são utilizadas para criar marcadores nos mapas gerados, as figuras 13 e 14 demonstram o código para utilização dessas APIs.

Figura 13 - Maps JavaScript API e Geocoding API

```
import { Component, AfterViewInit } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';
import { Observable } from 'rxjs';
import { FirestoreService } from '../services/firestore.service';
import { Campaign } from '../services/firestore.service';
declare var google: any;

@IonicPage()
@Component({
  selector: 'page-visualizar-campanhas',
  templateUrl: 'visualizar-campanhas.html',
})

export class VisualizarCampanhasPage implements AfterViewInit {
  // (...)

  showMap(element, address) {
    const position = new google.maps.LatLng(-23.5505, -46.6333);

    const options = {
      center: position,
      zoom: 13,
      mapTypeId: 'roadmap'
    };

    const map = new google.maps.Map(element, options);

    var geocoder = new google.maps.Geocoder();

    this.geocodeAddress(geocoder, map, address);
  }

  geocodeAddress(geocoder, resultsMap, address) {
    geocoder.geocode({'address': address}, function(results, status)
    {

      var icon = {
        url: "../assets/imgs/myblood-marker.png",
        scaledSize: new google.maps.Size(27,40),
      };

      if (status === 'OK') {
        resultsMap.setCenter(results[0].geometry.location);
        /* tslint:disable:no-unused-variable */
        var marker = new google.maps.Marker({
```

Fonte: Elaborado pelo autor (2019).

Figura 14 - Maps JavaScript API e Geocoding API parte 2

```

        map: resultsMap,
        position: results[0].geometry.location,
        icon: icon
    });
} else {
    console.log('Geocode was not successful for the following reason: ' + status);
}
});
}
}

```

Fonte: Elaborado pelo autor (2019).

4.2.5 API do Dialogflow

A API do Dialogflow foi inserida para interação com os usuários que possuem dúvidas remanescentes sobre doação de sangue fornecendo “[...] meios de interagir com seu produto concebendo interfaces de conversação embasada em voz e texto, como aplicativos de voz e *chatbots*, com tecnologia AI” (Dialogflow, 2019).

As figuras 15 e 16 demonstram parte do componente onde estão alocados os códigos que fazem o consumo da API do Dialogflow. O *token* da API está localizado em arquivo separado por motivos de segurança:

Figura 15 - API do Dialogflow

```

import { Component, ViewChild } from '@angular/core';
import { IonicPage, Content, Platform } from 'ionic-angular';
import { Message } from '../models/message';
import { ApiAiClient } from 'api-ai-javascript';
import { FormControl, FormBuilder } from '@angular/forms';
import { credentials } from '../../app/config';

@IonicPage()
@Component({
  selector: 'page-guia-informativo',
  templateUrl: 'guia-informativo.html',
})
export class GuiaInformativoPage {

  @ViewChild(Content) content: Content;
  protected accessToken: string = credentials.dialogflow.accessToken;
  protected messages: Message[] = [];
  protected messageForm: any;
  protected chatBox: any;

```

Fonte: Elaborado pelo autor (2019).

Figura 16 - API do Dialogflow parte 2

```

protected loading: boolean;
protected user;

constructor(public platform: Platform, public formBuilder: FormBuilder) {
  this.chatBox = '';

  this.messageForm = formBuilder.group({
    message: new FormControl('')
  });

  this.user = new ApiAiClient({
    accessToken: this.accessToken
  });
}

public sendMessage(req: string) {
  if (!req || req === '') {
    return;
  }
  this.messages.push({ from: 'Eu', text: req });
  this.loading = true;

  this.user
    .textRequest(req)
    .then(response => {

      this.messages.push({
        from: 'Sanguinho',
        text: response.result.fulfillment.speech
      });
      this.scrollToBottom();
      this.loading = false;
    })
    .catch(error => {
      console.log('Erro foi diagnosticado');
    });

  this.chatBox = '';
}

public scrollToBottom() {
  setTimeout(() => {
    this.content.scrollToBottom();
  }, 100);
}

```

Fonte: Elaborado pelo autor (2019).

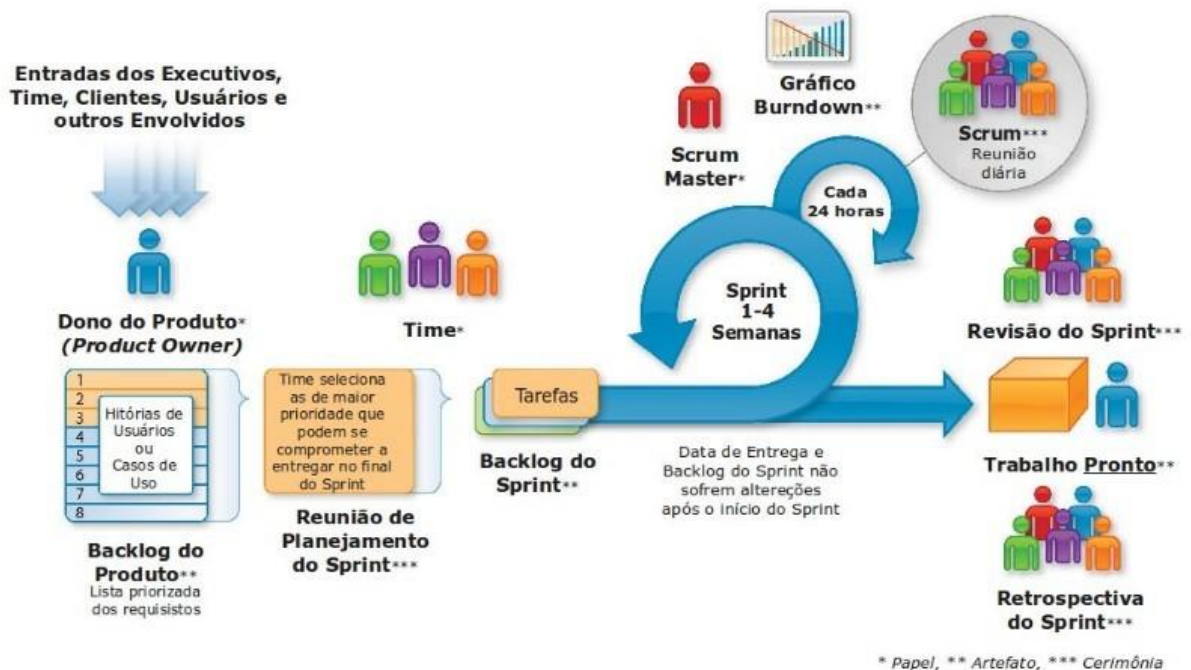
4.3 Metodologia Ágil

A metodologia ágil é um modelo e uma filosofia que propõe alternativas à gerenciamnto de projetos tradicional e tem a função de aprimorar o processo de desenvolvimento de um produto ou serviço. O objetivo final é fazer entregas com rapidez e com maior frequência, conforme surgem as necessidades do cliente.

4.3.1 Metodologia ágil Scrum

O Scrum é um *framework* para organizar e gerenciar trabalhos complexos, tal como projetos de desenvolvimento de software, uma equipe Scrum é composta basicamente por três papéis.

Figura 17 - Processo de desenvolvimento utilizando Scrum



Fonte: iMasters (2019).

- **Product Owner.**

É o ponto central com poderes de liderança sobre o produto. Ele é o único responsável por decidir quais recursos e funcionalidades serão construídos e qual a ordem que devem ser feitos.

É responsabilidade dele manter e comunicar a todos os outros participantes uma visão clara do que a equipe Scrum está buscando alcançar no projeto. Como tal, ele é responsável pelo sucesso global da solução.

Para garantir que a equipe construa rapidamente o que o Product Owner precisa, ele deve colaborar ativamente com o Scrum Master e equipe de desenvolvimento como também, estar disponível para responder às perguntas.

- **Scrum Master.**

Deve ajudar a equipe a resolver problemas e fazer melhorias no uso do Scrum. Também é responsável por proteger a equipe contra interferências externas e assume um papel de liderança na remoção de impedimentos que podem atrapalhar a produtividade.

- **Time Scrum.**

No Scrum é definido o papel do Time de Desenvolvimento, que é simplesmente a junção de todas essas pessoas em uma equipe multidisciplinar, e que são responsáveis pela concepção, construção e testes do produto.

A ideia principal é que a equipe de desenvolvimento se organize para determinar a melhor maneira de realizar o trabalho para atingir a meta estabelecida pelo Product Owner.

4.4 Etapas de desenvolvimento

Durante as etapas de desenvolvimento a metodologia ágil Scrum foi aplicada para potencializar o trabalho da equipe e controlar a evolução do produto de forma eficaz, atendendo a qualidade esperada e aos prazos com um período máximo de 21 dias por etapa. De uma forma breve, as etapas foram:

- **Primeira entrega:** abertura do desenvolvimento (criação de página de login e cadastro, integração com banco de dados);
- **Segunda entrega:** efetuação de pesquisa de campo, inicialização de novas páginas (publicar campanha, visualizar campanha, minha conta) e *back-end* das mesmas;
- **Terceira entrega:** validação de formulário, filtro visualizar campanhas disponíveis, *splash screen* e atualizações nas páginas e *back-end* da entrega 2;

- **Quarta entrega:** levantamento de informações para o *chatbot* e criação de *chatbot*, diagramas de caso de uso, diagrama de classe, dicionário de dados, DER, nova página (minhas doações);
- **Quinta entrega:** aperfeiçoar a experiência do usuário no *chatbot* e a sua interface, aperfeiçoar o cálculo de dias para doar no “Minhas Doações”, implementar o “Esqueci minha senha” e atualização dos diagramas, DER e JSON.
- **Sexta entrega:** levantamento de *bugs* existentes, desenvolvimento do mapa para visualizar as campanhas, implementar o esqueci minha senha e verificar as atividades do CQ no Trello.
- **Sétima entrega:** melhorias necessárias para futura publicação do aplicativo, implementações para melhorar usabilidade do usuário.

4.4.1 Primeira entrega

Na data 25 de fevereiro de 2019 inicializaram-se as atividades propostas pelos membros. As atividades foram estabelecidas e foram definidos níveis de complexidade ou esforço necessário para concluir cada uma, representado por pontos e quantidade de dias necessários. De forma genérica as atividades estavam relacionadas a criação de páginas iniciais e integração com o Firebase. A tabela 26 apresenta especificadamente as atividades propostas com dias alocados e pontos definidos.

