

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CAMPUS TIMÓTEO**

Vinícius Henrique de Moraes

**SANGUE++, APLICATIVO PARA APRIMORAMENTO DA
LOCALIZAÇÃO DE DOADORES DE SANGUE**

Timóteo

2022

Vinícius Henrique de Moraes

**SANGUE++, APLICATIVO PARA APRIMORAMENTO DA
LOCALIZAÇÃO DE DOADORES DE SANGUE**

Monografia apresentada à Coordenação de Engenharia de Computação do Campus Timóteo do Centro Federal de Educação Tecnológica de Minas Gerais para obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Dr Maurílio Alves Martins da Costa

Timóteo

2022

Vinicius Henrique de Moraes

**SANGUE++, APLICATIVO PARA APRIMORAMENTO DA LOCALIZAÇÃO
DE DOADORES DE SANGUE**

Trabalho de Conclusão de Curso
apresentado ao Curso de Engenharia de Computação
do Centro Federal de Educação Tecnológica de
Minas Gerais, campus Timóteo, como requisito
parcial para obtenção do título de Engenheiro de
Computação.

Trabalho aprovado. Timóteo, 18 de Agosto de 2022:



Prof. Dr. Maurílio Alves Martins da Costa
Orientador



Prof. Me. Marcelo de Souza Balbino
Professor Convidado



Prof. Ma. Deisymar Botega Tavares
Professor Convidado

Timóteo
2022

Dedico a
algumas pessoas.

Agradecimientos

Resumo

A doação de sangue é um processo essencial para manutenção da saúde em qualquer país. De acordo com a OMS, no mínimo 3,5% da população total devia ser doadora. A base brasileira de doadores de sangue é constituída por apenas 1,8% da população total do país, longe do valor considerado mínimo pela OMS e ainda mais distante dos 5% considerados suficientes para atender as necessidades brasileiras. Levando em consideração este cenário crítico, se faz necessário a busca de estratégias para contornar o problema relacionado a falta de doadores. Para contornar tal problema, pode-se pensar na área de tecnologia que vem se desenvolvendo muito nos últimos anos e agregando em diversos pontos da área da saúde. Mais especificamente, podemos falar do crescimento das aplicações para dispositivos móveis. Essas aplicações tem como resultado a anulação de restrições existentes na computação tradicional formada por computadores estáticos, possibilitando ao usuário poder de processamento independente de onde esteja, além de comodidade e integração fácil com outros usuários. O presente trabalho se baseia na implementação de recursos tecnológicos sendo eles, geolocalização e aprendizado de máquina, além do desenvolvimento de uma aplicação capaz de localizar usuários e classificá-los como possíveis doadores. Para geração do modelo foi utilizada a linguagem de programação Python e o algoritmo Extra Trees Classifier, para implementação da geolocalização foi utilizado o Expo e para desenvolvimento do protótipo foi utilizado React Native e Node.js. A aplicação final alcançou o objetivo de localização e classificação de doadores.

Palavras-chaves: Geolocalização, Aprendizado de máquina, Aplicações móveis.

Abstract

Blood donation is an essential process for maintaining health in any country. According to the WHO, at least 3.5% of the total population should be donors. The Brazilian blood donor base is constituted by only 1.8% of the country's total population, far from the value considered minimum by the WHO and even more distant from the 5% considered sufficient to meet the Brazilian needs. Considering this critical scenario, it is necessary to look for strategies to overcome the problem related to the lack of donors. To get around this problem, we can think of the area of technology that has been developing a lot in recent years and adding to several points in the health area. More specifically, we can talk about the growth of applications for mobile devices. These applications have as a result the annulment of existing restrictions in traditional computing formed by static computers, allowing the user processing power regardless of where he is, as well as convenience and easy integration with other users. The present work is based on the implementation of technological resources such as geolocation and machine learning, as well as the development of an application capable of locating users and classifying them as possible donors. The Python programming language and the Extra Trees Classifier algorithm were used to generate the model, the Expo was used to implement geolocation, and React Native and Node.js were used to develop the prototype. The final application achieved the goal of donor location and classification.

keywords: Geolocation, Machine Learning, Mobile Applications.

Lista de ilustrações

Figura 1 – Notificação Hemominas	13
Figura 2 – Tela de cadastro do aplicativo BloodSYS	16
Figura 3 – Aplicativo DOE+	17
Figura 4 – Fluxo de desenvolvimento do projeto	19
Figura 5 – Classificação do aprendizado indutivo	24
Figura 6 – Métricas de avaliação dos modelos de aprendizado de máquina	26
Figura 7 – Árvore de decisão	27
Figura 8 – Node.Js. Gerenciamento de conexões	31
Figura 9 – Etapas do trabalho	34
Figura 10 – Fluxograma	38
Figura 11 – Estrutura da base de dados	39
Figura 12 – Importância das variáveis no modelo	43
Figura 13 – Instalação Pacotes Expo	44
Figura 14 – Importação Pacotes Expo	44
Figura 15 – Implementação função getLocationAsync	45
Figura 16 – Código de implementação da API	46
Figura 17 – Requisição para API de predição	47
Figura 18 – Função linkingWhats	48
Figura 19 – Telas de cadastro	50
Figura 20 – Tela de login	51
Figura 21 – Tela de listagem de doadores por distância	52
Figura 22 – Tela de listagem de possíveis doadores	54
Figura 23 – Detalhes doador e envio de mensagem Whatsapp	56

Lista de tabelas

Tabela 1 – Comparação da separação de base de 30%	40
Tabela 2 – Comparação da separação de base de 20%	41
Tabela 3 – Médias das métricas da base 30%	41
Tabela 4 – Médias das métricas da base 20%	42
Tabela 5 – Comparação da separação de base sem aleatoriedade	42
Tabela 6 – Previsão do modelo	43
Tabela 7 – Avaliação da aplicação - Técnica de geolocalização	53
Tabela 8 – Avaliação da aplicação - Técnica de aprendizado de máquina	55

Lista de abreviaturas e siglas

PRF	Polícia Rodoviária Federal
API	Application Programming Interface (Interface de Programação de Aplicação)
NPM	Gerenciador de Pacotes do Node (Node Package Manager)
OMS	Organização Mundial da Saúde
APP	Application (Aplicativo)
GPS	Global Positioning System (Sistema de Posicionamento Global)
RAM	Random Access Memory (Memória de Acesso Aleatório)

Sumário

1	INTRODUÇÃO	11
1.1	Justificativa	13
1.2	Objetivos	14
2	TRABALHOS CORRELATOS	15
2.1	BloodSYS: controlando o processo de doação de sangue para hemo-centros	15
2.2	DOE+: Um Aplicativo Móvel de Cunho Social para Agendamento de Do-ação de Sangue no Hemocentro Público de Alagoas	16
2.3	Uso de Aprendizado de Máquina para Classificação de Risco de Aciden-tes em Rodovias	17
2.4	Classificação da Vegetação do Parque Nacional da Chapada das Mesas, Maranhão, Usando OBIA, Machine Learning e Softwares Livres	20
3	FUNDAMENTAÇÃO TEÓRICA	21
3.1	Doação de sangue	21
3.1.1	Requisitos para doação	21
3.1.2	Fracionamento	22
3.2	Tecnologia e o apoio à saúde	22
3.3	Aplicativos móveis	23
3.4	Aprendizado de máquina	23
3.4.1	Métricas de Avaliação de modelo de aprendizado de máquina	24
3.4.1.1	Matriz de confusão	24
3.4.1.2	Acurácia, Precisão, Recall e F1-Score	25
3.5	Decision Tree	26
3.5.1	Extra Trees Classifier	27
3.6	Python	28
3.6.1	Scikit-Learn	28
3.7	React e React Native	28
3.7.1	Expo	29
3.8	Node JS	30
3.8.1	NPM	32
3.9	Geolocalização	32
4	PROCEDIMENTOS METODOLÓGICOS	34
4.1	Escolha e estudo aprofundado das técnicas	34
4.2	Levantamento de requisitos	34
4.3	Implementação das técnicas	35
4.4	Criação da API Python	36

4.5	Implementação do protótipo	36
4.6	Avaliação do protótipo	36
5	DESENVOLVIMENTO DO PROJETO	37
5.1	Levantamento de requisitos	37
5.2	Estruturação do aplicativo	37
5.3	Implementação do aplicativo	38
5.3.1	Escolha das tecnologias e técnicas	38
5.3.2	Treinamento do modelo	39
5.3.3	Implementação da Geolocalização	44
5.3.4	Implementação da API de predição	45
5.3.5	Implementação do protótipo	47
6	RESULTADOS	49
6.1	Implementação dos recursos	49
6.2	Implementação do aplicativo	49
7	LIMITAÇÕES E TRABALHOS FUTUROS	57
7.1	Limitações	57
7.2	Trabalhos futuros	57
8	CONSIDERAÇÕES FINAIS	59
	REFERÊNCIAS	60

1 Introdução

“O que se compartilha, se multiplica”.
Papa Francisco

Conforme cita Souza e Elias (2005), o sangue é um líquido que flui pelo sistema circulatório tendo como principais funcionalidades o transporte de oxigênio e outros nutrientes, hormônios, eletrólitos, água e resíduos do metabolismo celular. Esse líquido pode ser transferido de uma pessoa para outra em um procedimento chamado de transfusão de sangue, desde que se garanta que o sangue do receptor e o do doador serão compatíveis, reduzindo desta forma o desenvolvimento de reações. Geralmente, a finalidade da transfusão é tratar alguma condição, principalmente anemia profunda ou hemorragias graves, além de também ser indicada em caso de hemofilia, queimaduras ou durante grandes cirurgias (SAÚDE, 2021).

O sangue usado na transfusão é coletado através do processo de doação de sangue, podendo ser fracionado e estocado para uso posterior. Espera-se que este estoque seja suficiente para atender as demandas por este recurso, portanto, as doações de sangue devem ser frequentes e realizadas por grande parte da população.(GURGEL, 2019)

No século passado, para manutenção de estoque, acontecia bonificação para quem realizasse o ato de doar sangue, também era comum, a venda do sangue. Conforme o artigo 14 da Lei Nº 10.205 da Constituição Federal do Brasil (2001) é proibida a remuneração ao doador pela doação de sangue e a comercialização da coleta, processamento, estocagem, distribuição e transfusão do sangue, componentes e hemoderivados. A manutenção do nível de sangue passou a ser estritamente dependente de doações voluntárias, o que tem sido uma tarefa delicada visto que um dos problemas da área de saúde é a falta de doação de sangue(RODRIGUES; JÚNIOR; PEREIRA, 2017).

Conforme aponta o Ministério da Saúde (2020), apenas aproximadamente 1,8% da população doa sangue no Brasil, longe dos 3,5% considerados mínimos pela OMS. Para que fosse atendida a demanda de sangue, 5% da população deveria ser doador frequente. O Brasil necessita diariamente de 5500 bolsas de sangue e como a quantidade de doação é abaixo do esperado, milhares de pessoas acabam perecendo todo ano por falta de sangue disponível (SANTOS, 2010).

A situação do país é refletida no Vale do Aço, região metropolitana brasileira no interior do estado de Minas Gerais, composta pelas cidades de Coronel Fabriciano, Ipatinga, Santana do Paraíso, Timóteo e pelo colar metropolitano, que é constituído por outros 24 municípios. A realidade do Vale do Aço é que falta sangue o ano inteiro visto que menos de 2% da população é assídua na doação (Jornal Diário do Aço, 2021).

Pereira et al. (2016) explicam no artigo que entre os problemas que levam à falta de doação de sangue se destacam o medo do doador, tanto de sentir dor quanto de contrair doenças e a grande quantidade de requisitos a serem atendidos para que se possa realizar a doação . De

acordo com informações do Ministério da Saúde (2020), enquadram nesses requisitos, ter entre 18 e 69 anos, estar pesando mais de 50 quilos, respeitar os intervalos de doação, obedecer condições básicas de saúde, entre outros.

Se faz necessário a busca de estratégias para contornar o problema relacionado a falta de doadores. Uma alternativa seria o uso da tecnologia da informação que se apresenta como uma das áreas que mais sofrem evolução nos últimos anos e pode ser aliada de diversas outras áreas, sendo uma delas a área de saúde.

Mais especificamente, podemos falar da propagação de dispositivos móveis. A ideia básica é a anulação de restrições existentes na computação tradicional formada por computadores estáticos, possibilitando ao usuário poder de processamento independente de onde esteja, além de comodidade e integração fácil com outros usuários, permitindo assim, o apoio em diversos cenários. (MATEUS; LOUREIRO, 1998).

Kestring (2014) relata em seu artigo que com a disseminação de dispositivos móveis, uma técnica amplamente utilizada é a de geolocalização. Esse é um recurso que permite determinar a posição geográfica de um dispositivo com base em um sistema de coordenadas. No cenário de doação de sangue, essa técnica pode ser utilizada para encontrar os doadores que estão em uma região próxima. Outra técnica que sofreu grande evolução e pode agregar ao processo de doação é a de aprendizado de máquina, onde a partir de dados de entrada, coletados previamente, é gerado um modelo capaz de prever o resultado de saída.

Existem diversas aplicações móveis voltadas para doação de sangue, como por exemplo DOE+ desenvolvido no trabalho do Marcílio Junior (2018) e o BloodSYS, projeto desenvolvido por Severo e Santos (2018), porém elas possuem foco voltado para gerenciamento do estoque de sangue e de doadores, não dando ênfase ao aprimoramento da busca por doadores.

Pensando no cenário de inúmeros avanços tecnológicos e problemas relacionados a doação de sangue. Seria possível a criação de uma ferramenta computacional que fosse capaz de aprimorar a busca por doadores de sangue na região do Vale do Aço através do uso de técnicas como aprendizado de máquina e geolocalização?

Vários estudos têm demonstrado a importância do uso de algoritmos de aprendizado de máquina para classificação de alvos com intuito de fazer previsões e analisar dados rapidamente criando oportunidades lucrativas (CARVALHO et al., 2019).

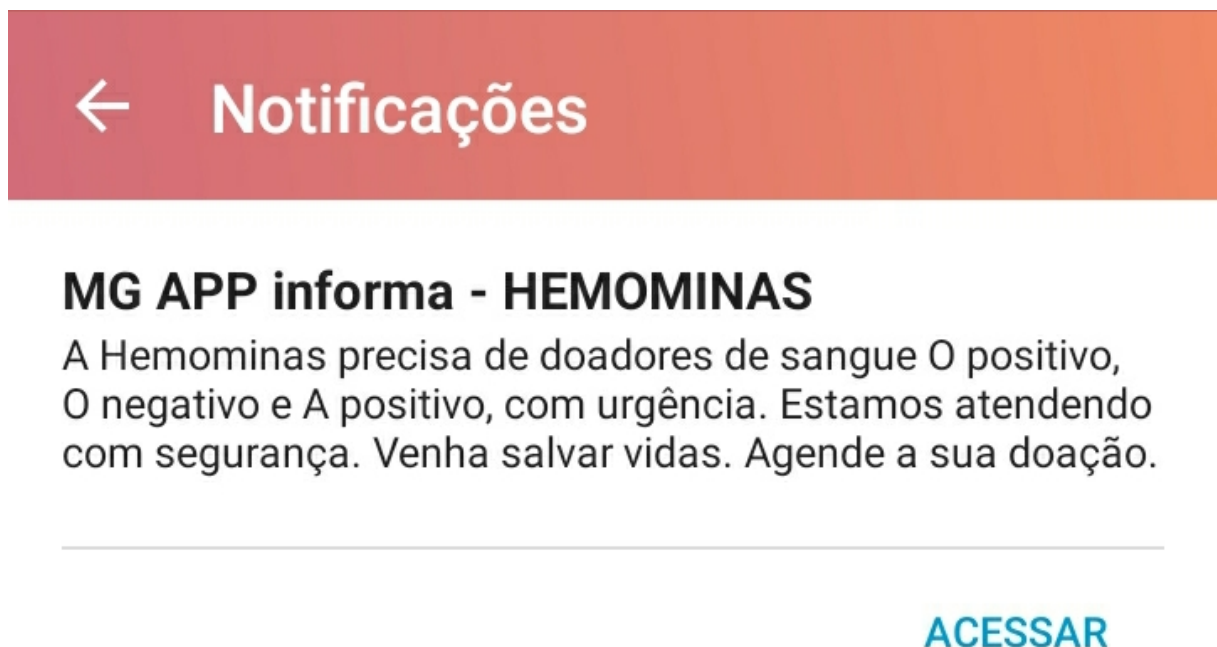
Por esta razão, este trabalho se baseia na melhoria da busca de doadores de sangue através do desenvolvimento de um aplicativo que implemente aprendizado de máquina e geolocalização usando como alimentação de dados uma base histórica de doadores de sangue contendo informações de número total de doações, total de sangue doado, meses desde a primeira doação, meses desde a última doação e uma variável binária que representa doação em março de 2007.