Tabela 26 - Primeira etapa de desenvolvimento

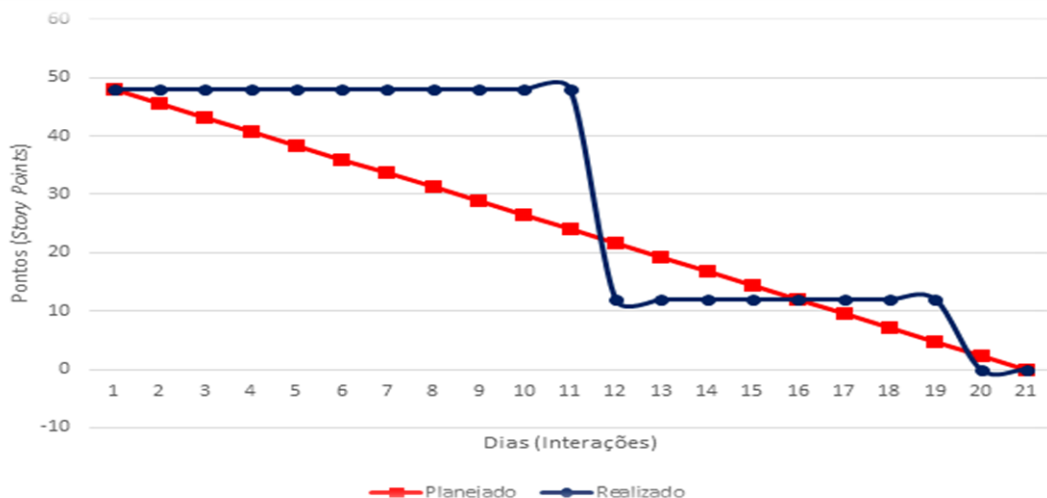
Atividades	Tempo (em dias)	Pontos
Criação de página de Cadastro do usuário	1	6
<i>Front-end</i> da página de cadastro	1	6
Criação do objeto Usuário no banco de dados	1	12
Integração do banco de dados com o aplicativo	1	12

Implementar o cadastro de usuário	1	12
Total	5	48

Fonte: Elaborado pelo autor (2019).

Os primeiros dias dessa *sprint* foram trabalhados para adquirir conhecimento sobre as ferramentas que seriam utilizadas, o que resultou em um retardamento no começo do desenvolvimento, mas posteriormente a redução de pontos ocorreu dentro dos conformes. Na figura 18 é apresentado o *burndown* da primeira etapa de desenvolvimento e a pontuação atingida.

Figura 18 - Gráfico de *Burndown* da primeira etapa de desenvolvimento.



Fonte: Elaborado pelo autor (2019).

Após a finalização da primeira *sprint*, houve uma reunião para localizar eminentes pontos positivos, ações de melhorias e erros cometidos. Os pontos levantados foram:

- O que deu certo: todas as tarefas que propostas foram feitas.
- O que deu errado: nenhum imprevisto, além ocorrer problemas técnicos que já eram esperados.
- Ações de melhorias: definiu-se que os membros precisam estudar as ferramentas de desenvolvimento utilizadas, o grupo precisa definir melhor os responsáveis por cada atividade e definir prazos para entregas individuais.

- Observações: ocorreram diversos problemas técnicos durante o desenvolvimento, porém foram solucionados após a realização de pesquisas na internet. Os problemas técnicos eram esperados pelo grupo, pois nenhum dos integrantes possuía experiência com as tecnologias trabalhadas.

4.4.2 Segunda entrega

No dia 18 de março de 2019 deu-se início às atividades da segunda etapa propostas pelos membros. As atividades foram definidas e o nível de dificuldade foi representado por pontos e quantidade de dias designados. Em geral, as atividades estavam relacionadas a criação de páginas e *back-end* do aplicativo e pesquisa de campo sobre hemocentros. A tabela 27 apresenta especificadamente as atividades propostas com dias e pontos concedidos.

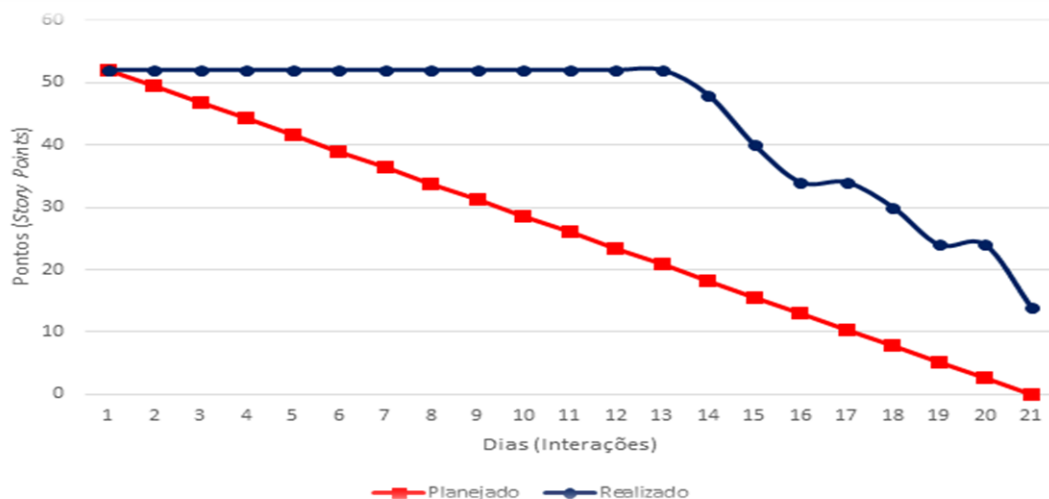
Tabela 27 - Segunda etapa de desenvolvimento

Atividade	Tempo (em dias)	Pontos
Menu lateral no aplicativo	1	4
Design do publicar campanha	1	4
<i>Front-end</i> do publicar campanha	1	4
<i>Back-end</i> publicar campanha	2	6
Design do visualizar campanha	1	4
<i>Front-end</i> do visualizar campanha	1	4
<i>Back-end</i> visualizar campanha	2	6
<i>Logoff</i>	1	4
Doar sangue (pesquisa de campo)	1	6
<i>Front-end</i> da minha conta	1	4
<i>Back-end</i> da minha conta	1	6
Total	13	52

Fonte: Elaborado pelo autor (2019).

Durante os 21 dias, houve um processo contínuo nas pesquisas para aquisição de novos conhecimentos sobre as ferramentas utilizadas, tendo atrasos no início do desenvolvimento, realizando redução de pontos conforme as atividades eram concluídas. O *burndown* dessa etapa com a pontuação atingida é demonstrado na figura 19.

Figura 19 - Gráfico de *Burndown* da segunda etapa de desenvolvimento.



Fonte: Elaborado pelo autor (2019).

Após a finalização da segunda etapa houve uma reunião no dia 07 de abril de 2019 para localizar eminentes pontos positivos, ações de melhorias e erros cometidos. Os pontos levantados foram:

- O que deu certo: *front-end* do publicar campanha e minha conta realizados.
- O que deu errado: ocorreu imprevisto no *back-end* do publicar campanha necessitando a implantação de uma nova página (minha conta).
- Ações de melhorias: os membros precisam estudar as ferramentas de desenvolvimento utilizadas, aproveitar melhor os 21 dias.
- Observações: foi realizada uma visita ao hemocentro de Americana e coletadas informações sobre a padronização adotada pelas bases de dados dos hemocentros, foram obtidos contatos de profissionais especializados na área.

4.4.3 Terceira entrega

No dia 8 de abril de 2019 as atividades da terceira etapa propostas pelos membros iniciaram. Estas foram definidas por dificuldade representada por pontos e quantidade de dias gasto. As funções estavam relacionadas a continuação das atividades não concluídas da etapa anterior, novo filtro na página visualizar campanha, *splash screen* e estudo para realizar *build* para plataformas (iOS, Android). A tabela 28 apresenta especificadamente as tarefas propostas com dias gastos e pontos cedidos.

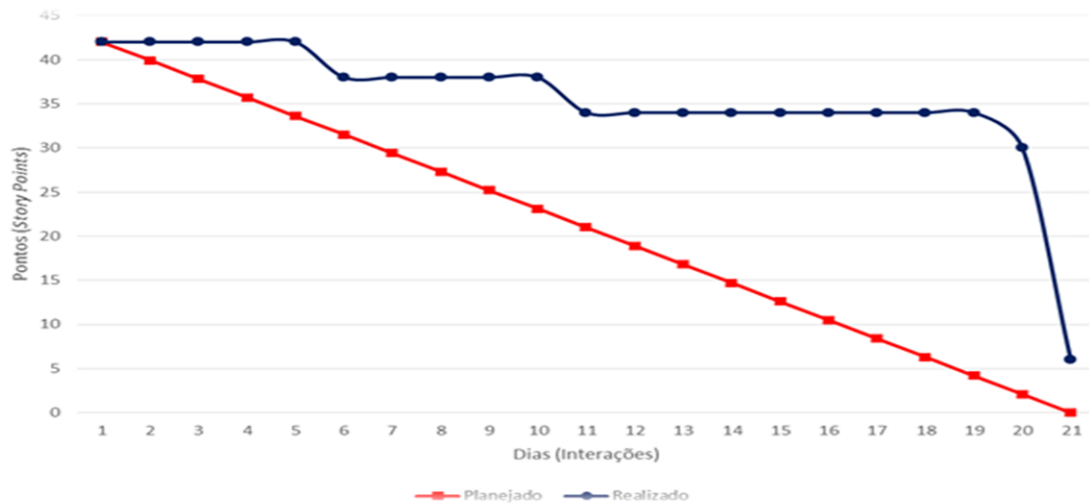
Tabela 28 - Terceira etapa de desenvolvimento

Atividades	Tempo (em dias)	Pontos
<i>Back-end</i> do publicar campanha	2	6
Design do visualizar campanha	1	4
<i>Front-end</i> do visualizar campanha	1	6
Validação de formulário publicar campanha	3	6
Filtro de visualizar campanha (região, cidade)	3	6
<i>Splash screen</i>	1	4
Alteração no <i>front-end</i> do publicar campanha	0,5	4
<i>Build</i> para plataformas (iOS, Android)	1	6
Total	12,5	42

Fonte: Elaborado pelo autor (2019).

Ao iniciar os 21 dias, o foco foi finalizar o *back-end* do publicar campanhas e visualizar campanhas, ambos os pontos cruciais da aplicação, realizando redução de pontos conforme as atividades eram concluídas. O *burndown* da terceira etapa pode ser visualizado na figura 20.

Figura 20 - Gráfico de *Burndown* da terceira etapa de desenvolvimento.



Fonte: Elaborado pelo autor (2019).

Após a finalização da terceira etapa houve uma reunião no dia 28 de abril de 2019 para analisar pontos positivos, ações de melhorias e erros cometidos. Os pontos levantados foram:

- O que deu certo: sucesso ao desenvolver os pontos principais (publicar campanha e visualizar campanha) e a comunicação da equipe melhorou.
- O que deu errado: não foi finalizado a validação de todos os campos e não foi possível gerar o aplicativo no *iOS*.
- Ações de melhorias: maior dedicação do grupo para bom aproveitamento das tecnologias e melhorar ainda mais o tempo de resposta na comunicação do grupo.
- Observações: o desenvolvimento fluiu bem, mesmo com as dificuldades.

4.4.4 Quarta entrega

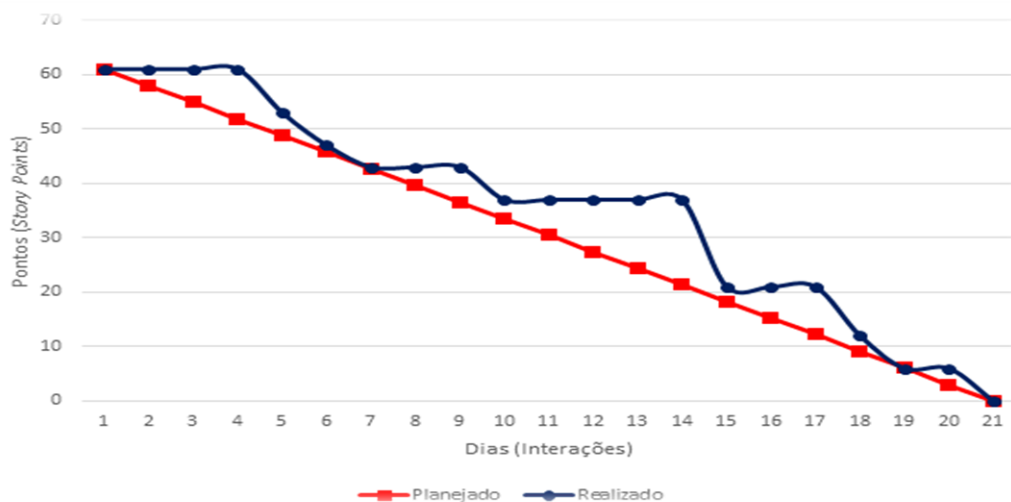
Na data 29 de abril de 2019 começaram as atividades da quarta etapa propostas pelo grupo. As tarefas foram definidas por dificuldade representada por pontos e quantidade de dias necessários. As tarefas estavam relacionadas ao desenvolvimento de um *chatbot*, documentação e criação de nova página (minhas doações). A tabela 29 apresenta especificadamente as funções propostas com dias gastos e pontos designados.

Tabela 29 - Quarta etapa de desenvolvimento

Atividades	Tempo (em dias)	Pontos
Levantamento de informações para o guia e <i>chatbot</i>	1	6
Diagrama de caso de uso	1	4
Diagrama de classe	1	4
Dicionário de dados	1	4
Criação do <i>chatbot</i> (<i>front-end</i> , <i>back-end</i>)	6	16
Design minhas doações	1	6
Documento geral	6	5
<i>Front-end</i> (minhas doações)	1	6
<i>Back-end</i> (minhas doações)	1	6
DER	1	4
Total	20	61

Fonte: Elaborado pelo autor (2019).

Ao iniciar os 21 dias, o foco foi desenvolver o *chatbot* para a aplicação. Por conta da criação dos diagramas, houve progresso na documentação. Na figura 21 é apresentado o *burndown* da quarta etapa e a pontuação atingida.

Figura 21 - Gráfico de *Burndown* da quarta etapa de desenvolvimento.

Fonte: Elaborado pelo autor (2019).

Na data 20 de maio de 2019 foi realizada uma reunião para levantar os pontos positivos, negativos e ações de melhorias da quarta *sprint*. No levantamento foi constatado que:

- O que deu certo: todas as atividades propostas foram concluídas com sucesso e não houve incidentes nessa etapa.
- O que deu errado: nada.
- Ações de melhorias: distribuir melhor as atividades entre os membros e estudar continuamente as ferramentas utilizadas.

4.4.5 Quinta entrega

Na data 28 de maio de 2019 foram iniciadas as atividades da quinta entrega proposta pelo grupo. As tarefas foram definidas pelo grau de dificuldade. As atividades consistiam em melhorar a documentação do projeto e remoção de *bugs*. Na tabela 30 é demonstrado as especificações das atividades propostas com dias utilizados e pontos concedidos.

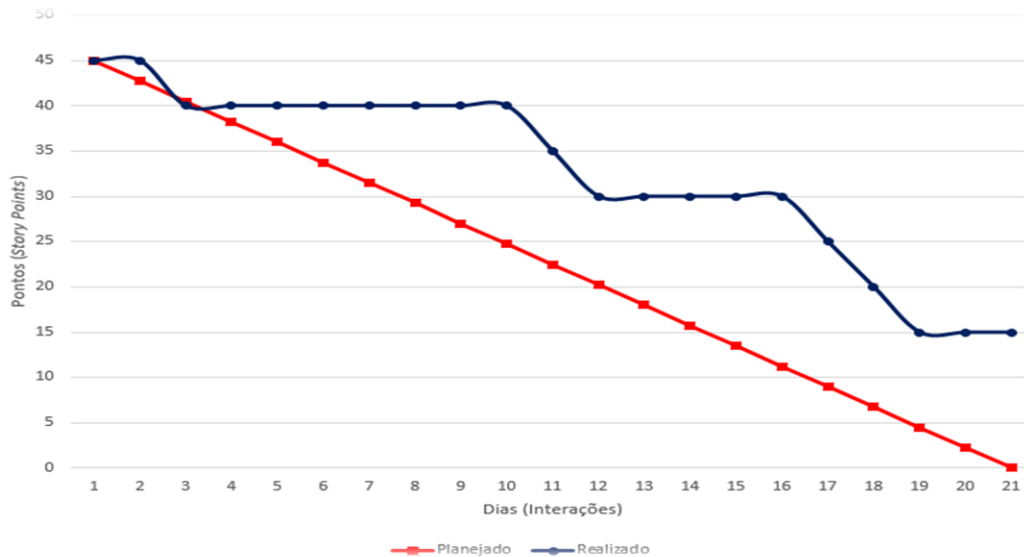
Tabela 30 - Quinta etapa de desenvolvimento

Atividades	Tempo (em dias)	Pontos
Aperfeiçoar da experiência do usuário no <i>chatbot</i>	1	5
Aperfeiçoar interface do <i>chatbot</i>	1	5
Aperfeiçoar o cálculo de dias para doar em “Minhas doações”	1	5
Implementar o “Esqueci minha senha”	1	5
Atualizar os diagramas	1	5
Atualizar DER e JSON	1	5
Documentação geral	1	5
Diagrama de sequência	1	5

Fonte: Elaborado pelo autor (2019).

No início dos 22 dias o foco foi em melhorar a documentação e desenvolver continuamente a aplicação, realizando redução de pontos conforme as atividades fossem finalizadas. O *burndown* da quinta etapa é apresentado na figura 21.

Figura 22 - Gráfico de *Burndown* da quinta etapa de desenvolvimento.



Fonte: Elaborado pelo autor (2019).

No dia 19 de junho de 2019 foi realizada a reunião para levantamento de pontos positivos e negativos da *sprint* realizada. Os pontos levantados foram:

- O que deu certo: atividades com maiores importâncias foram concluídas.
- O que deu errado: devido à falta de conhecimento das linguagens e tecnologias utilizadas, foi gasto tempo em excesso, pesquisando e assistindo aulas e não houve sucesso no desenvolvimento do aplicativo.
- Ações de melhorias: aperfeiçoar conhecimento relacionado a programação.

4.4.6 Sexta entrega

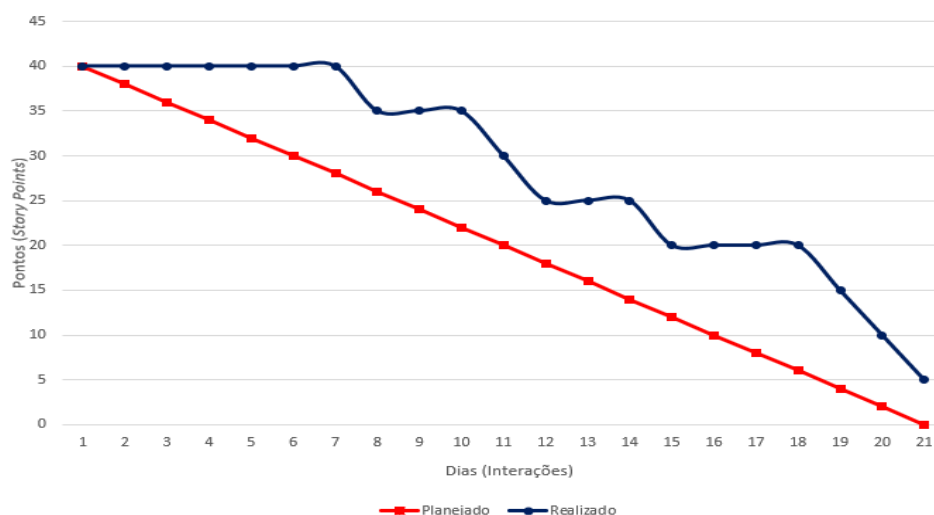
No dia 14 de setembro de 2019 foi iniciado a sexta etapa de desenvolvimento. As atividades consistiam em fazer um levantamento dos *bugs* existentes na aplicação e separá-los por prioridade, desenvolver o mapa da aplicação, gerar o *Android application package* (APK) de uma nova versão para realizar os testes, implementar o esqueci minha senha, verificar as atividades que estavam no CQ do Trello, começar os slides para apresentação e desenvolver um vídeo demonstrativo do aplicativo. A tabela 31 mostra a sexta etapa de desenvolvimento.

Tabela 31 - Sexta etapa de desenvolvimento

Atividades	Tempo (em dias)	Pontos
Levantamento de <i>bugs</i>	3	5
Desenvolvimento do mapa	3	5
Realizar o <i>build</i> da nova versão do aplicativo	1	5
Esqueci minha senha	3	5
Verificar os CQ's do Trello	3	5
Incluir <i>sprint</i> 5 no documento	3	5
Desenvolvimento de um vídeo com áudio	1	5
Começar slide TCC	1	5

Fonte: Elaborado pelo autor (2019).

No início dos 21 dias foram iniciadas as atividades propostas, focando em corrigir os *bugs* levantados e desenvolver o mapa de visualização das campanhas publicadas que era o foco principal. A figura 23 ilustra o progresso das atividades realizadas na sexta etapa de desenvolvimento.

Figura 23 - Gráfico de *Burndown* da sexta etapa de desenvolvimento

Fonte: Elaborado pelo autor (2019).

No dia 6 de outubro de 2019 foi realizada uma reunião para levantamento de pontos positivos e negativos da *sprint* realizada, os pontos levantados foram:

- O que deu certo: as atividades indispensáveis foram realizadas com sucesso.
- O que deu errado: falta de gestão do próprio tempo para realizar as atividades e dificuldade para reunir a equipe.
 - Ações de melhorias: melhorar a divisão de atividades por membro.