1.1 Justificativa

Segundo o Ministério da Saúde (2020), o número de doadores de sangue atualmente é muito abaixo do necessário para cobrir as demandas de sangue, portanto, esse trabalho justifica-se pela necessidade de alcançar um nível maior de doação sanguínea mantendo assim, os níveis de sangue em estoque em situação estável.

Conforme informado pela Fundação Hemominas (2021), desde o início da pandemia, o nível de sangue em seus estoques caiu em média de 30% e alguns tipos sanguíneos, a queda alcançou a marca de 50% entrando em nível crítico. Campanhas foram lançadas como pode ser observado na figura 1, com a finalidade de declarar emergência e encontrar doadores para suprirem essa necessidade.

Figura 1 – Notificação Hemominas



Fonte: Hemominas via aplicativo MG App Cidadao

Para tentar captar novos doadores e suprir a necessidade dos estoques de sangue, são realizadas campanhas como fonte de conscientização à população, mostrando a importância da doação voluntária, porém para que se efetive realmente a doação se faz necessária, a boa vontade por parte do doador uma vez que não se tem contato com o mesmo (BARRA, 2015).

O trabalho procura auxiliar na busca por doadores de sangue através do desenvolvimento de um aplicativo para dispositivos móveis que utiliza técnicas para aperfeiçoamento da busca por doadores compatíveis. O aplicativo permite o contato de forma direta ao usuário tornando o processo de doação muito mais ágil podendo assim, contribuir para que vidas humanas sejam poupadas.

O aplicativo irá permitir ao usuário a realização de cadastro, a realização de acesso e após

acessar a ferramenta será possível a busca por doadores compatíveis com maior probabilidade de doação e a busca por doadores mais próximos. Essas funcionalidades serão implementadas de forma simples do ponto de vista de interface, procurando evitar a desistência do uso do aplicativo por parte de usuários.

1.2 Objetivos

O objetivo geral deste trabalho é desenvolver uma ferramenta para aperfeiçoamento da busca por doadores de sangue na região do Vale do Aço utilizando técnicas de aprendizado de máquina e geolocalização.

Também, objetivam-se mais especificamente:

1. Identificar e implementar um modelo de aprendizado de máquina para classificação de prováveis indivíduos para doação;
2. Identificar e implementar técnicas de localização geográfica para encontrar doadores mais próximos;
3. Realizar avaliação da ferramenta baseada na assertividade¹ das técnicas implementadas.

¹ Assertividade no contexto deste trabalho, se refere a quão próximo do real a técnica de geolocalização informou as distâncias entre doadores e a quão preciso o modelo de aprendizado foi ao classificador novos doadores.

2 Trabalhos correlatos

Nesta seção se encontra os trabalhos que se relacionam com a ideia central deste trabalho e com as técnicas e tecnologias usadas por ele. As buscas foram feitas usando o critério de que os objetivos dos trabalhos deveriam ser baseados em aperfeiçoar e/ou controlar o processo de doação de sangue. Outro fator relevante para escolha dos trabalhos foi a data de publicação, onde dado preferência para os mais recentes.

As pesquisas foram realizadas nas seguintes bases de pesquisas científicas: Google Scholar e ACM Digital Library. Os projetos escolhidos foram: BloodSYS (SEVERO; SANTOS, 2018) e DOE+ (Marcílio Junior, 2018).

2.1 BloodSYS: controlando o processo de doação de sangue para hemocentros

BloodSys é um trabalho realizado por Severo e Santos (2018) que teve como objetivo implementar uma ferramenta de software para controle das etapas envolvidas no processo de doação de sangue para hemocentros e apresentar os seus resultados.

O primeiro passo do trabalho foi estudar o processo de doação de sangue e identificar problemas relacionados ao mesmo que poderiam ser resolvidos ou minimizados pelo uso de uma ferramenta de software. Após fazer o estudo, os autores definiram como objetivos específicos a identificação com antecedência de doadores não aptos e a integração das etapas de identificação, pré-triagem, triagem e doação.

Para ser possível o desenvolvimento da ferramenta, os autores fizeram análise de trabalhos semelhantes extraindo informações relevantes para o trabalho, fizeram um levantamento de requisitos do software além de realizarem a modelagem da ferramenta através de diagramas de casos de uso e de classes.

Na fase de implementação foi optado o uso das seguintes tecnologias:

- Java como linguagem de desenvolvimento;
- NetBeans IDE (Ambiente de Desenvolvimento Integrado) para elaboração do software;
- GlassFish como servidor de aplicação;
- Postgresql como gerenciador do banco de dados relacional.

Como resultado final foi obtido com êxito um protótipo como pode ser observado na figura 2. Essa ferramenta incorporou a principal funcionalidade objetiva que era a identificação prévia de doadores não aptos, reduzindo desta forma a morosidade no processo de doação. Embora este trabalho tenha feito uso de técnicas e tecnologias diferentes, compartilha um objetivo em comum com esta monografia: aperfeiçoamento do processo de busca por doadores.

Figura 2 – Tela de cadastro do aplicativo BloodSYS

Identificação:

Cadastro de pré-triagem:

The screenshot displays the registration interface of the BloodSYS application. It features a header with a rainbow-colored bar. Below the header, there are two main sections: 'Identificação:' and 'Cadastro de pré-triagem:'. The 'Cadastro de pré-triagem:' section contains several input fields for user data: 'Pulso (BPM)', 'Pressão Mín. (mmHg)', 'Pressão Máx. (mmHg)', 'Temperatura (°C)', 'Peso (Kg)', 'Altura (m)', 'Hemoglobina (g/dL)', and 'Hematócrito (%)'. At the bottom of this section, there is a dropdown menu with the name 'Pedro da Rosa' and a small downward arrow. Below the input fields, there are two buttons: a green 'Salvar' button and a blue 'Voltar' button.

Fonte: (SEVERO; SANTOS, 2018)

2.2 DOE+: Um Aplicativo Móvel de Cunho Social para Agendamento de Doação de Sangue no Hemocentro Público de Alagoas

O DOE+ aplicativo desenvolvido por Marcílio Junior (2018) para dispositivos móveis tem como objetivo potencializar a captação e a fidelização de doadores, principalmente do público jovem, além de buscar despertar a consciência da sociedade para a necessidade de manutenção do estoque de sangue com nível estratégico no atendimento às necessidades da população do estado. Ele oferece ao usuário a visualização e notificação de campanhas institucionais, cadastro e agendamentos de doações em horários pré-definidos, visualização do nível dos estoques de sangue e notificação de doadores quando o estoque do seu tipo de sangue estiver baixo conforme demonstrado na figura 3.

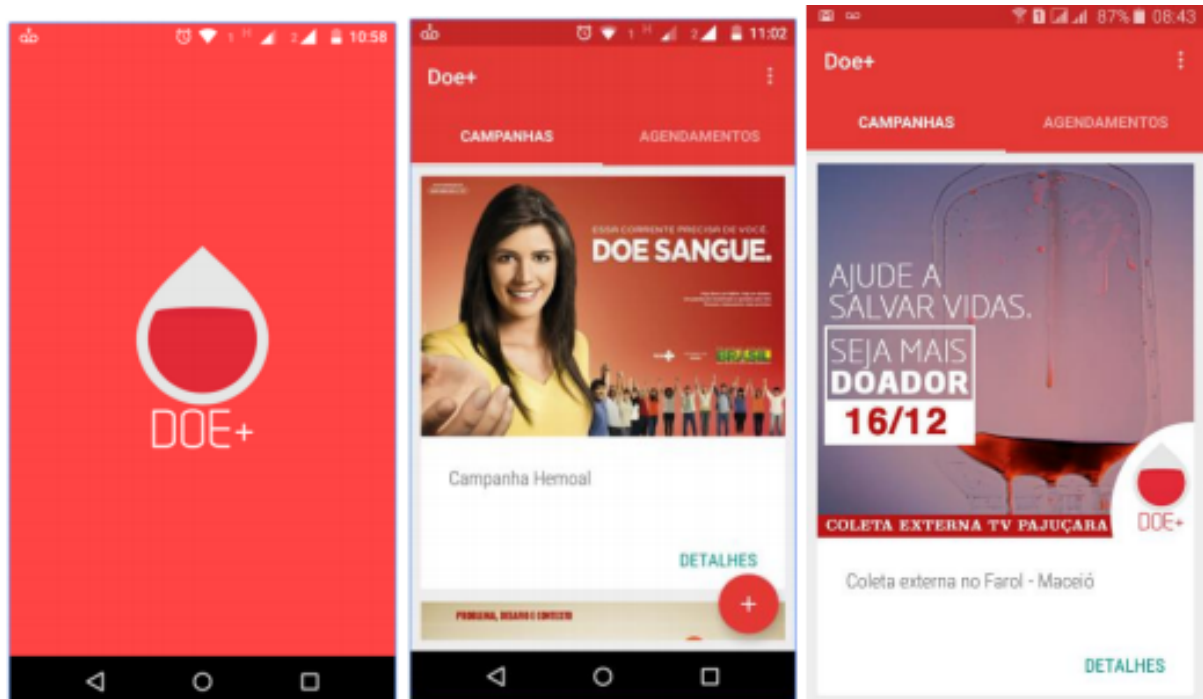
Este trabalho realizado por Marcílio Junior (2018) surgiu da necessidade de um hemocentro de Alagoas em ter um meio de divulgação de suas campanhas de doação. Para implantação do mesmo, o projeto foi dividido em 3 partes, sendo elas: visitas técnicas e entrevistas junto aos gestores e técnicos do hemocentro para levantamento do processo da doação de sangue, desenvolvimento e teste do aplicativo e realização de treinamento e capacitação dos técnicos do hemocentro para operacionalização do aplicativo.

Após três anos de uso, o DOE+ atingiu mais de 3.000 downloads, 1.432 usuários ativos e 420 agendamentos de doação confirmados. O aplicativo permitiu também que os usuários acompanhassem em tempo real os níveis de estoque de sangue, além de receberem notifica-

ções de campanhas de doação.

Embora seja um trabalho voltado para apoio da doação de sangue, o aplicativo não possui objetivo de evolução da de busca de doadores compatíveis e portanto não implementa nenhuma técnica relacionada.

Figura 3 – Aplicativo DOE+



Fonte: (Marcílio Junior, 2018)

2.3 Uso de Aprendizado de Máquina para Classificação de Risco de Acidentes em Rodovias

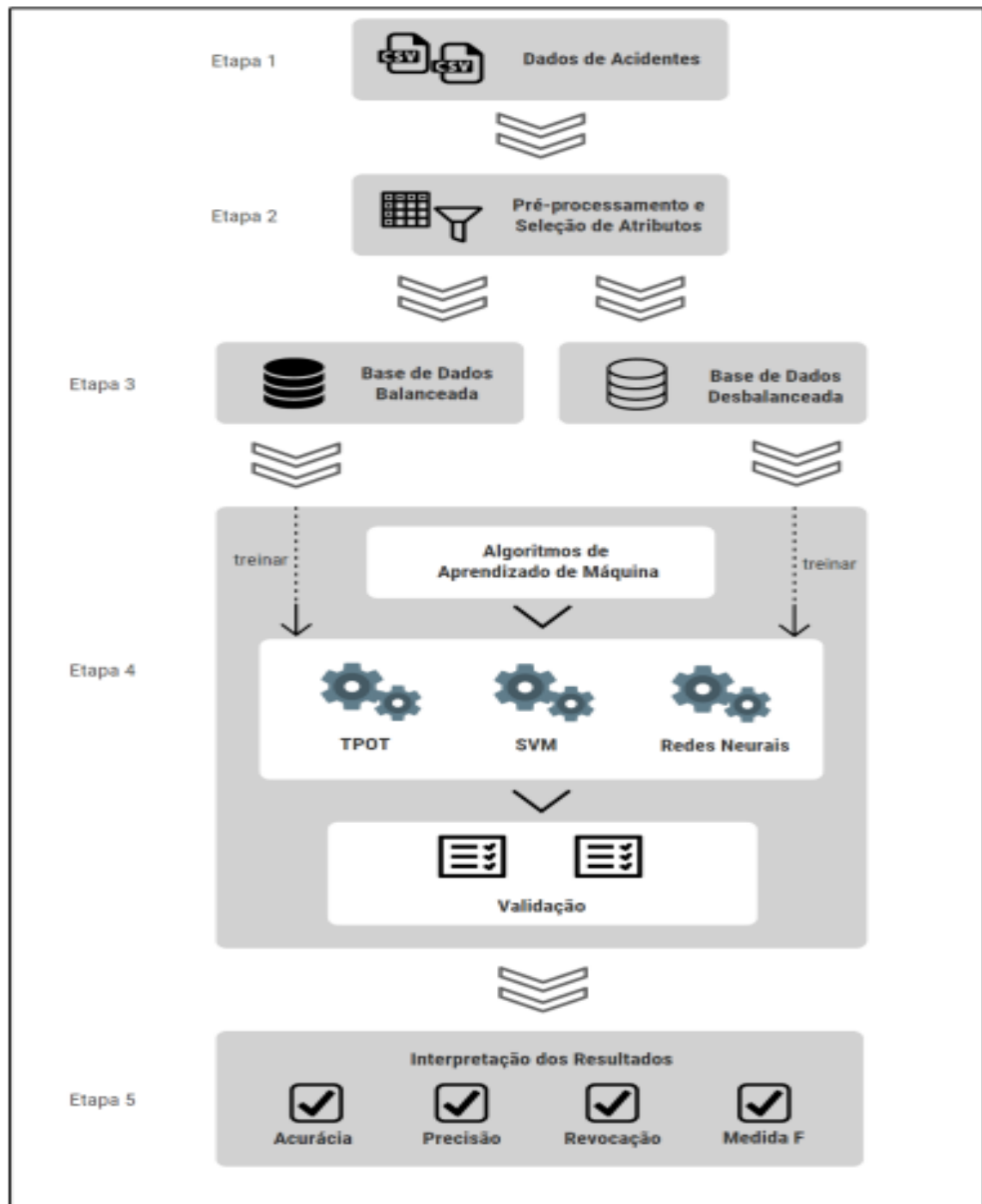
Embora aplicado em contexto muito diferente do trabalho em questão, o trabalho de Uso de Aprendizado de Máquina para Classificação de Risco de Acidentes em Rodovias fez uso de técnicas de aprendizado de máquina muito semelhantes:

- Support Vector Machine
- Redes Neurais
- Logistic Regression
- Extra Trees Classifier
- XGBoost Classifier
- Random Forest
- BernoulliNB

Foi um trabalho desenvolvido por Amorim (2019) no curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande. Este trabalho tinha como objetivo fazer uso de aprendizado de máquina para identificar e prever, nas rodovias brasileiras, trechos que possuem risco de acidentes graves.