4.4.7 Sétima entrega

No dia 29 de outubro de 2019 foi realizada a abertura de uma nova entrega de desenvolvimento. As tarefas pretendiam melhorar o aplicativo implementando funcionalidades necessárias para a publicação do aplicativo, como novo filtro para as campanhas de doações, inclusão do preenchimento automático de endereço através do CEP, inserção dos tipos sanguíneos restantes e outras correções. As atividades desta etapa são demonstradas na tabela 32.

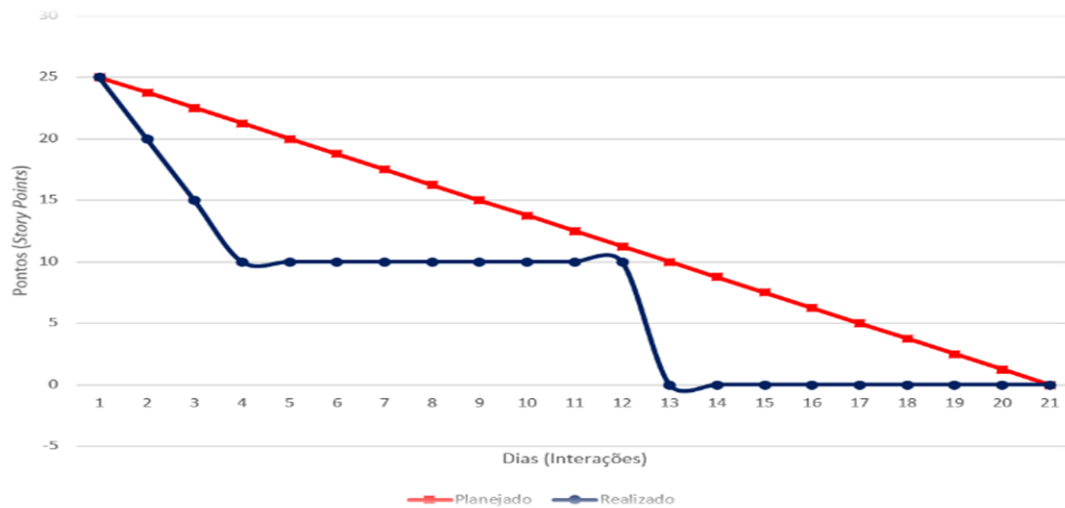
Tabela 32 - Sétima etapa de desenvolvimento

Atividades	Tempo (em dias)	Pontos
Colocar campo gênero	1	5
Inserir busca por CEP	5	10
Adicionar tipos de sangue	3	5
Adicionar filtro de sangue	3	5

Fonte: Elaborado pelo autor (2019).

As tarefas que haviam sido acordadas pelo grupo foram iniciadas, seguindo os 21 dias. A sétima entrega foi voltada em funcionalidades para a futura publicação do aplicativo. A figura 24 exibe o progresso das atividades.

Figura 24 - Gráfico de *Burndown* da sétima etapa de desenvolvimento



Fonte: Elaborado pelo autor (2019).

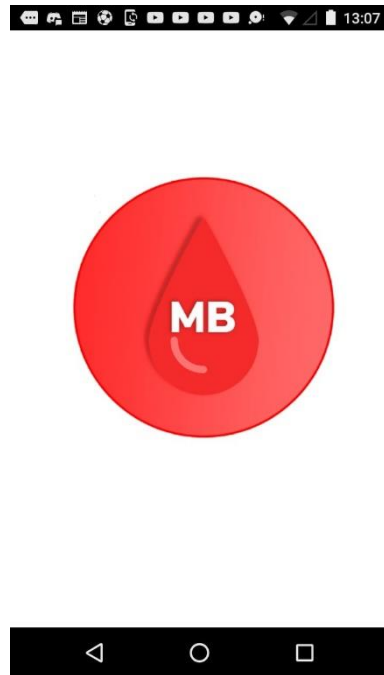
Dia 19 de novembro ocorreu uma reunião para o levantamento de pontos positivos e negativos da sétima *sprint* realizada, os pontos levantados foram:

- O que deu certo: foram realizadas as tarefas com facilidade não possuindo nenhum imprevisto, foi demonstrado o aplicativo para a comunidade e recolhido *feedback*.
- O que deu errado: nenhum imprevisto.
- Ações de melhorias: manter as *sprints* equilibradas para permanecer tempo e produção alinhadas.

4.5 Interfaces de Usuário

A primeira tela exibida assim que o aplicativo é aberto é a *splash screen*, que atualmente consiste em uma tela com a logotipo do aplicativo que carrega por alguns segundos enquanto ocorre a inicialização, como demonstrado na figura 25.

Figura 25 - Tela de *splash screen*



Fonte: Elaborado pelo autor (2019).

A figura 26 ilustra a interface da tela de cadastro, o usuário ao realizar esse registro terá acesso as demais telas do aplicativo, é necessário inserir um e-mail com caracteres válidos (uso da arroba e do .com) e criar uma senha de no mínimo 6 caracteres, o botão só será habilitado caso as validações exigidas pelos campos sejam cumpridas.

Figura 26 - Tela de cadastro

Fonte: Elaborado pelo autor (2019).

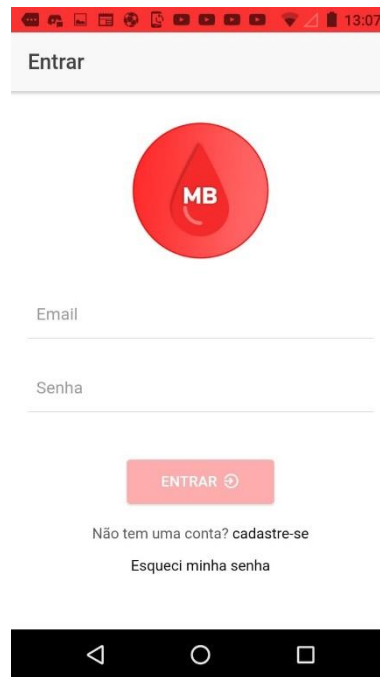
Como demonstrado na figura 27, ao clicar no botão cadastrar é exibido um alerta perguntando se o usuário deseja terminar de preencher os dados, tela que será demonstrada nas figuras 45, 46 e 47 ou se deseja navegar pelo aplicativo.

Figura 27 - Tela de cadastro com botões do alerta

Fonte: Elaborado pelo autor (2019).

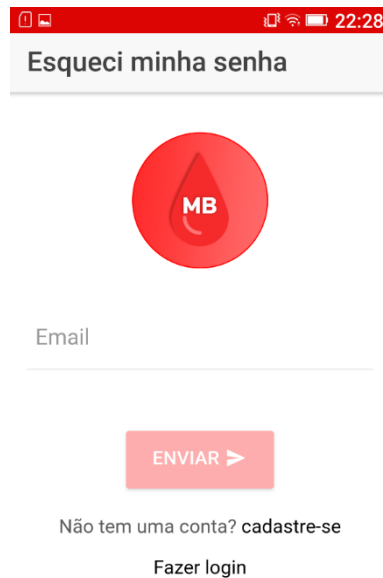
Conforme exibido na figura 28, na tela de login o usuário digita o e-mail e senha de no mínimo 6 dígitos fornecidos na tela de cadastro para fazer a autenticação e navegar entre as telas do aplicativo, caso o e-mail ou senha não atenda as exigências o botão de entrar permanece desabilitado para ser clicado.

Figura 28 - Tela de login



Fonte: Elaborado pelo autor (2019).

Essa tela permite que o usuário envie um link de redefinição de senha para o seu e-mail caso esqueça a senha de acesso e ainda se lembre do e-mail que utilizava para acessar o aplicativo, após redefinir a senha o usuário voltará a acessar o aplicativo normalmente com o e-mail que utilizava antes do esquecimento. Conforme mostrado na figura 29.

Figura 29 - Esqueci minha senha

Esqueci minha senha

MB

Email

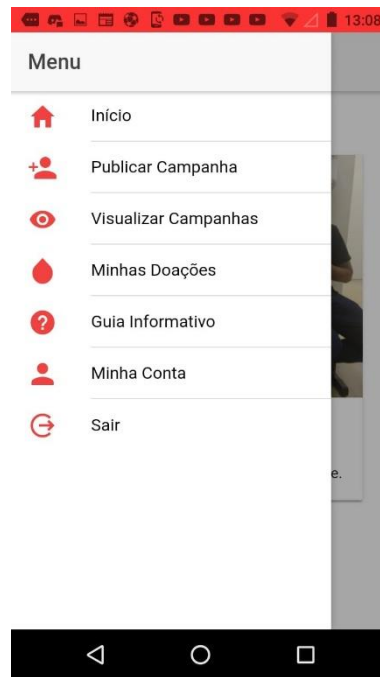
ENVIAR >

Não tem uma conta? cadastre-se

Fazer login

Fonte: Elaborado pelo autor (2019).

O menu lateral de navegação tem por objetivo deixar a navegação entre as telas mais fácil para o usuário, garantindo que ele se localize mais facilmente no aplicativo e deixando mais interativo. A figura 30 mostra a exibição desse menu que pode ser aberto ao segurar no canto esquerdo da tela e arrastar para o canto direito, ou clicando no ícone localizado no canto superior esquerdo.

Figura 30 - Menu lateral

Fonte: Elaborado pelo autor (2019).

Tela de início, pretende-se exibir as fotos dos doadores voluntários, após realizar uma doação ao entrar no aplicativo será capaz de fazer upload de uma foto e publicar na página inicial do aplicativo, mostrando que todo o processo é indolor e incentivando outras pessoas a doar. A figura 31 ilustra o estado atual da tela inicial do aplicativo.

Figura 31 - Tela de início

Fonte: Elaborado pelo autor (2019).

Tela de publicar campanha é a tela principal do aplicativo, é onde o usuário preenche os dados relativos à campanha e clica no botão para publicar, para que assim outros usuários possam visualizar as campanhas e colaborar com elas. Observada essa tela nas figuras 32, 33, 34, 35 e 36.

O usuário elege que tipo de doação deseja fazer uma publicação se é de sangue ou de plaquetas, seleciona o tipo de sangue e a data limite para receber doações para a campanha, conforme ilustrados na figura 32 e 33. Já na figura 34 é onde o usuário vai preencher os dados relativos ao hemocentro no qual o doador irá comparecer para realizar a doação, como, nome, CEP, endereço, número, cidade e uf do hemocentro.

Conforme exibido na figura 35 o usuário pode clicar no botão e selecionar se ele é ou não o paciente, caso não seja é possível indicar um paciente, também é possível adicionar observações caso seja necessário e finalmente publicar a campanha.

Figura 32 - Publicar campanha exibição dos dados da doação

The screenshot shows a mobile application interface for publishing a campaign. At the top, there is a status bar with the time 15:36. Below it is a navigation bar with a back arrow and the text 'Publicar Campanha'. A red circular profile picture with the initials 'MB' is displayed. The main heading is 'Campanha' with the subtitle 'Tipo de doação'. There are three radio button options: 'Sangue' (selected), 'Plaquetas', and 'Tipo de Sangue'. Below these is a 'Data limite' section with a 'Data' input field. At the bottom, the text 'Hemocentro' is visible above the Android navigation bar.

Fonte: Elaborado pelo autor (2019).

Figura 33 - Publicar campanha exibição do campo de data

This screenshot shows the same 'Publicar Campanha' form as in Figure 32, but with a date picker overlay. The date picker is positioned over the 'Data' field and shows a list of dates from 18/09/2021 to 22/12. The date '20/11/2019' is highlighted in blue. At the top of the date picker, there are 'CANCEL' and 'DONE' buttons. The background of the form is dimmed.

Fonte: Elaborado pelo autor (2019).

Figura 34 - Publicar campanha exibição dos dados do hemocentro

Publicar Campanha

Data

Hemocentro

Nome

CEP

Endereço

Numero

Cidade

UF

Contato

Mostrar menu

Eu sou o paciente

◀ ○ □

Fonte: Elaborado pelo autor (2019).

Figura 35 - Publicar campanha exibição dos dados de contato da campanha

Publicar Campanha

Cidade

UF

Contato

Mostrar menu

Eu sou o paciente

Sim

Não

Observações

PUBLICAR ❤️

◀ ○ □

Fonte: Elaborado pelo autor (2019).

Figura 36 - Publicar campanha exibição da validação de campos

Publicar Campanha

Hemocentro

Nome
Digite o nome até 15 caracteres
13400370

Rua XV de Novembro

Numero

Piracicaba

SP

Contato

Mostrar menu

E-mail

Fonte: Elaborado pelo autor (2019).

Tela de visualizar campanha permite que o usuário observe as campanhas que foram publicadas por outros usuários e o mapa indicando o local do hemocentro, também é possível filtrar as publicações por localidade e por tipo sanguíneo.

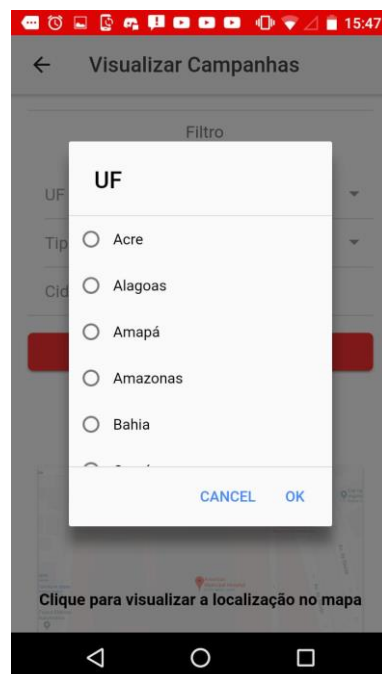
As figuras 37 a 44 ilustram a tela de visualizar campanhas com os campos de filtro e o mapa.

Conforme mostrado na figura 37, o usuário pode escolher o tipo de filtro, se deseja filtrar campanhas por UF, por tipo sanguíneo, pela cidade ou usar ambos os filtros, tudo para facilitar que ele possa encontrar um local próximo e com sangue compatível, para que então o auxílio seja realizado.

Figura 37 - Visualizar campanhas tipos de filtro

Fonte: Elaborado pelo autor (2019).

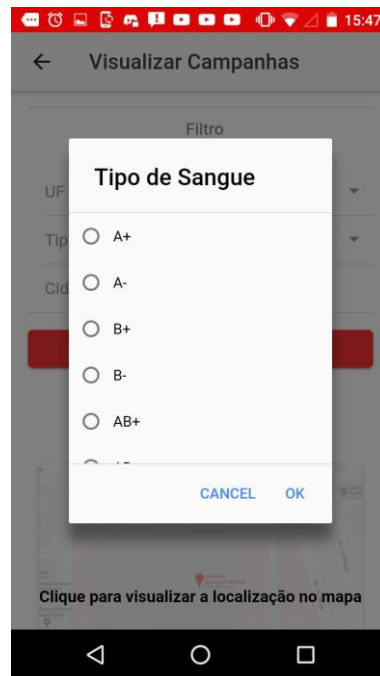
Na figura 38, pode-se ver o comportamento da tela ao ser indicado o filtro por UF, é exibido para o usuário os 26 estados do Brasil sendo selecionado o estado no qual quer pesquisar as campanhas publicadas.

Figura 38 - Visualizar campanhas filtro por UF

Fonte: Elaborado pelo autor (2019).

Na figura 39 pode-se ver o comportamento da tela ao ser escolhido o filtro por tipo sanguíneo, o usuário seleciona o seu tipo de sangue e será exibido as campanhas com esse tipo de sangue específico.

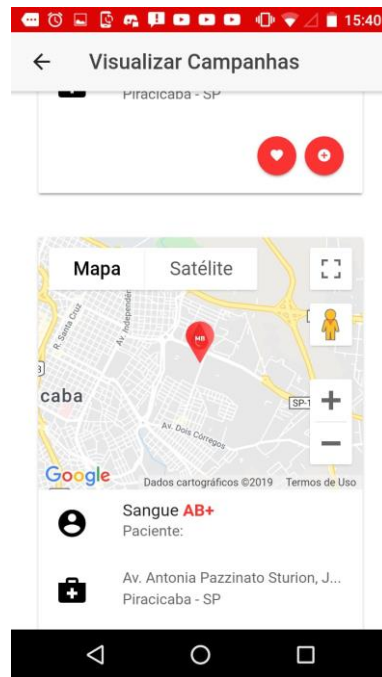
Figura 39 - Visualizar campanhas filtro por tipo de sangue



Fonte: Elaborado pelo autor (2019).

Na tela de visualizar campanhas, ao arrastar para baixo são exibidas as publicações, conforme mostrado nas figuras 40 e 41 visualizadas como mapa, e na figura 42 com a foto do local.

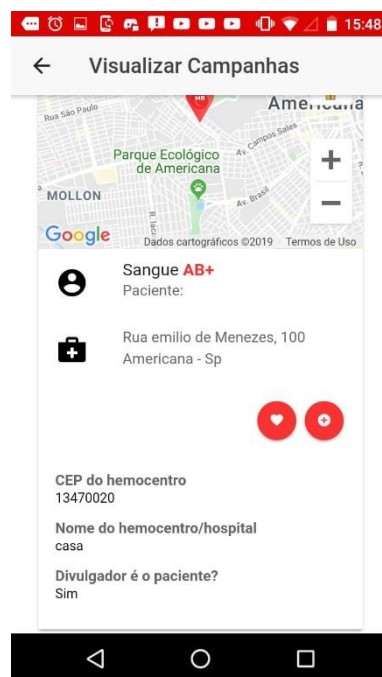
Figura 40 - Visualizar campanhas exibição do mapa



Fonte: Elaborado pelo autor (2019).

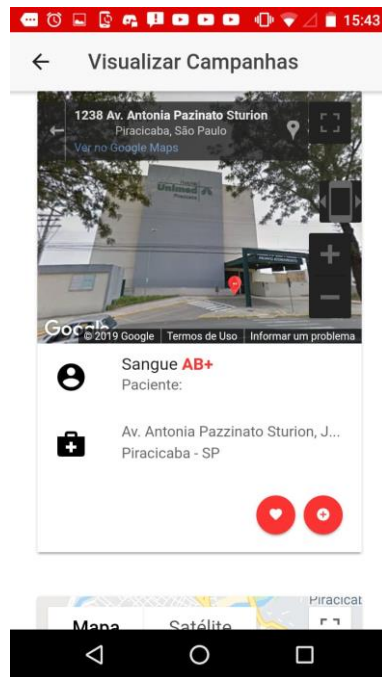
Ao clicar no botão localizado no canto inferior direito de cada campanha, é expandido para baixo as informações da campanha, conforme ilustrado na figura 41.

Figura 41 - Visualizar campanhas exibição do mapa com os dados da campanha



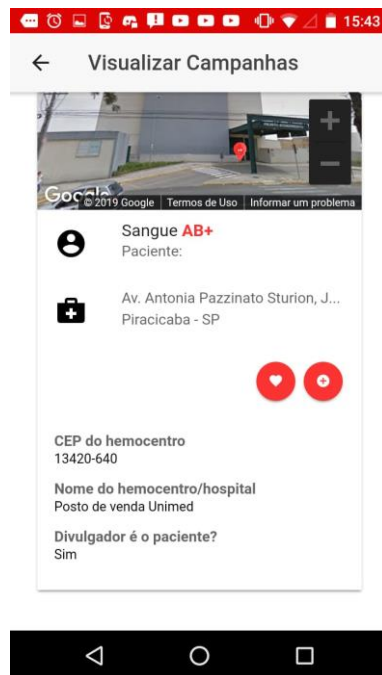
Fonte: Elaborado pelo autor (2019).

Figura 42 - Visualizar campanhas exibição da foto com a localidade do hemocentro publicado



Fonte: Elaborado pelo autor (2019).

Figura 43 - Visualizar campanhas exibição da foto com a localidade do hemocentro publicado e dados da campanha



Fonte: Elaborado pelo autor (2019).

Quando clicado sobre o mapa, é possível expandir o local do hemocentro, conforme ilustrado na figura 44, uma publicação de campanha de doação de sangue no hospital Unimed de Piracicaba.

Figura 44 - Visualizar campanhas visualização do mapa expandido



Fonte: Elaborado pelo autor (2019).

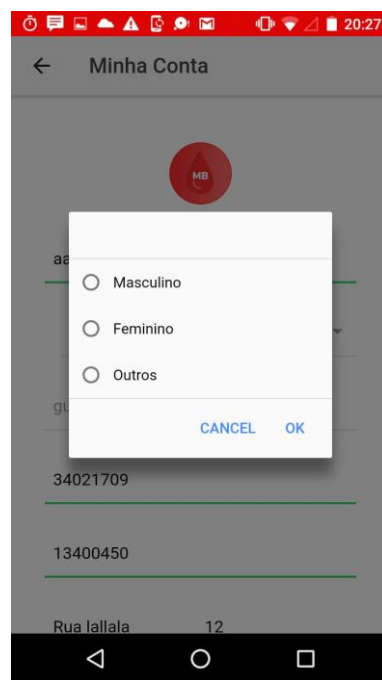
Tela de minha conta, é onde o usuário pode cadastrar alguns dados pessoais como preencher o gênero, nome, telefone, CEP, endereço e tipo sanguíneo e atualizar quando necessário.

Caso o usuário escolha a opção preencher que foi demonstrada na figura 27, ele será redirecionado para esta página, ilustrada nas figuras 45, 46 e 47.