Para realização do desenvolvimento do trabalho o autor seguiu o fluxo demonstrado pela figura 4.

Figura 4 – Fluxo de desenvolvimento do projeto



Fonte: (AMORIM, 2019)

Primeiro o autor coletou dados abertos do governo, disponibilizados pela Polícia Rodoviária Federal (PRF) todo ano referentes à acidentes e infrações de trânsito que ocorreram em todo o Brasil. Essa coleta gerou dados de 2007 à 2017 e possui aproximadamente 1,6 milhões de entradas de acidentes trazendo informações sobre o mesmo como data e horário, motivo, localização, condição meteorológica, quantidade de mortos, feridos, entre outros.

Após a coleta veio o pré-processamento de dados e a seleção dos atributos relevantes ao estudo. A base disponibilizada pela PRF possuía muitas informações irrelevantes para análise de riscos das rodovias, portanto o autor teve que realizar um filtro nas informações além de gerar algumas novas informações a partir dos dados originais como por exemplo a criação do atributo gravidade a partir da quantidade de mortos e feridos informados.

Com os dados coletados e bem estruturados, o autor teve que avaliar as melhores formas e algoritmos para obter um resultado mais consistente sobre os risco nas rodovias. Foram escolhidos diversas algoritmos e técnicas de aprendizado de máquina entre eles ExtraTreesClassifier e RandomForest. Foram realizados testes com esses algoritmos e comparado a performance de cada um levando em consideração a taxa de precisão.

Como resultado final o autor obteve resultados bastante satisfatórios, obtendo através dos classificadores, uma melhora considerável na identificação de trechos com risco de acidentes graves nas rodovias do Brasil.

2.4 Classificação da Vegetação do Parque Nacional da Chapada das Mesas, Maranhão, Usando OBIA, Machine Learning e Softwares Livres

Este estudo foi realizado por Carvalho et al. (2019). O principal objetivo foi classificar a vegetação do Parque Nacional da Chapada das Mesas (PNCM), utilizando métodos de OBIA e ML implementados em softwares de livre acesso. Foram utilizadas imagens de alta resolução espacial da constelação Dove PlanetScope, operada pela empresa Planet adquiridas nas estações seca e chuvosa.

No processo de classificação foi utilizado o algoritmo de RandomForest e foram estabelecidas 9 classes de vegetação: cerradão, cerrado denso, cerrado aberto, cerrado rupestre, mata de galeria, agricultura, pastagem e água.

De modo geral, o resultado atendeu as expectativas e alcançou acurácia geral de 71,5%.

De forma semelhante ao trabalho em questão, este trabalhou se baseou no uso de aprendizado de máquina para classificação de alvos para um objetivo final. Embora os algoritmos utilizados são distintos, a metodologia usada em ambos os trabalhos são muito parecidas.

3 Fundamentação teórica

“Doar sangue é mais que um ato de amor. Trata-se de solidariedade e civismo, coisas que se encontram no sangue do doador”.
Jeferson Calixto

Neste capítulo, são abordados os conceitos necessários para compreensão e realização deste trabalho.

3.1 Doação de sangue

Doação de sangue é o processo onde um doador, por vontade própria, tem seu sangue coletado para armazenamento em um banco de sangue para que posteriormente seja usado em uma transfusão de sangue. Trata-se de um processo de fundamental importância para o funcionamento de um hospital e que pode salvar inúmeras vidas (Ministério da Saúde, 2015).

3.1.1 Requisitos para doação

Conforme informa o Ministério da Saúde (2020), diversos são os critérios que devem ser atendidos pelo candidato no momento da doação para que o ato seja efetivado. Os critérios são:

- Ter idade entre 16 e 69 anos, desde que a primeira doação tenha sido feita até 60 anos (menores de 18 anos devem possuir consentimento formal do responsável legal);
- Pesar no mínimo 50 kg;
- Estar alimentado. Não ingerir alimentos gordurosos antes da doação;
- Ter dormido pelo menos 6 horas nas últimas 24 horas;
- Apresentar documento de identificação com foto emitido por órgão oficial.
- Apresentar pressão arterial dentro da faixa de 180 mmHg (máxima) e 100 mmHg (mínima);
- Estar com pulso com ritmo regular ou com frequência dentro de 50 a 100 batimentos por minuto;
- Não estar em período gestacional;
- Não ter passado por quadros de gripe, resfriado ou febre nos últimos 7 dias;
- Não ter contraído coronavírus (COVID-19) nos últimos 30 dias;
- Não ter vacinado contra febre amarela ou sarampo nas últimas 4 semanas.

Além dos critérios que devem ser seguidos no momento da realização da doação, o Ministério da Saúde (2020) determina impedimentos relacionados à condição de saúde do doador onde impossibilita definitivamente a doação de sangue, sendo eles:

- Ter passado por um quadro de hepatite após os 11 anos de idade;
- Ter evidência clínica ou laboratorial de doenças sexualmente transmissíveis, como hepatite B e C, aids (vírus HIV);
- Ser portador da doença de chagas;
- Ter feito uso de drogas ilícitas injetáveis

3.1.2 Fracionamento

Após o término da doação, o sangue passa por um processo de centrifugação em equipamento especial para coleta dos hemocomponentes que serão utilizados em transfusão. Esses hemocomponentes são concentrado de hemácias, concentrado de plaquetas, plasma fresco congelado e crioprecipitado (HEMOMINAS, 2014).

Segundo definição da Hemominas (2014), concentrado de hemácias é a parte do sangue que contém as hemácias, células sanguíneas responsáveis pelo transporte do oxigênio para todo o corpo humano e que é utilizado em anemias agudas como as causadas por hemorragias. Concentrado de plaquetas é um componente utilizado em caso de alteração da função ou diminuição do número de plaquetas. Plasma fresco congelado e crioprecipitado são as partes utilizadas em sangramento e deficiência de vários fatores de coagulação.

Alguns desses componentes possuem prazo de validade de apenas alguns dias após a coleta, e por esta razão, doadores devem comparecer regularmente. Uma redução no comparecimento impacta rapidamente o estoque desses componentes (HEMOMINAS, 2014).

3.2 Tecnologia e o apoio à saúde

A tecnologia de informação vem se tornando uma das maiores aliadas à área da saúde. Ela serve de apoio a diversos serviços relacionados à saúde, atuando em cumprimento de aspectos legais, automação, qualidade, agilidade e geração de informações de forma eficiente e segura subsidiando a tomada de decisão (AZEVEDO, 2013).

Apoiar a saúde por meio digital (e-Health) é uma nova perspectiva em relação à saúde global. Um dos componentes do e-health é o m-health, que é definida como prática médica e de saúde pública apoiada por dispositivos móveis. Diversas são as aplicações já desenvolvidas para dispositivos móveis visando o suporte à saúde. A doação de sangue é um ato nobre e o uso de aplicativos podem aperfeiçoar esse processo, seja na parte de controle de informações ou na captação de novos usuários (OUHBI et al., 2015).

3.3 Aplicativos móveis

Aplicativos móveis, são aplicações gratuitas ou pagas desenvolvidos exclusivamente para dispositivos móveis como celulares e tablets. Eles se popularizaram a partir do primeiro smartphone, em 2007, com o sistema operacional iOS. Hoje já estão em outros dispositivos como smartTVs e vestíveis. O objetivo deles é disponibilizar funcionalidades de forma fácil e intuitiva além de favorecer a portabilidade (BOCARD, 2021).

Esses aplicativos são conjuntos de algoritmos desenvolvidos utilizando uma ou várias linguagens de programação codificados para realizar uma funcionalidade específica porém o processo de desenvolvimento de aplicativos não se baseia somente em escrever os scripts e publicá-los nas lojas de apps (BOCARD, 2021).

Muito antes de chegar no processo de codificação e publicação do aplicativo é necessário de planejamento. É essencial determinar o escopo de funcionalidades do aplicativo, além de fazer levantamento dos requisitos necessários para implementação e por fim, com base nas informações geradas elaborar uma proposta de valor (BOCARD, 2021).

3.4 Aprendizado de máquina

Aprendizado de Máquina é uma área da Inteligência Artificial onde são desenvolvidas técnicas computacionais de aprendizado bem como sistemas capazes de adquirir conhecimento de forma automática (MONARD; BARANAUSKAS, 2003).

De forma resumida é um método de análise de dados baseada nos princípios do aprendizado indutivo onde algoritmos processam dados conhecidos e gera um modelo capaz de representar os dados sob algum aspecto. Posteriormente, esse modelo pode ser usado para caracterizar novos dados com informações semelhantes as informações dos dados iniciais, de forma mais clara, o modelo gerado é capaz de a partir de dados de entrada, prever o resultado de saída (BRUNIALTI; FRATI; RAVERA, 2015).

O aprendizado de máquina pode ser classificado em supervisionado e não supervisionado. No não supervisionado o indutor analisa os exemplos fornecidos e tenta determinar se alguns deles podem ser agrupados de alguma maneira. Para o caso do supervisionado é fornecido exemplos de treinamento ao algoritmo de aprendizado. Em geral, cada exemplo é descrito por um conjunto de características e o rótulo da classe associada. O objetivo do algoritmo é construir um classificador que possa determinar corretamente a classe de novos exemplos ainda não rotulados. O problema é chamado de classificação quando possui rótulos de classe discretos e regressão para valores contínuos (MONARD; BARANAUSKAS, 2003). A figura 5 retrata a classificação de aprendizado descrita anteriormente.

Figura 5 – Classificação do aprendizado indutivo



Fonte: (MONARD; BARANAUSKAS, 2003)

Embora o aprendizado de máquina seja um método muito eficiente e esteja sendo utilizado cada vez mais, é necessário um estudo de viabilidade de uso desta técnica perante o problema abordado e suas limitações, como por exemplo o número de exemplos disponíveis. Se os exemplos forem insuficientes, ou se os exemplos não forem bem escolhidos, as hipóteses obtidas podem ser de pouco valor ou ainda trazer resultados falsos (MONARD; BARANAUSKAS, 2003).

Rodrigues (2019) diz em seu artigo que em problemas que envolvam classificação para a avaliação do modelo de aprendizado de máquina é preciso fazer o cálculo das métricas alcançadas pelo seu modelo. Essas métricas ajudam a decidir como e em que situações devem ser usadas o modelo escolhido. Essas métricas são as seguintes:

3.4.1 Métricas de Avaliação de modelo de aprendizado de máquina

Para modelos de classificação deve-se fazer uma avaliação dos resultados de predição do treinamento afim de verificar a performance para cada classe, para isso existem algumas métricas que nos dão os resultados para cada classe indicando se o modelo é capaz de fazer a classificação para cada classe ou se existe alguma classe que se sobressai a outra. As métricas mais utilizadas para validação dos modelos são: Matriz de Confusão, precisão, Recall e F1 Score, métricas essas que serão explicadas nas seções a seguir.

3.4.1.1 Matriz de confusão

É uma tabela que mostra quais e quantos foram os erros do modelo enumerando os verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos.

- Verdadeiros Positivos [VP]: classificação correta da classe Positivo;
- Falsos Negativos [FN]: erro em que o modelo previu a classe Negativo quando o valor real era classe Positivo;
- Falsos Positivos [FP]: erro em que o modelo previu a classe Positivo quando o valor real era classe Negativo;
- Verdadeiros Negativos [VN]: classificação correta da classe Negativo.

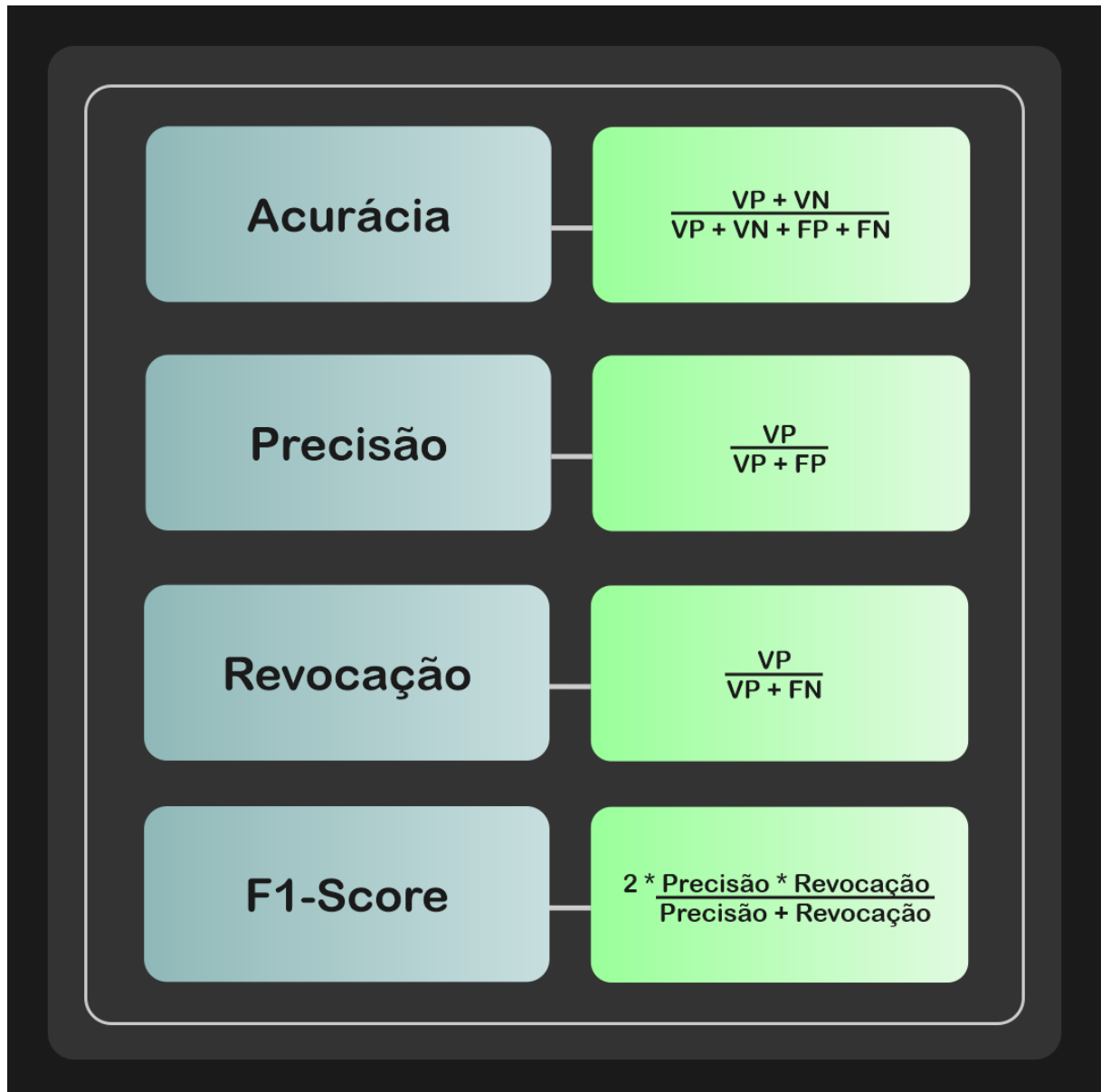
3.4.1.2 Acurácia, Precisão, Recall e F1-Score

Rodrigues (2019) define assim as métricas de avaliação dos modelos de classificação:

- Acurácia: indica uma performance geral do modelo. Dentre todas as classificações, quantas o modelo classificou corretamente
- dentre todas as classificações de classe Positivo que o modelo fez, quantas estão corretas;
- Recall/Revocação/Sensibilidade: dentre todas as situações de classe Positivo como valor esperado, quantas estão corretas;
- F1-Score: média harmônica entre precisão e recall.