Figura 45 - Minha conta dados pessoais do usuário

The screenshot shows a mobile application interface titled "Minha Conta". At the top, there is a status bar with the time 20:18. Below the title bar, there is a dropdown menu labeled "Escolha o gênero". The form contains several input fields: an email field with "guilberty@hotmail.com", a phone number field with "34021709", another phone number field with "13400450", a street address field with "Rua lallala" and "12", and a city/state field with "Piracicaba" and "SP". Below these fields is a section titled "Tipo de sangue" with a radio button selected for "A+". The bottom of the screen shows a navigation bar with three icons: a back arrow, a home circle, and a recent apps square.

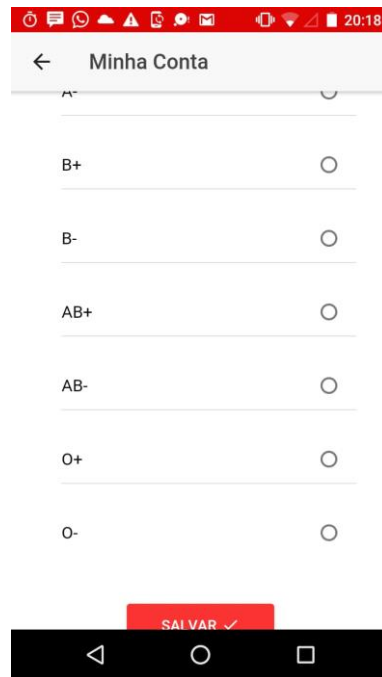
Fonte: Elaborado pelo autor (2019).

Figura 46 - Minha conta campo de gênero selecionado

The screenshot shows the same mobile application interface as Figure 45, but with a dialog box open for gender selection. The dialog box has a white background and contains three radio button options: "Masculino", "Feminino", and "Outros". At the bottom of the dialog, there are two buttons: "CANCEL" and "OK". The background of the app is dimmed, and the status bar at the top shows the time 20:27. The navigation bar at the bottom is also visible.

Fonte: Elaborado pelo autor (2019).

Figura 47 - Minha conta tipo sanguíneo do usuário



Fonte: Elaborado pelo autor (2019).

A tela minhas doações mostra todas as doações realizadas e inseridas pelo usuário e exibe a quantidade de dias necessários para que o doador possa efetuar a próxima doação, como demonstrado na figura 48, é necessário clicar no botão para adicionar uma doação realizada e então o aplicativo começa a realizar a apuração.

Figura 48 - Minhas doações

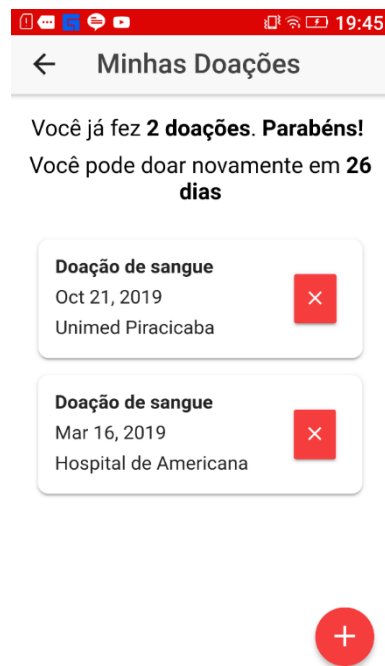


Fonte: Elaborado pelo autor (2019).

A figura 49 exibe as doações cadastradas com data e local e na parte superior é exibido quantas doações o usuário fez e determinado através do cálculo quanto tempo deve-se esperar para doar novamente, baseado no sexo do usuário.

O intervalo de doação de sangue para homens é de 60 dias e para as mulheres é de 90 dias. Levando em consideração que o número máximo de doações por ano é de 4 vezes para homens e 3 vezes para mulheres.

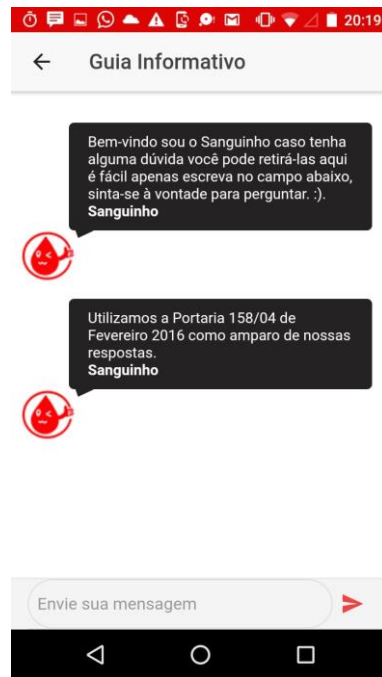
Figura 49 - Minhas doações cadastradas



Fonte: Elaborado pelo autor (2019).

A tela do guia informativo mais cativante do aplicativo, é por meio dela que o usuário pode esclarecer dúvidas conversando com o *Sanguinho* um *chatbot* implementado que consegue aprender com o usuário e de esclarecer dúvidas frequentes. Sua interface é exibida na figura 50.

Figura 50 - Guia informativo tela de boas vindas

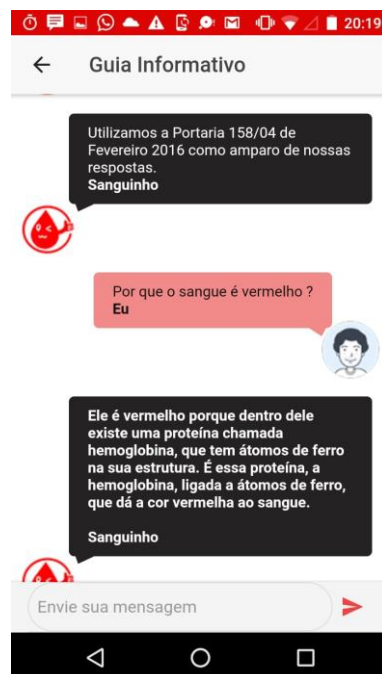


Fonte: Elaborado pelo autor (2019).

Conforme mostrado na figura 51, é demonstrado uma das 58 perguntas que atualmente o *Sanguinho* já sabe responder.

A pergunta realizada foi: “Por que o sangue é vermelho?”.

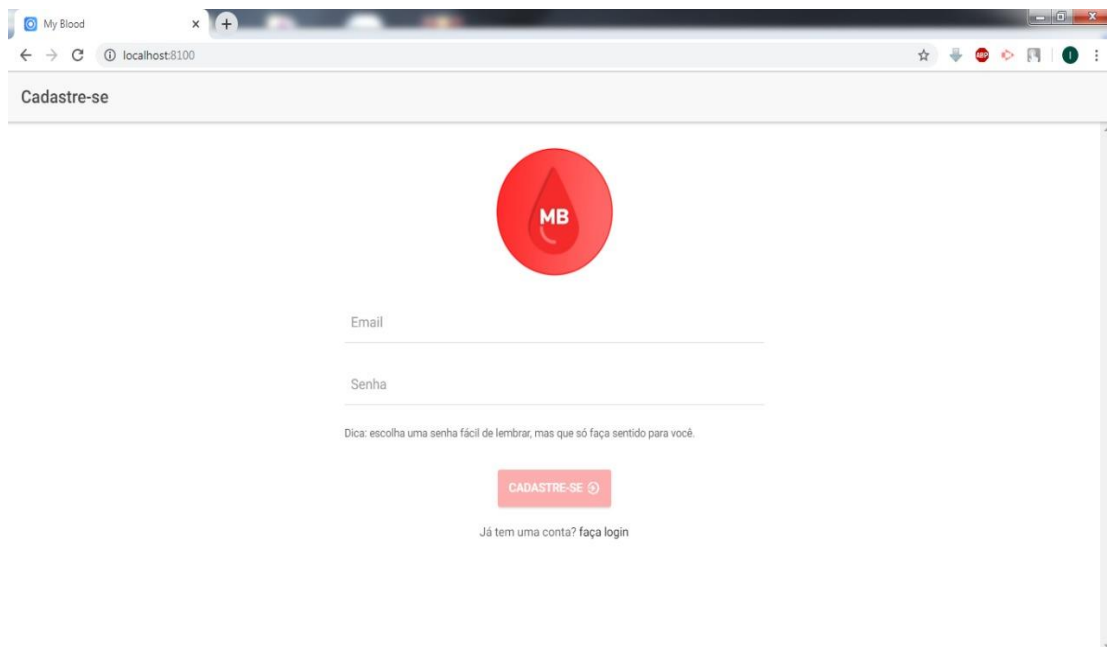
Figura 51 - Guia informativo exemplo de pergunta



Fonte: Elaborado pelo autor (2019).

As figuras 52 e 53, ilustram respectivamente, as telas de cadastro e de login na versão para web, a tela é exibida em uma resolução maior diferente da versão mobile demonstrada anteriormente.

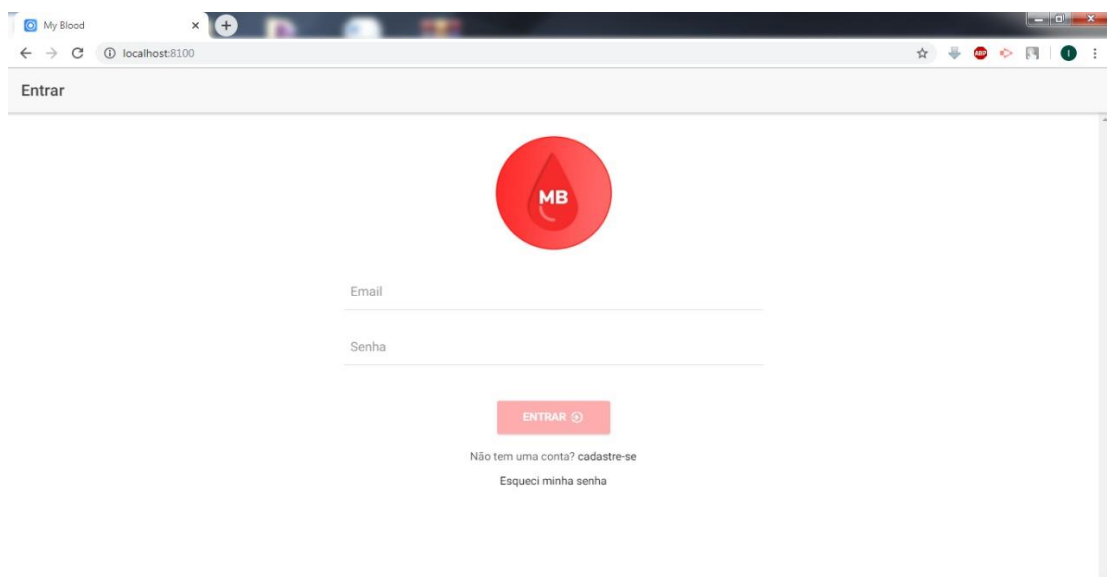
Figura 52 - Cadastro visualização na versão web



The screenshot shows a web browser window with the address bar displaying 'localhost:8100'. The page title is 'Cadastro-se'. At the top center is a red circular logo with a white blood drop and the letters 'MB'. Below the logo are two input fields labeled 'Email' and 'Senha'. A hint text reads: 'Dica: escolha uma senha fácil de lembrar, mas que só faça sentido para você.' Below the fields is a red button labeled 'CADASTRE-SE'. At the bottom, there is a link: 'Já tem uma conta? faça login'.