As fórmulas de cada métrica podem ser observadas na figura 6.

Figura 6 – Métricas de avaliação dos modelos de aprendizado de máquina



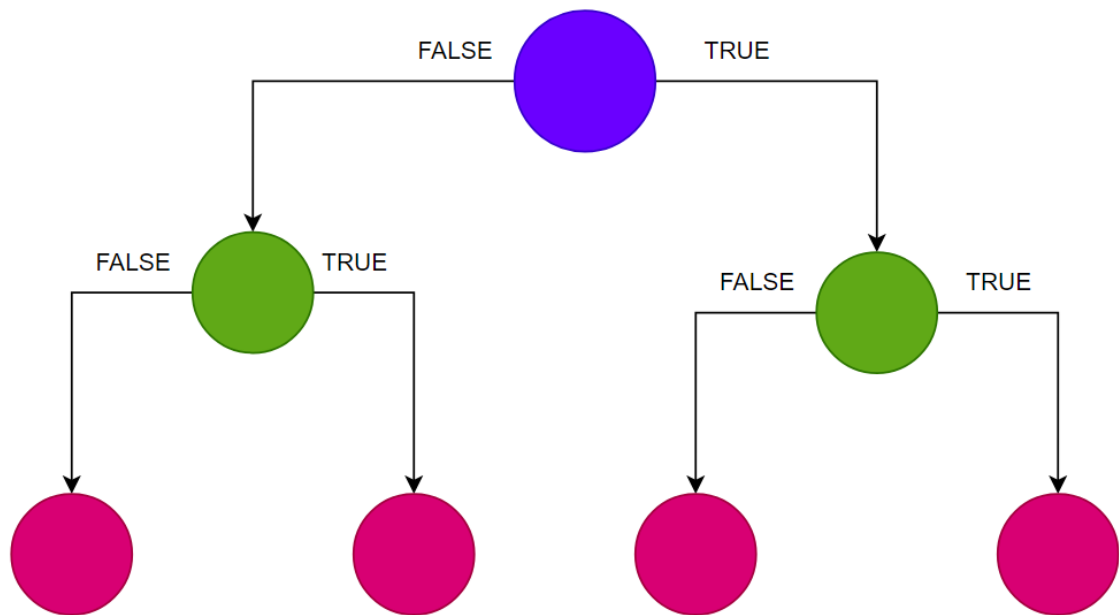
fonte : (RODRIGUES, 2019)

3.5 Decision Tree

Algoritmos do tipo árvore de decisão tem uma estrutura semelhante a uma árvore, isso se deve ao fato desses algoritmos possuírem um nó raiz inicial onde dele é tomada uma decisão do tipo verdadeiro ou falso e a partir dessa resposta são geradas trilhas de decisões a serem tomadas, cada caminho gerado por cada decisão gera um novo nó chamado nó folha, onde são tomadas novas decisões gerando novos caminhos a seguir. A estrutura final se assemelha a uma árvore como o exemplo da figura 7, e por este motivo estes tipo de algoritmo recebem essa nomenclatura.

Árvores de decisão podem ser de dois tipos, Árvore de decisão de variável categórica, onde cada resposta do nó de decisão depende apenas do nó em questão como por exemplo, sim ou não, cara ou coroa, carro ou moto, separando-se em dois lados diferentes e Árvore de decisão de variável contínua, neste tipo de árvore os nós de decisão as respostas não dependem apenas do nó em questão mas da combinação desta com as outras respostas anteriores (MASTERSINDATASCIENCE, s.d).

Figura 7 – Árvore de decisão



fonte : O autor

3.5.1 Extra Trees Classifier

O ExtraTreesClassifier (Extremely Randomized Trees) é um método de aprendizado de máquina baseado em árvores de decisão. Em sua execução, o ExtraTreesClassifier randomiza certas decisões e subconjuntos de dados para minimizar o excesso de aprendizado dos dados e o overfitting, que é quando um modelo se ajusta muito bem ao conjunto de dados anteriormente observado, mas se mostra ineficaz para prever novos resultados (AMORIM, 2019).

Esse algoritmo difere de outros algoritmos baseados em árvore em seu processo de aleatorização e amostragem. Este algoritmo escolhe os pontos de cortes completamente aleatórios, diferente dos demais que utilizam as melhores divisões.(FABIO, 2020)

Para criação do modelo foi implementado um meta estimador que irá estimar através de estatística ajustando várias árvores extras de decisão aleatórias, daí o nome extra-tree, em várias partes de amostras do conjunto de dados usando a média para melhorar a precisão da predição. O extra-tree é uma versão extremamente aleatória do algoritmo Decision Tree (MULLOVERTHING, 2019).

3.6 Python

Matos (2019) traz como informação que segundo pesquisa do StackOverflow a linguagem Python foi a terceira linguagem de programação mais popular no mundo em 2021 e é a linguagem mais utilizada no campo de ciência de dados . E suas qualidades que lhe garantem essa popularidade são muitas, entre elas podemos destacar:

- Comunidade Ativa: A comunidade do Python está em constante crescimento devido a popularidade e ao fato do python ser escolhida por programadores de primeira viagem pela sua maior facilidade de aprendizagem. A comunidade do Python está sempre presente em fóruns, grupos e redes sociais ajudando na resolução de problemas.
- Grande número de bibliotecas: Outra grande vantagem do Python é seu alto número de bibliotecas entre eles estão algumas que usaremos no trabalho: Scikit-Learn, Pandas, Numpy e Jbilib.
- Facilidade de aprendizagem: A estrutura simples do Python facilita a sua aprendizagem por pessoas que nunca tiveram contato com programação por isso é a mais escolhida por programadores iniciantes.
- Jupyter-notebook: Essa ferramenta permite a execução de linhas de código junto com partes textuais o que facilita a organização dos projetos.

3.6.1 Scikit-Learn

A principal biblioteca Python usada para aprendizado de máquina o Scikit-learn fornece uma série de ferramentas como algoritmos, funções estatísticas, funções analíticas que fornecem a base para a criação do nosso modelo de aprendizado de máquina. Kumar (2021) escreve em seu artigo algumas das vantagens da biblioteca Scikit-learn: É uma ferramenta fácil de usar, gratuita, está em constante atualização pela comunidade e bem documentada. Entre os algoritmos que ela possui em seu acervo estão os algoritmos baseados em árvores de decisão como o Extra-Tree Classifier o algoritmo que foi usado para a criação da funcionalidade de predição do nosso aplicativo (TECHOPEDIA, s.d).

3.7 React e React Native

O React surgiu como uma solução para sincronizar a atualização constante de informações na tela, pela estrutura que o HTML, CSS e javascript possuem uma limitação onde elas devem existir com uma certa separação entre elas, por isso a dificuldade na sincronização dos eventos que deixam a solução para este problema muito complexo.

Foi então que um engenheiro do Facebook conhecido como Jordan Walke criou uma solução em que cada acontecimento que as páginas deveriam ter ficassem descritas em cada pedaço resolvendo o problema de sincronia. Foi criado então o React que na época tinha o nome de FaxJSV e inicialmente era utilizado em projetos do Facebook (TARGETTRUST, 2021).

Em 2015 surgiu o Ract Native, um framework que proporciona o desenvolvimento de apps nativos com as soluções do React para desenvolvimento web. O React native utiliza o React para renderizar aplicações Android e IOS.

O React surgiu na necessidade que seus desenvolvedores tiveram de executar o javascript na máquina, e de acordo com o artigo no TargetTrust (2021) estas são as vantagens e desvantagens para desenvolvimento usando o React.

Vantagens:

- Desenvolvimento nativo – se refere à linguagem nativa dos aparelhos. Desta forma, o acesso aos hardwares e softwares é fácil e leve;
- Proteção de dados – por ser um framework de desenvolvimento nativo, possui o poder do hardware em sua totalidade, fornecendo maior segurança para os usuários;
- Alta performance – por conta da programação nativa não ser dividida, os recursos são completamente utilizados, o que deixa o desempenho do aparelho otimizado;

Desvantagens:

- Qualidade inferior – o desenvolvimento de um aplicativo através de uma plataforma cruzada pode comprometer a qualidade, principalmente para aplicativos que executem animações pesadas;
- Desenvolvedor nativo – embora os blocos nativos sejam incorporados ao código React Native, a criação deles precisa de um desenvolvedor iOS ou Android. Sendo assim, pode ser necessário gerenciar três bases de código.

3.7.1 Expo

Em seu artigo, Cunha (2021) descreve o Expo como uma ferramenta essencial no desenvolvimento de aplicativos em React Native pois umas das principais funções é permitir o acesso a várias API's nativas como por exemplo a API de câmera. Outra vantagem é poder programar para IOS sem a necessidade de ter um macOS, ainda que seja necessário um dispositivo com IOS instalado. O Expo está disponível nas lojas tanto do Google a Play Store quanto na da Apple a AppStore. Cunha (2021) escreve ainda as vantagens e desvantagens de usar o Expo no desenvolvimento de aplicativos mobile com React:

Vantagens:

- Facilidade no desenvolvimento de aplicações seja em aparelhos físicos ou emuladores;
- Configuração fácil do ambiente de desenvolvimento;
- Acesso simplificado e fácil as principais funções do Smartphone como notificações, câmeras, mapas, e etc;

Desvantagens:

- Nem todas as API's estão disponíveis para Android e IOS:

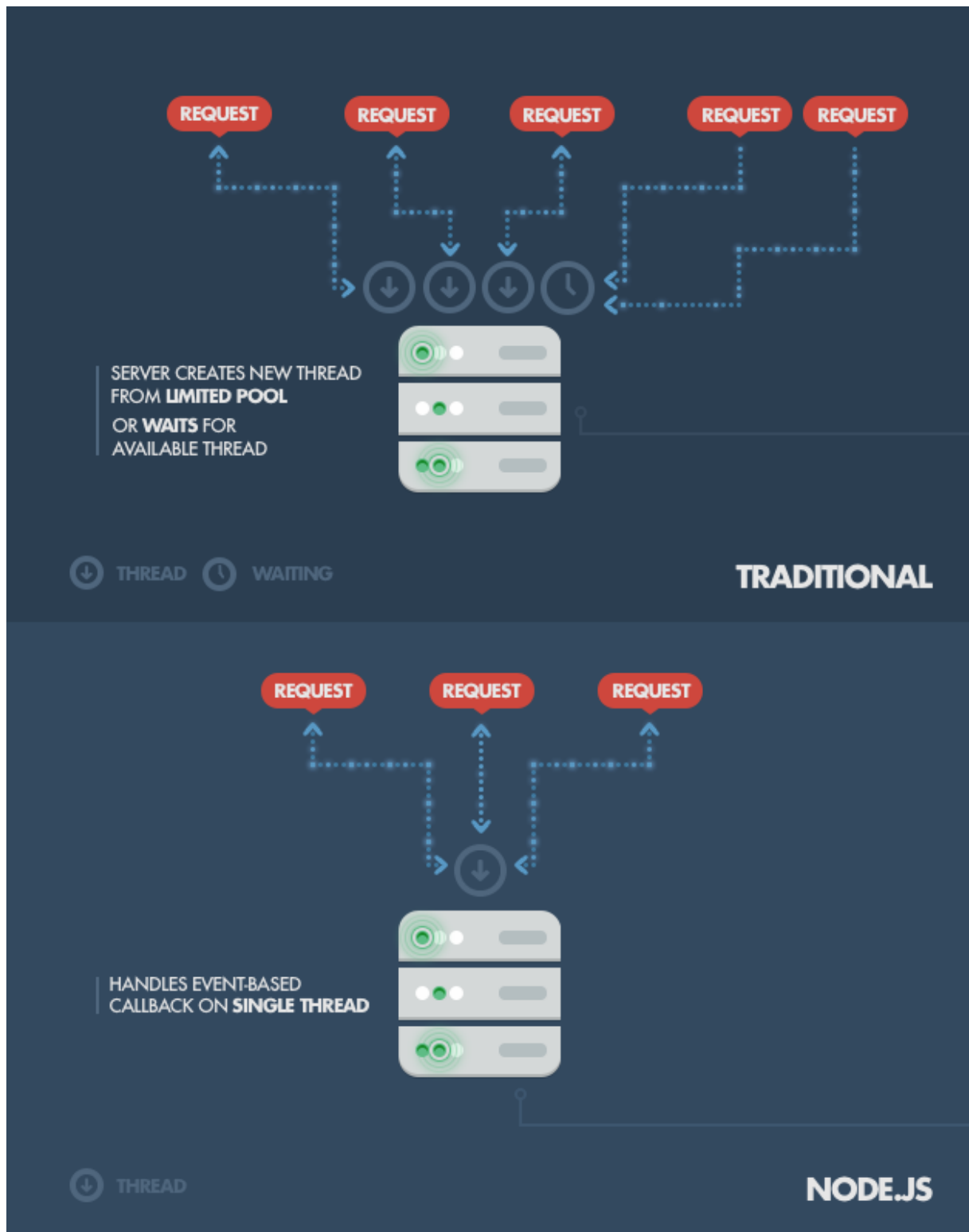
- Atualizações mais recentes do React estão com uma certa demora para a conexão com o Expo;
- Alguns serviços de execução em segundo plano ainda não disponíveis;

3.8 Node JS

De acordo com Capan (s.d) Node.js é uma plataforma criada para desenvolvimento em tempo de execução JavaScript, possibilitando criar aplicativos rápidos e escalonáveis. Node.js usa um modelo de E/S sem bloqueio e orientado a eventos que o torna leve e eficiente. Agora tanto cliente como servidor podem iniciar comunicação e trocar informações e dados livremente.

As técnicas tradicionais de serviços web utilizam um tipo de conexão em que gera um nova thread para cada conexão, ocupando a RAM do sistema e consumindo cada vez mais a quantidade de RAM disponível, enquanto o Node.js gerencia todas as conexões em único thread permitindo múltiplas conexões simultâneas como exemplo do cenários na figura 8.

Figura 8 – Node.Js. Gerenciamento de conexões



fonte : (CAPAN, s.d)

3.8.1 NPM

O Gerenciador de Pacotes do Node NPM(Node Package Manager) é um gerenciador de pacotes do que apresenta um conjunto de soluções reutilizáveis e disponíveis em várias versões. Para utilizar basta fazer a instalação através de repositório online.

Em seu artigo Capan (s.d) traz alguns dos módulos NPM mais importantes disponíveis que são os seguintes:

- Express.js — ou simplesmente Express — uma estrutura de desenvolvimento da Web inspirada em Sinatra para Node.js e o padrão de fato para a maioria dos aplicativos Node.js existentes hoje;
- hapi - uma estrutura centrada na configuração muito modular e simples de usar para criar aplicativos da Web e de serviços;
- connect - Connect é uma estrutura de servidor HTTP extensível para Node.js, fornecendo uma coleção de “plugins” de alto desempenho conhecidos como middleware; serve como base para o Express;
- Socket.io e Sockjs - componente do lado do servidor dos dois componentes de websockets mais comuns disponíveis hoje;
- Pug (anteriormente Jade) - Um dos mecanismos de modelagem populares, inspirado no HAML, um padrão no Express.js.
- Mongodb e Mongojs - wrappers MongoDB para fornecer a API para bancos de dados de objetos MongoDB em Node.js.
- Redis - biblioteca cliente Redis.
- Forever - Provavelmente o utilitário mais comum para garantir que um determinado script de nó seja executado continuamente. Mantém seu processo Node.js em produção em caso de falhas inesperadas.
- Bluebird - Uma implementação completa de Promises/A+ com desempenho excepcionalmente bom.
- Moment - Uma biblioteca de datas JavaScript para analisar, validar, manipular e formatar datas.

3.9 Geolocalização

Geolocalização é um recurso que permite determinar a posição geográfica de um dispositivo com base em um sistema de coordenadas. Mesmo com resultados de pouca precisão, conceitos de geolocalização eram aplicados antes do advento dos mapas quando viajantes usavam as estrelas para identificação de posição (KESTRING, 2014).

O uso de tecnologias de geolocalização teve início na década de 1920, quando os sinais de rádio terrestres calcularam a direção e a distância de navios, porém começou a ganhar forças somente anos mais tarde com a elaboração de estratégias militares. Hoje sabemos que sua utilização foi muito além englobando estratégias de negócios, diversas aplicações fazem uso dessa tecnologia com finalidades de otimizações diferentes (KESTRING, 2014).

Atualmente as principais formas de geolocalização são:

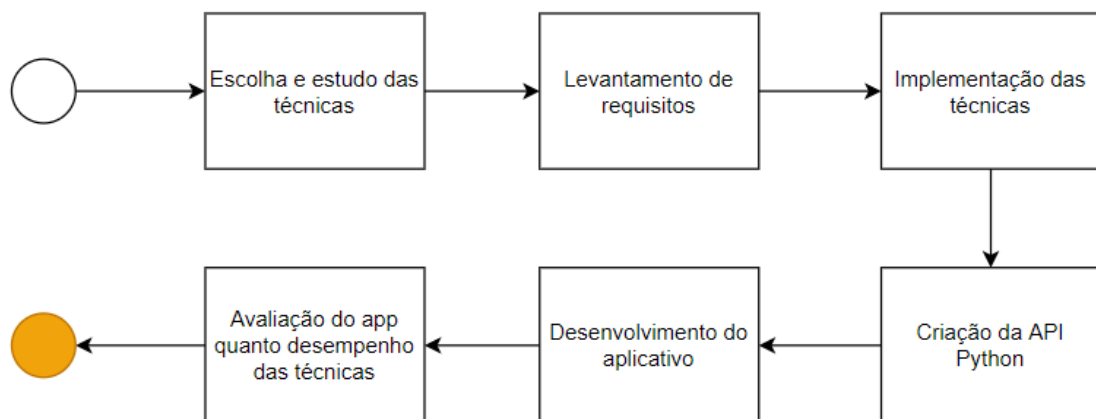
- Wi-fi: A distância é determinada de acordo com a intensidade do sinal, quanto mais longe, menos intenso e vice-versa;

- GPS: Satélites fornecem as coordenadas (latitude e longitude) de um ponto do nosso planeta. Para definir a posição de um objeto, é necessário que o dispositivo capte o sinal de pelo menos três destes satélites, permitindo a triangulação de dados que, após processados, revelam, com grande precisão, a posição do mesmo;
- Radiofrequência: Como o nome sugere, essa forma de geolocalização é pautada na emissão de ondas de rádio;
- AGPS: Evolução do GPS, além dos satélites também são usadas antenas de celulares na geolocalização.

4 Procedimentos metodológicos

Conforme definições de Diana (s.d), este estudo é classificado como pesquisa descritiva e exploratória. O presente capítulo apresenta o procedimento metodológico implantado no projeto com o intuito de alcançar os objetivos propostos. A figura 9 representa todas as etapas seguidas.

Figura 9 – Etapas do trabalho



fonte : O autor

4.1 Escolha e estudo aprofundado das técnicas

Conforme citado em seções anteriores, este trabalho teve como expectativa a criação de uma aplicação com uso de algoritmo de aprendizado de máquina e do recurso de geolocalização de forma a aperfeiçoar o processo busca por doadores de sangue.

Para que fosse possível a implementação de tais técnicas, a primeira etapa a ser realizada foi um estudado aprofundado das mesmas, onde foi possível adquirir o conhecimento necessário para executar essa tarefa. Esta tarefa teve por finalidade também, explorar os projetos relacionados de forma a usá-los com base..

4.2 Levantamento de requisitos

A etapa de levantamento de requisitos foi crucial no processo de desenvolvimento da aplicação. Desde o início do trabalho, levando em consideração o processo de doação de sangue, foram definidas as principais funcionalidades implementadas pelo sistema com a finalidade de trazer um resultado mais impactante na busca por doadores.

Foi elaborado um fluxograma, o qual permite a visualização das funcionalidades abordadas pelo aplicativo de forma a apoiar a implementação do protótipo, fazendo com que tal etapa seja muito mais organizada e correta.

Com as funcionalidades da solução já bem definidas, arquitetadas e documentadas foi feito um levantamento das tecnologias e ferramentas que foram utilizadas para criação do protótipo. Esta escolha foi baseada levando em consideração parâmetros como desempenho, compatibilidade com outros produtos, preço e complexidade de uso.

Os itens que inicialmente serão escolhidos são:

- Linguagens de desenvolvimento e Frameworks;
- Ambiente de desenvolvimento integrado;
- Controlador de versionamento

4.3 Implementação das técnicas

Após uma análise de artigos e trabalhos acadêmicos que tinham como tema a classificação e previsão em classes por aprendizado de máquina foi escolhido o algoritmo ExtraTreesClassifier para a criação do modelo. A escolha deste algoritmo foi dado pela grande capacidade do mesmo para trabalhar com problemas de overfitting, além de se mostrar em vários casos estudados até três vezes mais eficiente em tempo de execução do que outro algoritmo muito conhecido, este sendo o Random Forest.

A finalidade de uso dessa técnica neste projeto foi tentar prever se um determinado indivíduo seria efetivamente um doador no período de uma busca realizada por um receptor e/ou hospital, baseando em informações conhecidas.

Os dados que foram utilizados para treinamento dos modelos é uma base de dados do Centro de Serviços de Transfusão de Sangue da de Hsin-Chu em Taiwan encontrada na UCL Machine Learning Repository. A base é composta por informações de número total de doações, total de sangue doado, meses desde a primeira doação, meses desde a última doação e uma variável binária que representa doação em março de 2007 de 748 indivíduos. Dos 748 indivíduos, 178 realizaram a doação de sangue e 570 não doaram. Embora os dados coletados sejam de indivíduos de outra nacionalidade, espera-se que não exista interferência na classificação, visto que as características usadas para modelagem não são comportamentais e sim informações referentes a doação.

Para implementação da funcionalidade de geolocalização foi utilizado o Expo, a escolha do mesmo pode ser justificada pela facilidade de implementação do mesmo para coleta de informações geográficas além de sua alta precisão. O Expo permite o acesso a APIs nativas do dispositivo e desta forma é possível usar a tecnologia de GPS para identificação de posicionamento geográfico. A finalidade da técnica implementada é buscar os doadores mais próximos aumentando a velocidade de captação de sangue.

4.4 Criação da API Python

Foi desenvolvida uma API, baseada em Python, com o intuito de conseguir utilizar o modelo gerado através do aplicativo que foi desenvolvido.

4.5 Implementação do protótipo

Na implantação do protótipo, como o próprio nome sugere, foi desenvolvido um aplicativo de doação para dispositivos móveis fazendo conexão com as técnicas escolhidas e contemplando as funcionalidades estipuladas nas etapas anteriores.

4.6 Avaliação do protótipo

Após implementação do protótipo, foi realizado uma análise de desempenho do mesmo levantando em consideração os resultados obtidos pelas técnicas escolhidas. Esse passo buscou analisar se o projeto atendeu as expectativas iniciais. A melhor forma de avaliação do aplicativo, seria a realização de testes por hospitais, centros de doações ou profissionais da saúde, porém com a complexidade burocrática desse passo, além do período de coronavírus, essa avaliação se tornou inviável.

5 Desenvolvimento do projeto

Nesta seção se encontra todo processo de desenvolvimento do projeto, desde a definição da estrutura e definição de requisitos funcionais até a etapa final de implementação do aplicativo e validação do mesmo.

5.1 Levantamento de requisitos

A primeira etapa realizada no processo de implementação do aplicativo, foi o levantamento de requisitos funcionais do mesmo. Devido ao tempo disponível para desenvolvimento, foram priorizadas as funcionalidades consideradas essenciais para alcançar os objetivos propostos pelo trabalho. As funcionalidades levantadas foram:

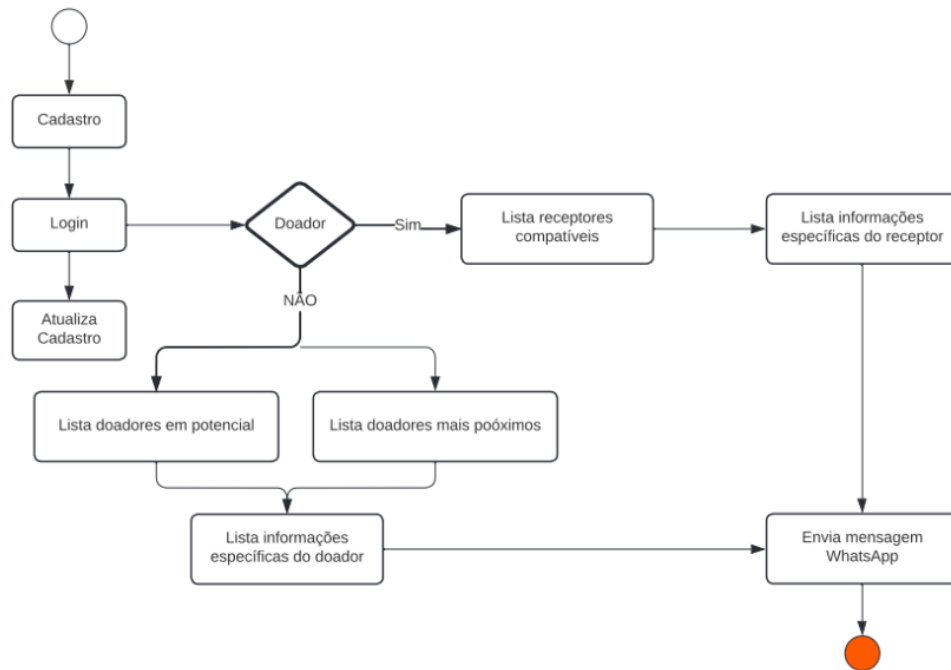
- Cadastro na plataforma;
- Atualização de cadastro;
- Priorização e exibição de doadores em potencial;
- Exibição de distância entre doadores.
- Listagem de doadores mais próximos.
- Envio de mensagem via WhatsApp para doadores.

5.2 Estruturação do aplicativo

Após estar com os requisitos iniciais bem parametrizados, foi realizada uma análise para definição do fluxo de execução do aplicativo.

Foi criado nesta etapa um fluxograma, representando o fluxo a ser seguido pelo usuário como pode-se ver na figura 10.

Figura 10 – Fluxograma



Fonte: O Autor

5.3 Implementação do aplicativo

Com as principais funcionalidades bem estabelecidas e com o fluxo do aplicativo desenhado, foi realizada a implementação da aplicação para dispositivos móveis.

O desenvolvimento do aplicativo basicamente se dividiu em 6 (seis) partes, sendo elas:

- Escolha das tecnologias;
- Treinamento do modelo;
- Criação da API de aprendizado de máquina;
- Implementação do recurso de Geolocalização;
- Desenvolvimento da parte interativa da aplicação;
- Consumo da API de aprendizado e desenvolvimento das funcionalidades;

5.3.1 Escolha das tecnologias e técnicas

Antes de dar início a parte prática da implementação (codificação da aplicação, geração do modelo), foi necessário escolher as tecnologias a serem utilizadas.

Para o desenvolvimento do aplicativo mobile, foram utilizadas as tecnologias React Native, Node.js e Expo. A escolha destas ferramentas se deve a performance que todas proporcionam na implementação de aplicativos, além da alta adesão que possuem no mercado de desenvolvimento de software.

O algoritmo de aprendizado de máquina escolhido foi o `ExtraTreesClassifier`. Este algoritmo minimiza o *overfitting*, sendo capaz de dar boas previsões de novos resultados, sendo ideal para o objetivo levantada neste trabalho.

O treinamento do modelo de aprendizado de máquina e desenvolvimento da API, foi utilizada a linguagem Python uma vez que essa é uma linguagem com muitos recursos de manipulação de dados e extremamente utilizada para treinamento de modelos.

Para implementação da geolocalização, foi utilizado a biblioteca `Location` da ferramenta Expo. Nesse caso, o Expo teve como responsabilidade acessar APIs nativas do dispositivo e desta forma atingir a tecnologia de GPS para identificação de posicionamento. A escolha do Expo foi devido a alta precisão fornecida pelo mesmo, além de sua fácil instalação e implementação.

Toda parte de codificação citada, foi desenvolvido no Visual Studio Code (VS Code), um editor de código de código aberto desenvolvido pela Microsoft. O controle de versão foi realizado através da ferramenta Git.

5.3.2 Treinamento do modelo

Com as tecnologias bem estabelecidas, a primeira etapa foi a geração do modelo de aprendizado de máquina que foi usado na implementação do aplicativo para classificação de possíveis doadores.

O treinamento do modelo foi realizado no Jupyter Notebook e foi utilizado a biblioteca `Scikit-learn`. Para construção do modelo foi seguido o passo a passo explicado nos parágrafos a seguir.

O primeiro passo para treinamento foi realizar a leitura da base. A estrutura era composta por informações de meses desde a última doação, quantidade de vezes doadas, quantidade de sangue doado, tempo desde a primeira doação e se o candidato doou em uma data específica (março de 2007). Tal estrutura é mostrada na figura 11.

Figura 11 – Estrutura da base de dados

	Recency (months)	Frequency (times)	Monetary (c.c. blood)	Time (months)	whether he/she donated blood in March 2007
0	2	50	12500	98	1
1	0	13	3250	28	1
2	1	16	4000	35	1
3	2	20	5000	45	1
4	1	24	6000	77	0

Fonte: O Autor

Em seguida, para que o modelo fosse baseado em dados sólidos, foi realizado uma limpeza da base, retirando informações referentes a doadores que possuíam dados faltantes.

Com a base bem estruturada, foi necessário a separação da mesma em 2 (duas) novas bases, base para treino e base para teste. Nesta etapa, com intuito de um melhor resultado a

separação das bases foi realizada primeiro com apenas 30% dos dados para teste e posteriormente 20%. A separação foi realizada utilizando a função "train_test_split()" do Scikit-learn. Por padrão, a divisão da base de treino e teste pelo train_test_split é feita aleatoriamente, por este motivo, foi realizado o treinamento do modelo 5 (cinco) vezes para cada separação de base e coletada a acurácia, o recall e a precisão de cada modelo gerado como se pode observar nas tabelas 1 e 2.

Tabela 1 – Comparação da separação de base de 30%

Teste	Métricas Classe 0	Métricas Classe 1
1	Acurácia: 0.73	
	Precision: 0.81	Precision: 0.45
	Recall: 0.90	Recall: 0.27
	F1 Score: 0.85	F1 Score: 0.34
2	Acurácia: 0.73	
	Precision: 0.82	Precision: 0.33
	Recall: 0.85	Recall: 0.28
	F1 Score: 0.84	F1 Score: 0.30
3	Acurácia: 0.76	
	Precision: 0.86	Precision: 0.34
	Recall: 0.85	Recall: 0.37
	F1 Score: 0.85	F1 Score: 0.36
4	Acurácia: 0.75	
	Precision: 0.83	Precision: 0.40
	Recall: 0.85	Recall: 0.38
	F1 Score: 0.84	F1 Score: 0.39
5	Acurácia: 0.78	
	Precision: 0.85	Precision: 0.46
	Recall: 0.88	Recall: 0.39
	F1 Score: 0.86	F1 Score: 0.42

Fonte: O autor

Tabela 2 – Comparação da separação de base de 20%

Teste	Métricas Classe 0	Métricas Classe 1
1	Acurácia: 0.72	
	Precision: 0.78	Precision: 0.48
	Recall: 0.85	Recall: 0.37
	F1 Score: 0.82	F1 Score: 0.42
2	Acurácia: 0.71	
	Precision: 0.82	Precision: 0.28
	Recall: 0.82	Recall: 0.28
	F1 Score: 0.82	F1 Score: 0.27
3	Acurácia: 0.76	
	Precision: 0.86	Precision: 0.34
	Recall: 0.86	Recall: 0.37
	F1 Score: 0.85	F1 Score: 0.36
4	Acurácia: 0.75	
	Precision: 0.83	Precision: 0.40
	Recall: 0.85	Recall: 0.38
	F1 Score: 0.84	F1 Score: 0.39
5	Acurácia: 0.78	
	Precision: 0.85	Precision: 0.46
	Recall: 0.88	Recall: 0.39
	F1 Score: 0.86	F1 Score: 0.42

Fonte: O autor

Com intuito de comparar os resultados obtidos nos casos de teste de cada separação de base, foi calculada a média e o desvio padrão de cada caso como pode-se observar nas tabelas 3 e 4.