Fonte: Elaborado pelo autor (2019).

Figura 53 - Login visualização na versão web



The screenshot shows a web browser window with the address bar displaying 'localhost:8100'. The page title is 'Entrar'. At the top center is a red circular logo with a white blood drop and the letters 'MB'. Below the logo are two input fields labeled 'Email' and 'Senha'. Below the fields is a red button labeled 'ENTRAR'. At the bottom, there are two links: 'Não tem uma conta? cadastre-se' and 'Esqueci minha senha'.

Fonte: Elaborado pelo autor (2019).

5. CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo final o desenvolvimento de um aplicativo que por meio da própria contribuição da comunidade promove campanhas de doação de sangue, permitindo que os usuários se informem dentro do aplicativo e encontrem hemocentros próximos com pacientes que possuem sangue compatível para receber a doação.

Conforme o progresso do aplicativo, surgiram alterações em seu planejamento, tais como modificações de requisitos e complemento de funcionalidades visando o melhor funcionamento do sistema. Levando em consideração o feedback dos usuários de aplicativos semelhantes já publicados na Play Store, foi proposto desenvolver funcionalidades não existentes nos aplicativos semelhantes, de forma que agregasse valor ao projeto.

No período de desenvolvimento, houve dificuldades para adaptação das tecnologias utilizadas, mas através de reuniões frequentes foi relatado a necessidade de estudar mais, para melhor compreensão da arquitetura do projeto e características da linguagem de programação. Durante os encontros para apresentar e revisar entregas, era realizado uma discussão sobre os objetivos concluídos, as estratégias que funcionaram ou não, e o planejamento para a etapa seguinte, o que foi um processo que constantemente alinhou a equipe e beneficiou a qualidade do projeto. O desenvolvimento foi um processo de constante aprimoramento, tais como aprimoramento de requisitos, implementação de novas funcionalidades e melhoria das já existentes, alterações em interfaces, validações, entre outros.

O aplicativo possui as principais funcionalidades implementadas, entretanto ainda se encontra em processo de desenvolvimento e serão realizados ajustes e atualizações conforme necessários. É imprescindível a publicação do aplicativo na Play Store e Apple Store para avaliar como o aplicativo será recebido pelos usuários de dispositivos Android e iOS e coletar o *feedback*. Pode-se planejar novas funcionalidades analisando o *feedback* recebido, como também atualizar e dar suporte ao aplicativo de modo geral.

Como possíveis trabalhos futuros, pensou-se em:

- Implementar notificações para alertar o usuário a possibilidade de realizar outra doação;
- Permitir login com redes sociais;
- Desenvolver mural na página inicial para doadores postarem fotos pós-doação;
- Disponibilizar rota até hemocentro através do Google Maps;
- Notificar o doador sobre pacientes que necessitam de doação da mesma região e de tipo sanguíneo compatível;
- Permitir compartilhamento de campanhas de doação nas redes sociais;
- Implementar o filtro de tipos sanguíneos que são compatíveis na visualização de campanhas.

REFERÊNCIAS

ADRIANO S., Thiago. "**Ionic 3: Criando uma Modal e recebendo os seus dados em uma página**". Medium. 2018. Disponível em: <https://medium.com/@programadriano/ionic-3-criando-uma-modal-e-recebendo-os-seus-dados-em-uma-p%C3%A1gina-30ba5b80b261>. Acesso em: 31 abr. 2019.

ADOBE PHOTOSHOP. "**O Photoshop está onde a sua inspiração estiver**". Adobe. 2019. Disponível em: <https://www.adobe.com/br/products/photoshop.html>. Acesso em: 09 dez. 2019.

ADOBE XD. "**Vamos trabalhar juntos**". Adobe. 2019. Disponível em: <https://www.adobe.com/br/products/xd.html>. Acesso em: 09 dez. 2019.

ALVAREZ, Sara. "**O que é um dicionário de dados?**". Criarweb 2019. Disponível em: <http://www.criarweb.dominiotemporario.com/faq/o-que-e-um-dicionario-dados.html>. Acesso em: 26 nov. 2019.

AMPUDIA, Ricardo. "**Celular é mais utilizado do que computador para acessar internet no Brasil**". Folha de São Paulo. 2018. Disponível em: <https://www1.folha.uol.com.br/tec/2018/07/celular-e-mais-utilizado-do-que-computador-para-acessar-internet-no-brasil.shtml>. Acesso em: 13 abr. 2019.

ANDRADE, Ana Paula de. "**O que é Ionic?**". TreinaWeb Tecnologia LTDA 2004 – 2019. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-ionic/>. Acesso em: 26 nov. 2019.

ANDROID STUDIO. "**Conheça o Android Studio**". Google Developers. 2019. Disponível em: <https://developer.android.com/studio/intro?hl=pt-br>. Acesso em: 26 nov. 2019.

ANGULAR. "**Architecture overview**". Angular. 2019. Disponível em: <https://angular.io/guide/architecture>. Acesso em: 24 nov. 2019.

ANGULAR. "**Introduction to modules**". Angular. 2019. Disponível em: <https://angular.io/guide/architecture-modules>. Acesso em: 24 nov. 2019.

APACHE CORDOVA. "**Architectural overview of Cordova platform - Apache Cordova**". Apache Cordova. 2019. Disponível em: <https://cordova.apache.org/docs/en/latest/guide/overview/index.html>. Acesso em: 24 nov. 2019.

APPSEED. "**How To Authenticate with Firebase and Ionic 3 - Email/Password and Google Sign-In**". 2015. Disponível em: <http://appseed.io/blog/how-authenticate-firebase-and-ionic-3-emailpassword-and-google-sign>. Acesso em: 10 mar. 2019.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6028**: informação e documentação: Resumo - Apresentação. Rio de Janeiro, 2003. 2p.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6027**: informação e documentação: Sumário - Apresentação. Rio de Janeiro, 2012. 3p.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 14724**: informação e documentação - Trabalhos acadêmicos - Apresentação. Rio de Janeiro, 2011. 11p.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023**: informação e documentação: Referências - Elaboração. Rio de Janeiro, 2018. 68p.

BARRUCHO, Luís. "**O que falta para o Brasil doar mais sangue?**". BBC. 2015. Disponível em: https://www.bbc.com/portuguese/noticias/2015/08/150812_sangue_doacoes_brasil_gb. Acesso em: 12 abr. 2019.

BERALDO, Nicole. "**Ministério da Saúde convoca população para doar sangue**". Portal do Governo Brasileiro. 2018. Disponível em: <http://www.saude.gov.br/noticias/agencia-saude/42673-ministerio-da-saude-convoca-populacao-para-doar-sangue>. Acesso em: 24 nov. 2019.

BERNARDI, Ricardo. "**brModelo Portable 2.0**". RBTech 2008-2019. Disponível em: <http://dev.rbtech.info/download-brmodelo-portable/>. Acesso em: 26 nov. 2019.

BOUCHEFRA, Ahmed. "**Create Ionic 4/Angular CRUD Application with Firestore and AngularFire2**". Techiediaries. 2017. Disponível em: <https://www.techiediaries.com/ionic-firestore-crud/>. Acesso em: 02 abr. 2019.

BRASIL ECONÔMICO. "**Brasil econômico: 69% dos brasileiros já têm acesso à internet pelo celular, afirma IBGE**". iG Tecnologia. 2018. Disponível em: <https://tecnologia.ig.com.br/2018-04-27/acesso-a-internet.html>. Acesso em: 13 abril. 2019.

BRASILEIRO, Roberto. "**Manifesto Ágil, o que é e qual a sua história**". Método ágil. 2017-2019. Disponível em: <http://www.metodoagil.com/manifesto-agil/>. Acesso em: 19 nov. 2019.

CARVALHO, Lucas. "**WhatsApp: história, dicas e tudo que você precisa saber sobre o app**". Olhar Digital 2005 – 2019. Disponível em: <https://olhardigital.com.br/noticia/whatsapp-historia-dicas-e-tudo-que-voce-precisa-saber-sobre-o-app/80779>. Acesso em: 26 nov. 2019.

CASTELLI, Ian. "**Trello: como esta ferramenta pode ajudar você a organizar a sua vida**". TecMundo 2019. Disponível em: <https://www.tecmundo.com.br/organizacao/75128-trello-ferramenta-ajudar-voce-organizar-vida.htm>. Acesso em: 26 nov. 2019.

CLOUDFLARE. "**What is BaaS? | Backend-as-a-Service vs. Serverless | Cloudflare**". Cloudflare. 2019. Disponível em: <https://www.cloudflare.com/learning/serverless/glossary/backend-as-a-service-baas/>. Acesso em 24 nov. 2019.

CODACY. **“Automate your code quality”**. Codacy 2019. Disponível em: https://www.codacy.com/?utm_source=GoogleAds&utm_medium=ppc&utm_campaign=GoogleAdsBrand-Phrase&utm_term=codacy&utm_device=c&gclid=Cj0KCQiAt_PuBRDcARIsAMNIBdoX2HHuwKbqqbMYoxogTYylxyQNmxdq_-zMloY7YxerlBB-B2pvkW0aAsmGEALw_wcB. Acesso em: 26 nov. 2019.

DIALOGFLOW. **“Build natural and rich conversational experiences”**. Google 2019. Disponível em: <https://dialogflow.com/>. Acesso em: 24 nov. 2019.

DEVMEDIA. **“Orientações básicas na elaboração de um diagrama de classes”**. DEVMEDIA. 2019. Disponível em: <https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama-de-classes/37224>. Acesso em: 11 dez. 2019.

DEVMEDIA. **“Visual Studio Code: O que achamos”**. DEVMEDIA 2019. Disponível em: <https://www.devmedia.com.br/visual-studio-code-o-que-achamos/37978>. Acesso em: 26 nov. 2019.

DUQUE, Gabriel. **“Metodologia ágil: um presente da indústria de software para todo o universo da gestão”** Runrun.it Blog. 2017. Disponível em: <https://blog.runrun.it/metodologia-agil/>. Acesso em: 19 nov. 2019.

EIS, Diego. **“O básico: O que é HTML?”**. Tableless 2019. Disponível em: <https://tableless.com.br/o-que-html-basico/>. Acesso em: 26 nov. 2019.

FIREBASE. **“Cloud Firestore”**. Firebase. 2018. Disponível em: <https://firebase.google.com/docs/firestore/>. Acesso em: 14 abr. 2019.

FIREBASE. **“Firebase helps mobile app teams succeed”**. Firebase. 2019. Disponível em: <https://firebase.google.com/?hl=pt-br>. Acesso em: 14 abr. 2019.