Tabela 3 – Médias das métricas da base 30%

Classes	Métricas			
	Acurácia	Precisão	Recall	F1 Score
Classe 0	0.75 ±0.02	0.834 ±0.021	0.866±0.023	0.848±0.008
Classe 1		0.396 ±0.060	0.338±0.058	0.362±0.046

Fonte: O autor

Tabela 4 – Médias das métricas da base 20%

Classes	Métricas			
	Acurácia	Precisão	Recall	F1 Score
Classe 0	0.744±0.028	0.828±0.031	0.852±0.021	0.838±0.017
Classe 1		0.392±0.083	0.359±0.039	0.372±0.062

Fonte: O autor

Ainda com a intuito de obter um melhor resultado, visto que através da análise da média e do desvio padrão não foi possível decidir uma melhor separação de base, foi passado o parâmetro shuffle com o valor falso para o traintestsplint. Este parâmetro com valor falso indica que a aleatoriedade na separação de bases deve ser retirada. Novamente foi gerado o modelo para cada separação de base e foram coletadas as respectivas métricas que podem ser visualizadas na tabela 5.

Tabela 5 – Comparação da separação de base sem aleatoriedade

Base (%)	Métricas Classe 0	Métricas Classe 1
20 %	Acurácia: 0.84	
	Precision: 0.90	Precision: 0.0
	Recall: 0.93	Recall: 0.0
	F1 Score: 0.91	F1 Score: 0.0
30 %	Acurácia: 0.73	
	Precision: 0.81	Precision: 0.21
	Recall: 0.87	Recall: 0.14
	F1 Score: 0.84	F1 Score: 0.17

Fonte: O autor

Como os valores foram semelhantes em ambas as separações de bases. foi dado continuidade ao treinamento seguindo a separação de 70% da base para treino e 30% para teste.

Neste momento também foi realizado a análise de quais variáveis impactavam mais no modelo e gerado um gráfico de barras contendo os resultados, como pode ser observado na figura 12.

Figura 12 – Importância das variáveis no modelo



Fonte: O Autor

Com o modelo treinado, para que fosse possível analisar o funcionamento do mesmo, foi realizada uma previsão de resultados sobre a base de testes. A tabela 6 retrata uma fração de dados de saída da base de testes e a previsão do modelo referente aos casos de testes separados.

Tabela 6 – Previsão do modelo

Teste	Variável de saída da base de teste	Previsão do Modelo
0	0	0
1	1	1
2	0	0
3	1	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	1	1

Fonte: O autor

Por fim, para que não fosse necessário realizar novo treinamento de modelo e pudesse utilizar o modelo na aplicação, o mesmo foi persistido em disco.

5.3.3 Implementação da Geolocalização

Para atender o objetivo proposto pelo trabalho de sinalizar a distância entre doador e receptor, foi necessário a implementação do recurso de localização.

Esta implementação foi baseada no uso da ferramenta Expo. Para que fosse possível obter as informações de localização através do Expo foi necessário seguir alguns passos fornecidas em documentação do mesmo.

Primeiro foi necessário realizar a instalação de dois pacotes. O primeiro pacote foi o Location, responsável pela extração de dados relacionados a localização. O segundo foi o permissions, voltado para questões de segurança. Para instalação dos mesmos, foram executados os comandos visualizados na figura 13

Figura 13 – Instalação Pacotes Expo

```
expo install-location / expo install expo-permissions
```

Fonte: O Autor

Também foi necessário a importação dos pacotes para serem utilizados posteriormente.

Figura 14 – Importação Pacotes Expo

```
import * as Location from 'expo-location';  
import * as Permissions from 'expo-permissions';
```

Fonte: O Autor

Por fim, com os pacotes instalados e importados, foi realizada a implementação da função getLocationAsync encontrada na figura 15

Figura 15 – Implementação função getLocationAsync

```
getLocationAsync = async () => {  
  let { status } = await Permissions.askAsync(Permissions.LOCATION_BACKGROUND);  
  if (status !== 'granted') {  
    this.setState({  
      errorMessage: 'Permission to access location was denied',  
    });  
  }  
  
  let location = await Location.getCurrentPositionAsync({accuracy: Location.Accuracy.Highest});  
  const { latitude, longitude } = location.coords  
  this.getGeocodeAsync({latitude, longitude})  
  this.setState({ location: {latitude, longitude}});  
  
  fetch('http://192.XXX.X.XXX:3000/update',{  
    method: 'POST',  
    headers: { 'Content-Type': 'application/json' },  
    body: JSON.stringify({  
      user: route.params.user,  
      latitude: latitude,  
      longitude: longitude  
    })  
  });  
};
```

Fonte: O Autor

Esta função é executada quando o usuário realiza login na ferramenta. Em sua implementação, primeiro é verificada a permissão de acesso ao recurso de localização.

Em seguida através da função `getCurrentPositionAsync` do `expo-location` é obtida a localização. Esta função leva como argumento o valor de precisão esperada. No caso do projeto desenvolvimento, foi utilizado `Accuracy Highest` que é o melhor nível de precisão disponível.

Após coleta das informações de localização é executada a função `getGeocodeAsync`. Esta função é responsável por extrair os dados de latitude e longitude.

Por fim, é realizada uma requisição com a finalidade de realizar uma atualização nos dados de localização do usuário no banco de dados para posteriormente ser possível realizar cálculos de distância com dados recentes.

5.3.4 Implementação da API de predição

Para que fosse possível consumir o modelo gerado no Jupyter Notebook, foi criada uma API. As tecnologias utilizadas para o desenvolvimento da API foram a linguagem Python e o Micro-Framework Flask.

A escolha do Python se deve ao fato de ser uma linguagem com grande poder de manipulação de dados devido ao grande número de bibliotecas voltadas para este objetivo. Já a escolha do Flask se deve a facilidade de criar um servidor simples e sua alta curva de aprendizado.

Na figura 16, é possível evidenciar o principal trecho de código da API para este projeto.

Figura 16 – Código de implementação da API

```
import os
from flask import request, Flask
import joblib
import numpy as np

app = Flask(__name__)

model = joblib.load('model.pkl')

@app.route('/predicao', methods=['POST'])
def predicao():
    valores = request.get_json()
    teste = np.array([[valores.get('variavel1'),
                        valores.get('variavel2'),
                        valores.get('variavel3'),
                        valores.get('variavel4')]])
    classe = model.predict(teste)[0]

    if classe == 1:
        return "1"
    else:
        return "0"

@app.route("/")
def index():
    return "<h1>Servidor Funcionando</h1>"

if __name__ == "__main__":
    from waitress import serve
    serve(app, host="0.0.0.0", port=8080)
```

Fonte: O Autor

O trecho retratado na figura, foi constituído por duas funções, sendo elas: predição e index. A função index tem com finalidade única verificar se o servidor esta em execução. Já a

função predição, é responsável por receber através de uma requisição POST, as informações de doações de um determinado usuário, realizar uma predição utilizando o modelo gerado e retornar o valor "1" se o usuário for classificado como possível doador ou "0" caso o mesmo seja classificado como não doador.

5.3.5 Implementação do protótipo

Após finalização da criação da API de predição foi dado início a etapa de implementação do protótipo.

Como principais pontos desta etapa, é possível destacar o desenvolvimento das telas da aplicação, a criação da lógica de exibição de possíveis doadores, a comunicação com API do WhatsApp para envio de mensagens para os doadores e a lógica exibição da distância entre os usuários

As telas foram criadas procurando seguir simplicidade e facilidade de utilização pelos usuários com a finalidade de atingir uma maior quantidade de possíveis doadores.

A lógica para exibição de possíveis doadores se baseia na montagem de uma lista priorizando os doadores considerados aptos para doação no momento em que o usuário manipulador da aplicação acessar a tela de possíveis doadores aptos. Para classificação do doador, foi consumida a API de predição criada anteriormente. A comunicação com tal API, foi realizada através de uma requisição do tipo POST onde foram passados no corpo da mesma, os parâmetros de doação do usuário como é possível analisar na figura 17

Figura 17 – Requisição para API de predição

```
let response = await fetch('http://192.xxx.x.xxx:8080/predicao',{
  method: 'POST',
  headers:{
    Accept: 'application/json',
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    variavel: json[i].Recency,
    variavel2: json[i].Frequency,
    variavel3: json[i].Monetary,
    variavel4: json[i].Time
  })
});
```

Fonte: O Autor

Foi implementado também nesta etapa, uma lógica que quando o usuário acessa a tela de "doadores próximos", é exibida a distância entre os usuários. Para realização de tal tarefa, foi

usada uma fórmula de distância utilizando as informações de latitude e longitude coletadas e salvas em banco anteriormente.

Por fim, para que fosse possível o usuário entrar em contato com o possível doador, foi implementado a lógica de envio por WhatsApp. Esta lógica se baseia na geração de um link de WhatsApp para o número do doador salvo no banco anteriormente.

A geração do link foi realizada através de uma requisição para API disponibilizada pelo WhatsApp. Para utilização da API, foi realizado um estudo da mesma e foi possível identificar o endpoint Send da API o qual é responsável pelo envio de mensagem e leva como parâmetros o número que o link será direcionado e a mensagem a ser enviada. Para realização da requisição foi implementada a função linkingWhats. Como podemos visualizar na figura 18, para trabalhar com o link do WhatsApp foi usada a função openURL contida no linking do React-Native.

Figura 18 – Função linkingWhats

```
function linkingWhats(){  
  Linking.openURL('https://api.whatsapp.com/send?phone=55319950XXXX&text="Bom dia prezado, tudo bem?"')  
}
```

Fonte: O Autor

6 Resultados

Nesta seção se encontra os resultados obtidos no desenvolvimento do projeto. Os resultados são baseados na implementação dos recursos de aprendizado de máquina, geolocalização e no desenvolvimento do protótipo, os quais são objetivos do trabalho.

6.1 Implementação dos recursos

O recurso de aprendizado de máquina foi criado a partir da geração do modelo baseado no algoritmo Extra Trees Classifier. O modelo obteve uma acurácia de 0.75 e para as métricas de Recall, Precisão e F1 Score foram encontrados os seguintes valores para as classes 0 e 1, estas sendo respectivamente, não doadores e doadores:

Classe 0:

- Precisão: 0.83.
- Recall: 0.86.
- F1 Score: 0.84.

Classe 1:

- Precisão: 0.39.
- Recall: 0.33.
- F1 Score: 0.36.

De acordo com as métricas coletadas, foi possível perceber que para classe de não doadores, o algoritmo teve um desempenho dentro do esperado para este trabalho. Para a classe de doadores, a qual foi caracterizada como mais importante, o algoritmo teve um desempenho relativamente baixo. Este resultado está relacionado ao desbalanceamento da base utilizada visto que a mesma possui uma quantidade de casos de não doadores muito maior do que de doadores.

6.2 Implementação do aplicativo

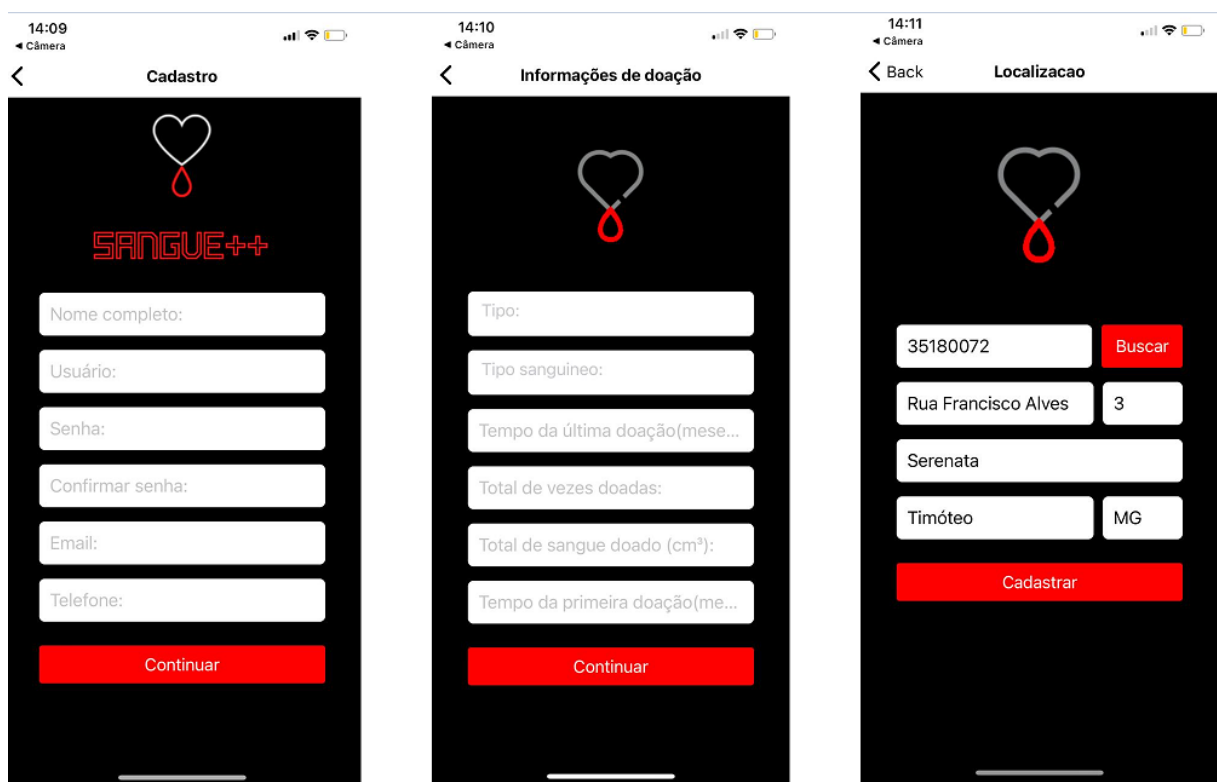
Conforme estabelecido nos objetivos, foi desenvolvido um protótipo de aplicativo para localização de doadores. O protótipo construído seguiu o modelo de fluxograma gerado anteriormente. O app desenvolvido é constituído de oito telas sendo elas:

- Três telas de cadastro;

- Uma tela de login;
- Uma tela para atualização de perfil;
- Uma tela buscar usuários mais próximos.
- Uma tela para buscar usuários classificados como possíveis doadores.
- Uma tela com detalhamento de dados do usuário buscado e comunicação com o mesmo via rede social.

A parte de cadastro implementada no aplicativo ficou subdividida em três telas como retratado na figura 19. A primeira tela é responsável pela coleta de informações pessoais do usuário, sendo elas: nome completo, usuário de acesso, senha, telefone de contato e email. Dando seguimento ao fluxo de cadastro, a segunda tela tem a responsabilidade de coletar os dados de doação do usuário que serão usadas para classificação do usuário como apto ou não através do modelo gerado. Por fim, a última tela de cadastro é responsável por fazer uma requisição para a API do ViaCep através do código de endereçamento postal fornecido pelo usuário para trazer o endereço do mesmo. Ainda nesta tela, ao clicar no botão "cadastrar", um novo usuário é salvo no banco de dados.

Figura 19 – Telas de cadastro

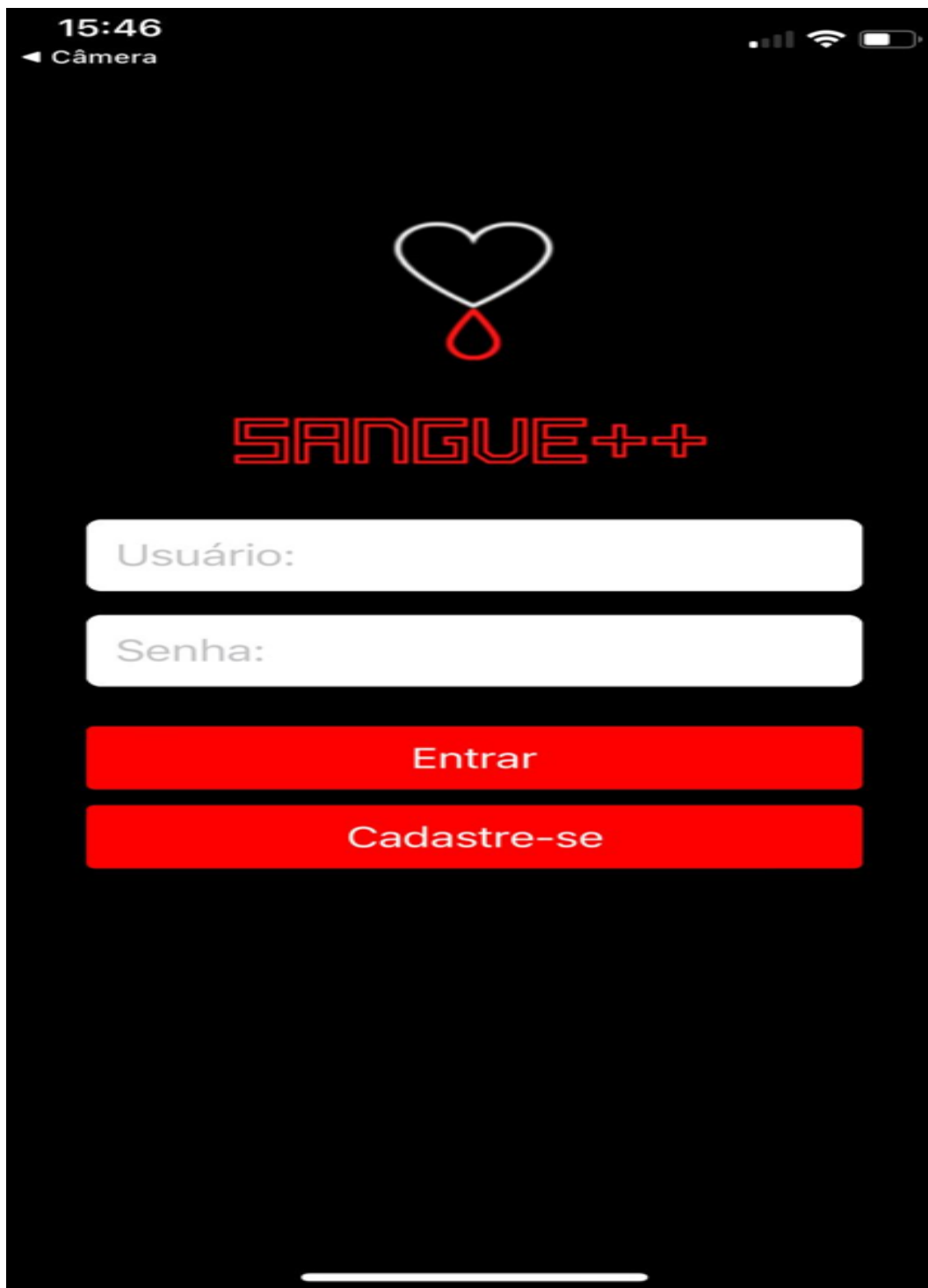


Fonte: O Autor

Foi desenvolvida a tela de login, na qual quando o usuário já está cadastrado no sistema é realizada uma validação dos dados informados e caso o resultado da validação seja verdadeiro

o usuário é redirecionado a tela principal do sistema. A figura 20 se refere a tela desenvolvida para login.

Figura 20 – Tela de login

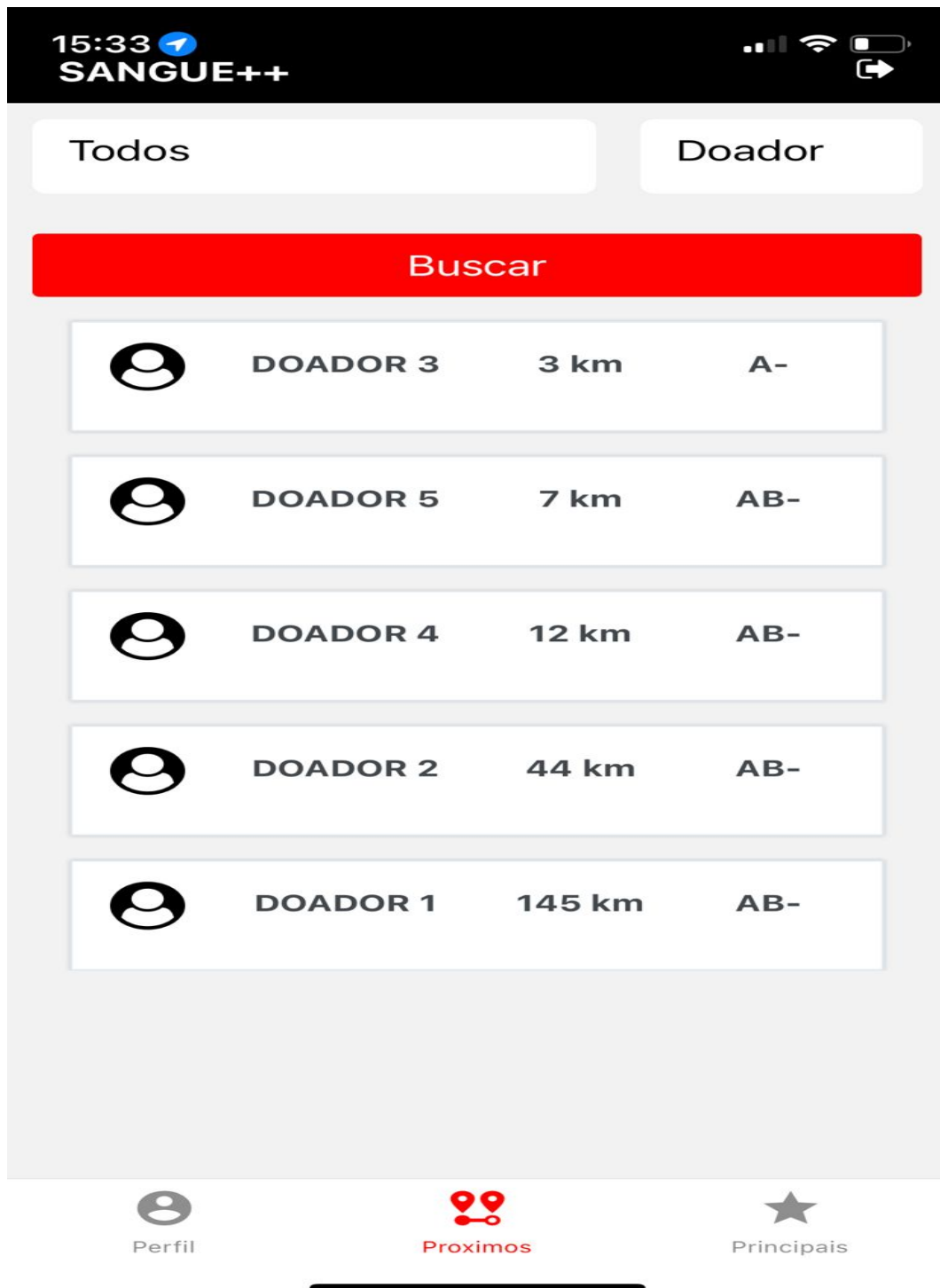


Fonte: O Autor

Ao realizar login na plataforma, o usuário se depara com três telas. A primeira tela é responsável pelas alterações de cadastro do usuário como senha e última doação.

Como pode-se observar na figura 21, a segunda tela é responsável por exibir os usuários e ordena-los pela distância. Nesta tela também é possível realizar a filtragem dos usuários pelo tipo sanguíneo e pelo tipo de usuário sendo este último classificado entre doador e receptor.

Figura 21 – Tela de listagem de doadores por distância



Fonte: O Autor

Conforme informado pela tabela 7, com intuito de avaliação da ferramenta através da per-

formance da técnica de geolocalização implementada, foi comparado a distância em linha reta entre um usuário e cinco doadores de teste via aplicativo e via Google Maps. Visto que a aplicação trabalha com sistema de arredondamento, os resultados obtidos retrataram com exatidão os casos de estudo.

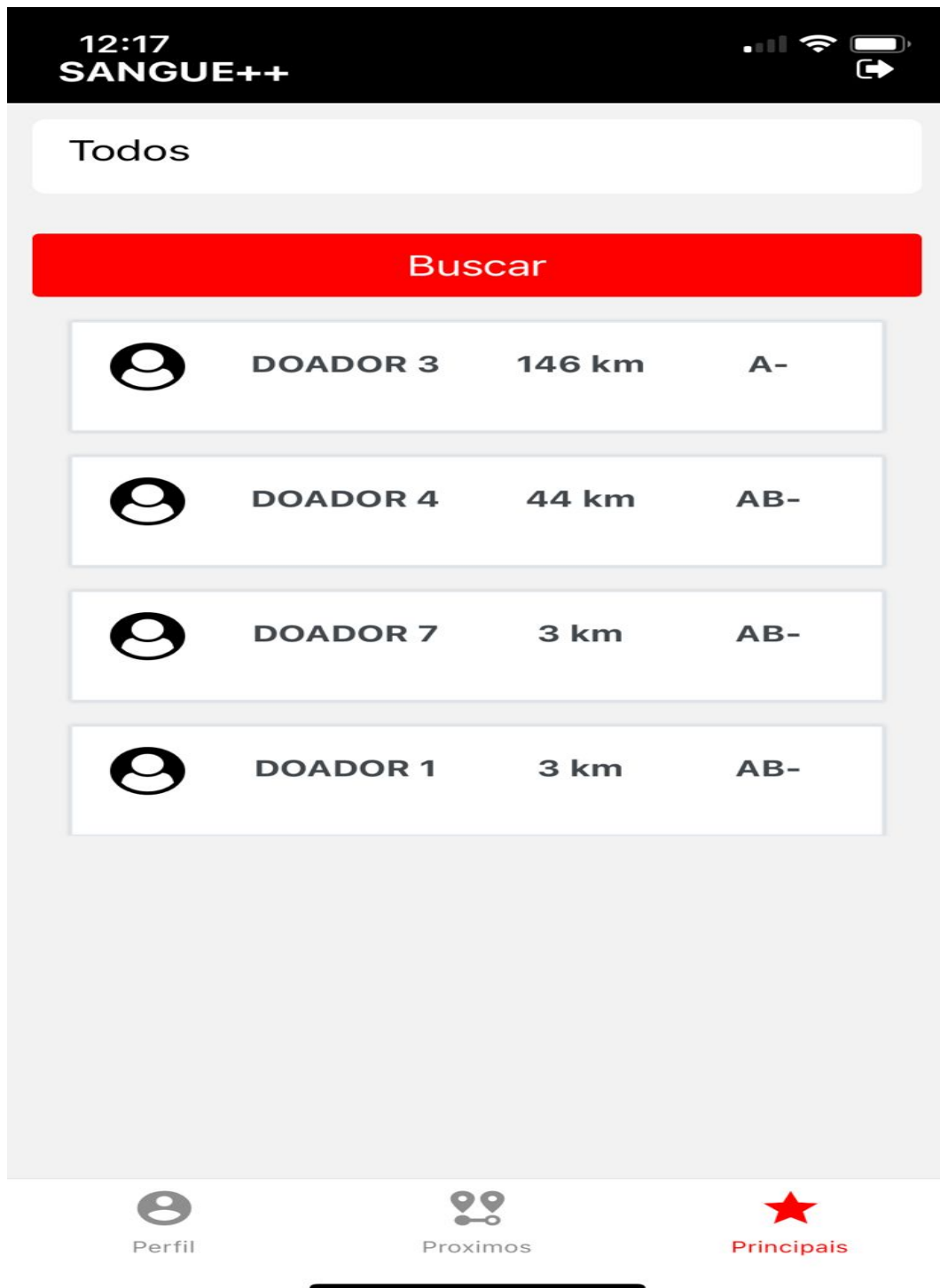
Tabela 7 – Avaliação da aplicação - Técnica de geolocalização

Nome do doador	Localização do doador	Localização do usuário	Distância Real	Distância no App
Doador 1	Shopping Del Rey Belo Horizonte - MG	Bairro: Arataquinha Timóteo - MG	145,34 km	145 km
Doador 2	Bairro: Centro Bom Jesus do Galho - MG	Bairro: Arataquinha Timóteo - MG	44,77 km	44 km
Doador 3	Cefet Timóteo - MG	Bairro: Arataquinha Timóteo - MG	3,16 km	3 km
Doador 4	Bairro: Novo Cruzeiro Ipatinga - MG	Bairro: Arataquinha Timóteo - MG	12,16 km	12 km
Doador 5	Bairro: Petrópolis Timóteo - MG	Bairro: Arataquinha Timóteo - MG	7,36 km	7 km

Fonte: O autor

A terceira tela que pode ser observada na figura 22, ficou responsável pela parte de aprendizado de máquina na aplicação. Nesta tela o usuário listava os demais usuários cadastrados no aplicativo e que foram classificados como possíveis doadores além de poder realizar a filtragem por tipo sanguíneo.

Figura 22 – Tela de listagem de possíveis doadores



Fonte: O Autor

Embora as métricas de acurácia, recall e precisão coletadas na etapa de desenvolvimento deste trabalho representem uma boa análise de performance da técnica de aprendizado de máquina implementada, para avaliação deste recurso implantado no aplicativo, foi sorteado dez casos de teste, onde os usuários foram cadastrados com as suas respectivas informações e foram classificados pelo modelo. Conforme pode ser visto na tabela 8, dos dez casos disponi-

bilizados para aplicação, a mesma acertou a previsão de nove.