FIREBASE. **“Supported data types”**. Google Developers. 2019. Disponível em: <https://firebase.google.com/docs/firestore/manage-data/data-types>. Acesso em: 29 maio. 2019.

FUNDAÇÃO PRO SANGUE. **“Onde e como doar?”**. 2019. Disponível em: <http://www.prosangue.sp.gov.br/home/Default.html>. Acesso em: 25 mar. 2019.

G1. **“Doação de sangue: 1,6% da população brasileira é doadora, diz Ministério da Saúde”**. G1. 2018. Disponível em: <https://g1.globo.com/bemestar/noticia/doacao-de-sangue-16-da-populacao-brasileira-e-doadora-diz-ministerio-da-saude.ghtml>. Acesso em: 12 abr. 2019.

GIL, A. C. Como elaborar projetos de pesquisa. São Paulo: Atlas, 2002. P. 41.

GIT. **“—everything-is-local”**. Git 2019. Disponível em: <https://git-scm.com/>. Acesso em: 26 nov. 2019.

GITGUARDIAN. **“Modern monitoring & security”**. GitGuardian 2019. Disponível em: <https://www.gitguardian.com/>. Acesso em: 26 nov. 2019.

GITHUB. "**Built for developers**". GitHub Inc. 2019. Disponível em: <https://github.com/>. Acesso em: 26 nov. 2019.

GONÇALVES, Tássio. "**Pré-processador CSS? Sass? O que é e por onde começar!**". Bencode 2017. Disponível em: <https://bencode.com.br/pre-processador-css-sass/>. Acesso em: 26 nov. 2019.

GONSALVES, Elisa Pereira. **Iniciação à pesquisa científica**. Campinas, SP: Editora Alínea, 2001.

GOOGLE. "**Comece gratuitamente e pague em escala somente pelo que usar**". Firebase. 2019. Disponível em: <https://firebase.google.com/pricing/?hl=pt-br>. Acesso em: 14 abr. 2019.

GOOGLE. "**Firestore Services**". Google Developers. 2019. Disponível em: <https://developers.google.com/actions/tools/assistant-firebase-services>. Acesso em: 14 abr. 2019.

GOOGLE. "**Google Maps Platform**". Google Developers. 2019. Disponível em: <https://developers.google.com/maps/documentation/javascript/reference/map>. Acesso em: 08 out. 2019.

GOOGLEPLAY. "**Hemoliga**". Google 2019. Disponível em: <https://play.google.com/store/apps/details?id=br.com.hemoliga&hl=pt-BR>. Acesso em: 11 mar. 2019a.

GOOGLEPLAY. "**Partiu doar Sangue**". Google 2019. Disponível em: <https://play.google.com/store/apps/details?id=com.ionicframework.app190851&hl=pt-BR>. Acesso em: 11 mar. 2019b.

GOOGLEPLAY. "**Time do Sangue**". Google 2019. Disponível em: <https://play.google.com/store/apps/details?id=com.time.sangue&hl=pt-BR>. Acesso em: 11 mar. 2019c.

GIT. "**About**". Git. 2019. Disponível em: <https://git-scm.com/about>. Acesso em: 14 abr. 2019.

GITHUB. "**Build better together**". GitHub. 2019. Disponível em: <https://github.com/>. Acesso em: 14 abr. 2019.

HANHN, Jéssica. "**Perfil dos usuários de smartphones no Brasil**". Bitzen. 2018. Disponível em: <https://bitzen.com.br/aplicativos/perfil-dos-usuarios-de-smartphones-no-brasil>. Acesso em: 14 abr. 2019.

HOMEHOST. "**O que é banco de dados?**". HomeHost 2006 – 2019. Disponível em: <https://www.homehost.com.br/blog/tutoriais/mysql/o-que-e-um-banco-de-dados/>. Acesso em: 26 nov. 2019.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Normas de apresentação tabular**. 3. ed. Rio de Janeiro: IBGE, 2003. 60p.

IONIC. "**All about Ionic**". **Ionic Framework**. 2019. Disponível em: <https://ionicframework.com/about>. Acesso em: 14 abr. 2019.

IONIC FRAMEWORK. "**Concepts**". Ionic Framework. 2019. Disponível em: <https://ionicframework.com/docs/v3/intro/concepts/>. Acesso em: 24 nov. 2019.

KNOP, Igor. "**Trabalho, estudo e diversão em um balai de gato só**". 2009-2019. Disponível em: <http://igorknop.com.br/wp/disciplinas/ifsudeste/analise/diagrama-de-atividades/>. Acesso em: 10 maio. 2019.

LABOISSIÈRE, Paula. "**Doadores de sangue somam 1,6% da população; jovens são maioria**". Agência Brasil. 2018. Disponível em: <http://agenciabrasil.ebc.com.br/saude/noticia/2018-06/pelo-menos-16-da-populacao-brasileira-doa-sangue-jovens-sao-maioria>. Acesso em: 12 abr. 2019.

LEASE, Diana. "**TypeScript: O que é e quando é útil?**". Medium 2019. Disponível em: <https://medium.com/front-end-weekly/typescript-what-is-it-when-is-it-useful-c4c41b5c4ae7>. Acesso em: 26 nov. 2019.

LEGADO BRASIL. "**Apenas 1,8% dos brasileiros são doadores de sangue**". 2017. Disponível em: <http://legado.brasil.gov.br/noticias/saude/2017/06/apenas-1-8-dos-brasileiros-sao-doadores-de-sangue>. Acesso em: 19 nov. 2019, às 18:06.

LIMA, Davi de. "**Modelos softwares com Astah Community**". Techtudo 2000-2019. Disponível em: <https://www.techtudo.com.br/tudo-sobre/astah-community.html>. Acesso em: 26 nov. 2019.

LINO, Mariana. "**O QUE É APACHE CORDOVA?**" JOURNEY 2017. Disponível em: <https://acordocoletivo.org/2017/07/02/o-que-e-apache-cordova/>. Acesso em: 26 nov. 2019.

LUIS, Guilherme. "**TypeScript: saiba tudo sobre a tecnologia**". Programathor. 2019. Disponível em: <https://programathor.com.br/blog/typescript>. Acesso em: 21 de nov. 2019.

MARQUES, Brendon. "**O Que é GitHub e Para Que é Usado?**". Hostinger Blog 2004 – 2019. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-github/>. Acesso em: 26 nov. 2019.

MARQUES, Rafael. "**O que é HTML? Entenda de forma descomplicada**". Home Host. 2006 - 2019. Disponível em: <https://www.homehost.com.br/blog/tutoriais/o-que-e-html/>. Acesso em 22 de nov. 2019.

MINISTÉRIO DA SAÚDE. "**Ministério da Saúde reforça campanha para incentivar doação de sangue**". Portalms. 2018. Disponível em: <http://portalms.saude.gov.br/noticias/agencia-saude/44728-saude-reforca-campanha-para-incentivar-doacao-de-sangue>. Acesso em: 12 abr. 2019.

MINISTÉRIO DA SAÚDE. **PORTARIA Nº 158, DE 4 DE FEVEREIRO DE 2016**. Disponível em: http://bvsms.saude.gov.br/bvs/saudelegis/gm/2016/prt0158_04_02_2016.html. Acesso em: 31 mar. 2019.

MONTEBELLO, Sandra Camargo. "**Doação de sangue: requisitos, cuidados e onde doar**". minhavida. 2019. Disponível em: <https://www.minhavida.com.br/saude/tudo-sobre/18656-doacao-de-sangue>. Acesso em: 19 nov. 2019, às 17:57.

NODEBR. "**O que é a NPM do Node.js**". Node BR 2019. Disponível em: <http://nodebr.com/o-que-e-a-npm-do-nodejs/>. Acesso em: 26 nov. 2019

O'GRADY, Stephen. "**The RedMonk Programming Language Rankings: January 2019**". RedMonk. 2019. Disponível em: <https://redmonk.com/sogrady/2019/03/20/language-rankings-1-19/>. Acesso em: 14 abr. 2019.

OLHAR DIGITAL. "**7 aplicativos para doadores de sangue**". 2005-2019. Disponível em: <https://olhardigital.com.br/lu-explica/noticia/7-aplicativos-para-doadores-de-sangue/72434>. Acesso em: 15 fev. 2019.

PHIXIES. "**O que é Adobe Photoshop?**". Phixies 2019. Disponível em: <https://phixies.com.br/tecnologias/Adobe-Photoshop/>. Acesso em: 26 nov. 2019.

PORTAL GSTI. "**O que é UML?**". Portal GSTI. 2019. Disponível em: <https://www.portalgsti.com.br/uml/sobre/>. Acesso em: 11 dez. 2019.

PRATES, Lazaro. "**5 vantagens do Ionic para desenvolver suas aplicações mobile**". Blog Alura. 2016. Disponível em: <https://blog.alura.com.br/5-vantagens-do-ionic-para-desenvolver-suas-aplicacoes-mobile/>. Acesso em: 14 abr. 2019.

RED HAT. "**What is an API?**". Red Hat. 2019. Disponível em: <https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces>. Acesso: 22 nov. 2019.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Universidades, 2011.

SPACEPROGRAMMER. "**Introdução ao Modelo de Dados e seus níveis de abstração**". 2016. Disponível em: <http://spaceprogrammer.com/bd/introducao-ao-modelo-de-dados-e-seus-niveis-de-abstracao/>. Acesso em: 13 maio. 2019.

STACK OVERFLOW. "**Regenerate Web API key of Google Firebase**". Stack Overflow. 2017. Disponível em: <https://stackoverflow.com/questions/44838881/regenerate-web-api-key-of-google-firebase>. Acesso em: 02 abr. 2019.

STACK OVERFLOW. "**What is the best way to delete a component with CLI**". Stack Overflow. 2017. Disponível em: <https://stackoverflow.com/questions/41354755/what-is-the-best-way-to-delete-a-component-with-cli>. Acesso em: 02 abr. 2019.

TEIXEIRA, Fabricio. "**Tudo sobre o Adobe Experience Design CC, a nova ferramenta de UX da Adobe**". UX Collective BR 2016. Disponível em: <https://brasil.uxdesign.cc/tudo-sobre-o-adobe-experience-design-cc-a-nova-ferramenta-de-ux-da-adobe-a24070ec6ca1>. Acesso em: 26 nov. 2019.

VIANA, Daniel. "**Firebase: descubra no que esta plataforma pode te ajudar**". TreinaWeb Tecnologia LTDA 2004 – 2019. Disponível em: <https://www.treinaweb.com.br/blog/firebase-descubra-no-que-esta-plataforma-pode-te-ajudar/>. Acesso em: 26 nov. 2019.

VIEIRA, Rodrigo. "**UML — Diagrama de Casos de Uso**". Operational TI. 2019. Disponível em: <https://medium.com/operacionalti/uml-diagrama-de-casos-de-uso-29f4358ce4d5>. Acesso em: 11 dez. 2019.