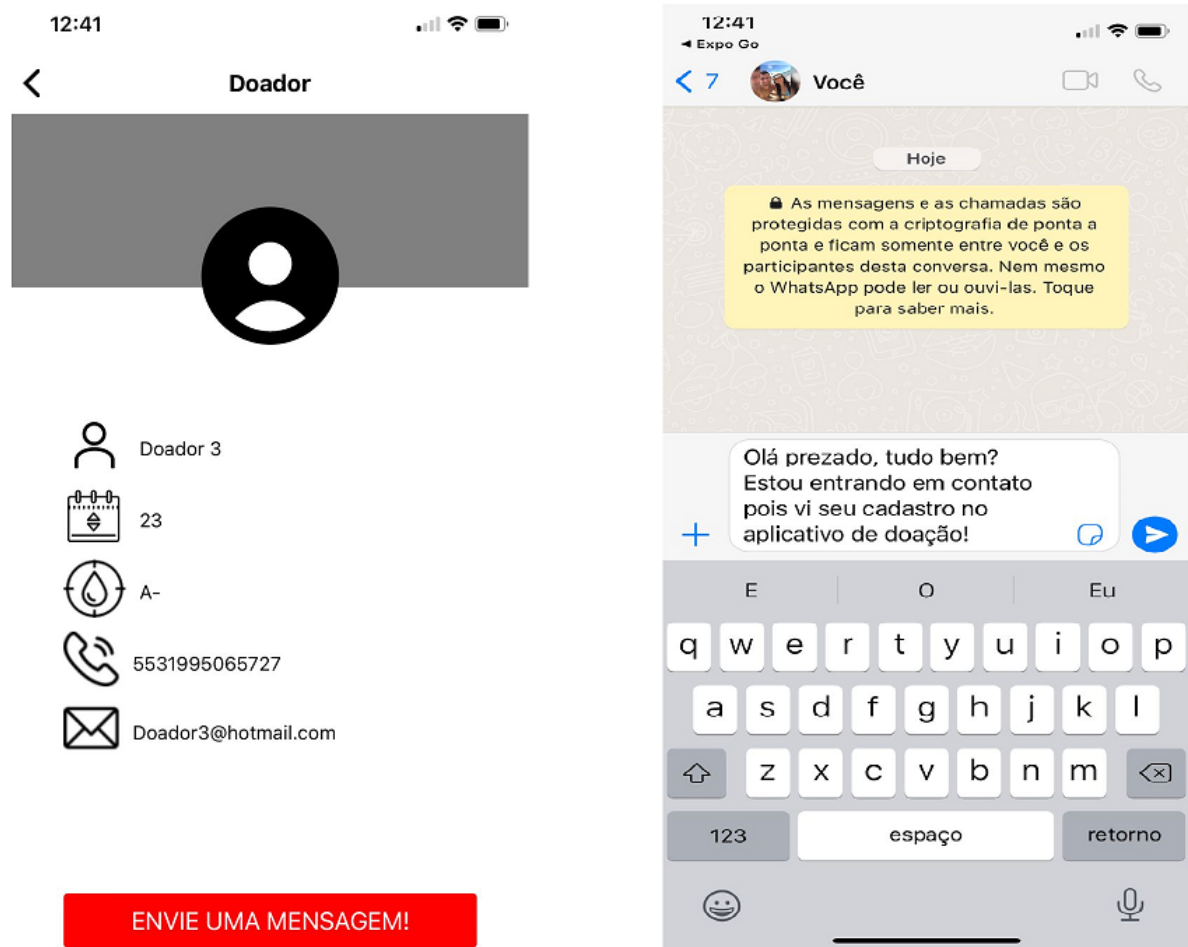
Tabela 8 – Avaliação da aplicação - Técnica de aprendizado de máquina

Doador	Meses desde a última doação	Total de doações	Total de sangue doado (cm ³)	Meses desde a primeira doação	Classe	Previsão no App
1	2	50	12500	98	1	1
2	0	13	3250	28	1	0
3	1	16	4000	35	1	1
4	2	20	5000	45	1	1
5	1	24	6000	77	0	0
6	4	4	1000	4	0	0
7	2	7	1750	14	1	1
8	1	12	3000	35	0	0
9	4	23	5750	58	0	0
10	0	3	750	4	0	0

Fonte: O autor

Por fim, foi desenvolvida uma tela visualiza na figura 23, onde é possível visualizar as informações detalhadas de um doador específico e também entrar em contato com o mesmo via rede social.

Figura 23 – Detalhes doador e envio de mensagem WhatsApp



Fonte: O Autor

7 Limitações e trabalhos futuros

7.1 Limitações

Neste trabalho, as principais limitações encontradas foram relacionadas ao deploy da ferramenta e a base de dados utilizada.

A base de dados encontrada e que foi utilizada como sustentação para o trabalho e para treinamento do modelo era formada por poucos casos de teste, poucas variáveis e era de nacionalidade taiwanesa. Levando em consideração as características da base, o treinamento do modelo ficou limitado.

Outro ponto de limitação foi o deploy da ferramenta. O cenário perfeito seria a liberação da ferramenta para que já começasse a refletir efeitos dentro da sociedade, porém, esta liberação não foi possível neste trabalho devido a alguns pontos, sendo eles:

- Custo para liberação da ferramenta;
- Tempo para execução da tarefa;
- Falta de mão de obra para manutenção e acompanhamento da ferramenta;
- Segurança da aplicação;

7.2 Trabalhos futuros

Levando em considerações os pontos que não foram realizados neste trabalho, pode-se levantar os seguintes trabalhos:

- Deploy da ferramenta;
- Treinamento do modelo com uma base de dados mais completa;
- Treinamento do modelo com novas variáveis coletadas;
- Geração do aplicativo com outros algoritmos de Machine Learning e com novos métodos de Geolocalização;
- Desenvolvimento de triagem de doadores na ferramenta.

O trabalho mais relevante seria o deploy da ferramenta, ou seja, liberar a ferramenta para o público em massa com a finalidade de explorar a potencialidade da aplicação para um bem social.

Com a liberação da aplicação, conseqüentemente seria gerada uma base bem mais robusta e com dados brasileiros. Desta forma poderia ser realizado um novo treinamento de forma a procurar gerar um modelo mais assertivo e portanto uma ferramenta mais confiável.

Ainda por consequência da liberação da ferramenta, a base de doadores seria gerada com informações adicionais como idade e sexo, desta forma poderia ser realizado novo treinamento procurando entender o impacto de tais variáveis e desta forma fazer um aperfeiçoamento no modelo.

Por fim, ainda objetivando uma melhor performance do aplicativo, o mesmo poderia ser implementado com outros algoritmos de aprendizado de máquina e com outros recursos de Geolocalização.

8 Considerações finais

Este trabalho teve como objetivo o desenvolvimento de uma ferramenta para aprimoramento da busca de doadores principalmente através da implementação dos recursos de geolocalização e aprendizado de máquina. A ferramenta foi construída seguindo os requisitos preestabelecidos e portanto os objetivos foram alcançados com sucesso.

Apesar do modelo de aprendizado de máquina utilizando o algoritmo Extra tree e a base encontrada na UCL Machine Repository não obteve a performance esperada para classe 1, visto que as métricas de recall e previsão, que medem os acertos de casos positivos, ou seja, doadores aptos neste caso de estudo, teve uma taxa baixa, o aplicativo foi desenvolvido de forma a consumir facilmente outros modelos de aprendizado, além de poder ser treinado novamente com uma base mais populosa. Desta forma, esta ferramenta pode servir como base em trabalhos futuros com novas bases e algoritmos de forma a alcançar resultados mais assertivos na previsão de doadores.

O recurso de geolocalização teve uma performance considerada muito boa dentro das expectativas deste trabalho e atrelada a comunicação via rede social pode vir a ser uma boa estratégia para localização rápida e eficiente de doadores.

De modo geral, a aplicação alcançou as expectativas criadas no início deste trabalho e para trabalhos futuros sugere-se o treinamento de novos modelos, com novas variáveis coletadas e com outros algoritmos, mas acima de tudo, que seja realizado o deploy da ferramenta, de forma a explorar ainda mais o potencial da solução criada gerando assim, impactos no sistema de saúde do Vale do Aço.

Referências

- AMORIM, B. *Uso de Aprendizado de Máquina para Classificação de Risco de Acidentes em Rodovias*. Agosto 2019, Agosto 2019. Citado nas páginas 18, 19 e 27.
- AZEVEDO, E. Y. F. d. *O papel estratégico da tecnologia de informação na área da saúde*. 2013, 2013. Citado na página 22.
- BARRA, B. L. M. d. P. *Fatores que levam o doador voluntário de sangue a realizar a doação*. 2015, 2015. Citado na página 13.
- BOCARD, T. *O que são aplicativos? Definição da desenvolvedora Usemobile*. 2021. Disponível em: <<https://usemobile.com.br/aplicativo-movel/>>. Citado na página 23.
- BRUNIALTI, G.; FRATI, L.; RAVERA, S. Structural variables drive the distribution of the sensitive lichen lobaria pulmonaria in mediterranean old-growth forests. *Ecological Indicators*, v. 53, p. 37–42, 01 2015. Citado na página 23.
- CAPAN, T. Why the hell would i use node.js? a case-by-case tutorial. *Toptal Blog*, s.d. Acessado dia 23 de Julho de 2022. Disponível em: <<https://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js>>. Citado nas páginas 30, 31 e 32.
- CARVALHO, I. et al. Classificação da vegetação do parque nacional da chapada das mesas, maranhão, usando obia, machine learning e softwares livres. *Simpósio Brasileiro de Sensoriamento Remoto*, 2019. Citado nas páginas 12 e 20.
- CUNHA, A. Como fazer a migração do expo para react native cli. *Alura*, Setembro 2021. Acessado dia 23 de Julho de 2022. Disponível em: <<https://www.alura.com.br/artigos/como-fazer-a-migracao-do-expo-para-react-native-cli>>. Citado na página 29.
- DIANA, J. Pesquisa descritiva, exploratória e explicativa. *diferenca.com*, s.d. Acessado dia 09 de Agosto de 2022. Disponível em: <<https://www.diferenca.com/pesquisa-descritiva-exploratoria-e-explicativa/>>. Citado na página 34.
- FABIO, B. *Previsão de Mortalidade em Cirurgia Cardíaca Congênita utilizando Aprendizagem de Máquina*. 2020, 2020. Citado na página 27.
- GURGEL, G. O que acontece com o sangue que você doa? *Faculdade de Medicina - UFMG*, 11 2019. Citado na página 11.
- HEMOMINAS, F. *Fracionamento*. 2014. Disponível em: <<http://www.hemominas.mg.gov.br/doacao-e-atendimento-ambulatorial/hemoterapia/fracionamento>>. Citado na página 22.
- HEMOMINAS, F. Convocação de doadores de sangue. *Fundação Hemominas*, 08 2021. Citado na página 13.
- Jornal Diário do Aço. Ipatinga faz apelo de urgência por novos doadores de sangue. *Jornal Diário do Aço*, 25 de Março 2021. Disponível em: <<https://www.diariodoaco.com.br/noticia/0086995-ipatinga-faz-apelo-de-urgencia-por-novos-doadores-de-sangue>>. Citado na página 11.
- KESTRING, B. A. *SISTEMA MÓVEL NA PLATAFORMA ANDROID PARA COMPARTILHAMENTO DE GEOLOCALIZAÇÃO USANDO MAPAS E NOTIFICAÇÕES DA GOOGLE*. 2014, 2014. Citado nas páginas 12 e 32.

KUMAR, B. What is scikit learn in python. *pythonguides*, Outubro 2021. Acessado dia 23 de Julho de 2022. Disponível em: <<https://pythonguides.com/what-is-scikit-learn-in-python>>. Citado na página 28.

Lei Nº 10.205 da Constituição Federal do Brasil. § 4o do art. 199 da constituição federal, relativo à coleta, processamento, estocagem, distribuição e aplicação do sangue, seus componentes e derivados, estabelece o ordenamento institucional indispensável à execução adequada dessas atividades, e dá outras providências. *Presidência da República Casa Civil*, 2001. Citado na página 11.

Marcílio Junior. Doe+: Um aplicativo móvel de cunho social para agendamento de doação de sangue no hemocentro público de alagoas. *CODAI – Universidade Federal Rural de Pernambuco (UFRPE)*, p. 1 – 8, 2018. Citado nas páginas 12, 15, 16 e 17.

MASTERSINDATASCIENCE. What is a decision tree? *mastersindatascience.org*, s.d. Acessado dia 23 de Julho de 2022. Disponível em: <<https://www.mastersindatascience.org/learning/machine-learning-algorithms/decision-tree>>. Citado na página 27.

MATEUS, G. R.; LOUREIRO, A. A. F. *Introdução a computação móvel*. 1. ed. [S.l.]: NC, 1998. Citado na página 12.

MATOS, D. Por que cientistas de dados escolhem python? *Digital House*, Abril 2019. Acessado dia 22 de Julho de 2022. Disponível em: <<https://www.cienciaedados.com/por-que-cientistas-de-dados-escolhem-python/>>. Citado na página 28.

Ministério da Saúde. *Caderno de Informação Sangue e Hemoderivados DADOS DE 2014*. 9. ed. Brasília, DF: MINISTÉRIO DA SAÚDE, 2015. Citado na página 21.

Ministério da Saúde. *Doação de sangue: como doar, quem pode doar, impedimentos*. 2020. Disponível em: <<https://antigo.saude.gov.br/saude-de-a-z/doacao-de-sangue>>. Citado nas páginas 11, 12, 13, 21 e 22.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. In: *Sistemas Inteligentes Fundamentos e Aplicações*. 1. ed. Barueri-SP: Manole Ltda, 2003. p. 89–114. ISBN 85-204-168. Citado nas páginas 23 e 24.

MULLOVERTHING. How is an extra tree classifier different from a decision tree? *mulloverthing.com*, Agosto 2019. Acessado dia 23 de Julho de 2022. Disponível em: <<https://mulloverthing.com/how-is-an-extra-tree-classifier-different-from-a-decision-tree/>>. Citado na página 27.

OUHBI, S. et al. Predicting software product quality: A systematic mapping study. *Computacion y Sistemas*, v. 19, p. 547–562, 09 2015. Citado na página 22.

PEREIRA, J. R. et al. Doar ou não doar, eis a questão: uma análise dos fatores críticos da doação de sangue. *ABRASCO - Associação Brasileira de Saúde Coletiva*, 2016. Citado na página 11.

RODRIGUES, T.; JÚNIOR, O.; PEREIRA, A. Partiu doar sangue - uma plataforma web e aplicativo mobile para captação e gestão inteligente de doação de sangue e hemocomponentes. *Anais do XXIII Simpósio Brasileiro de Sistemas Multimídia e Web: Workshops e Pôsteres*, p. 1 – 5, 2017. Citado na página 11.

RODRIGUES, V. Métricas de avaliação: acurácia, precisão, recall... quais as diferenças? *medium.com*, Abril 2019. Disponível em: <<https://vitorborbarodrigues.medium.com/métricas-de-avaliaç~ao-acurácia-precis~ao-recall-quais-as-diferenças-c8f05e0a513c>>. Citado nas páginas 24, 25 e 26.

SANTOS, T. INCENTIVOS À DOAÇÃO DE SANGUE: FOMENTO À DIGNIDADE DA PESSOA HUMANA X COMERCIALIZAÇÃO INCONSTITUCIONAL . *Cadernos da Escola de Direito e Relações Internacionais*, p. 1 – 19, 2010. Citado na página 11.

SAÚDE, T. *Transfusão de sangue: o que é, quando é necessária e como é feita*. 2021. Disponível em: <<https://www.tuasaude.com/em-que-situacoes-e-indicada-a-transfusao-de-sangue/>>. Citado na página 11.

SEVERO, C. E. P.; SANTOS, H. M. d. Bloodsys: controlando o processo de doação de sangue para hemocentros. *Revista Eletrônica de sistemas de informação e gestão tecnológica*, p. 1 – 16, 2018. Citado nas páginas 12, 15 e 16.

SOUZA, M. H. L.; ELIAS, D. O. *PRINCÍPIOS DE HEMATOLOGIA E HEMOTERAPIA*. 2. ed. Rio de Janeiro , RJ: Centro de Estudos Alfa Rio, 2005. Citado na página 11.

TARGETTRUST. React e react native: um passo além do javascript. *TargetTrust Escola de TI*, 2021. Acessado dia 23 de Julho de 2022. Disponível em: <<https://targettrust.com.br/blog/react-e-react-native-um-passo-alem-do-javascript/>>. Citado nas páginas 28 e 29.

TECHOPEDIA. Scikit-learn: What does scikit-learn mean? *Techopedia*, s.d. Acessado dia 20 de Julho de 2022. Disponível em: <<https://www.techopedia.com/definition/33860/scikit-learn>>. Citado na página 28.