Faculdades Integradas de Taquara - Faccat
Av. Oscar Martins Rangel, 4.500
Taquara, RS, CEP 95600-000

***Curso de Sistemas de Informação***

# SAFE FAMILY: APLICATIVO DE REDE COLABORATIVA PARA RASTREAMENTO DE *BEACON*

Rafael Marmitt Feiten
Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil
rafaelfeiten@sou.faccat.br

Leonardo Ribeiro Machado
Professor Orientador
Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil
leonardomachado@faccat.br

**Resumo**
O desaparecimento de crianças é um problema crescente em todo o mundo. Visando solucionar este problema, foi desenvolvido o aplicativo Safe Family, que através de *beacons,* pequenos dispositivos móveis que emitem sinal *Bluetooth Low Energy,* realiza o monitoramento do utilizador, que pode ser uma criança ou animal de estimação, criando um cercamento eletrônico, além de auxiliar o usuário a localizar um utilizador perdido, através do rastreio global e chat para comunicação entre usuários do aplicativo. A construção do aplicativo foi norteada pelo método Kanban. Este artigo apresenta o projeto de pesquisa e o processo de análise e desenvolvimento do sistema proposto.

**Palavras-chave:** *Beacons.* Rede colaborativa. Aplicativo. Crianças. Animais de estimação.

## SAFE FAMILY: *COLLABORATIVE NETWORK APPLICATION FOR BEACON TRACKING*

***Abstract***
*The disappearance of children is a growing problem worldwide. In order to solve this problem, the Safe Family application was developed, which through beacons, small mobile devices that emit Bluetooth Low Energy signal, monitors the user, who can be a child or pet, creating an electronic fence, in addition to assist the user in locating a lost user, through global tracking and chat for communication between users of the application. The construction of the application was guided by the Kanban method. This article presents the research project and the process of analysis and development of the proposed system.*

***Keywords:*** *Beacons. Collaborative network. App. Children. Pet.*

# 1 INTRODUÇÃO

O presente trabalho consiste na criação um aplicativo móvel, destinado às plataformas Android e iOS, com o objetivo de auxiliar os responsáveis por crianças ou animais de estimação (chamados de utilizadores na aplicação) a mantê-los à uma distância segura, a fim de evitar o desaparecimento dos mesmos, mas, caso ocorra o desaparecimento de um utilizador, o aplicativo também pretende auxiliar os responsáveis a localizar o mais rápido possível esse membro da família.

Somente no ano de 2018, o número de pessoas desaparecidas no Brasil passou de 82 mil casos, enquanto o número de pessoas localizadas representou apenas 63,7% em relação ao número de desaparecidos (FBSP, 2019). No mundo, o índice de desaparecimentos de crianças e adolescentes está em uma crescente de 10% ao ano, estes números, no Brasil, representam aproximadamente 50 mil casos registrados todos os anos (VIEIRA, 2018).

Através dos *beacons* (dispositivos que emitem sinal *Bluetooth Low Energy*), utilizados por crianças ou animais de estimação através de uma pulseira ou coleira, respectivamente, a aplicação realiza o monitoramento do *beacon* a fim de obter a distância do mesmo para o dispositivo *smartphone* executando a aplicação.

O resultado do trabalho foi originado a partir de seus objetivos geral e específicos, sendo seu objetivo principal desenvolver uma aplicação para o monitoramento e rastreio de *beacons*, que podem ser utilizados por crianças ou animais de estimação, controlando a distância entre o *beacon* e o dispositivo *smartphone*, disparando alertas quando a distância estabelecida for ultrapassada e criar uma rede colaborativa para ajudar na localização de utilizadores perdidos. Dos objetivos específicos, podem ser listados: (a) Pesquisar os trabalhos relacionados ao projeto que será desenvolvido; (b) Coleta de materiais científicos relacionados ao projeto proposto; (c) Realizar o rastreio de dispositivos *beacon* e estimar a distância do mesmo para o dispositivo *smartphone*, dentro do limite médio de até 50 metros entre os mesmos; (d) Realizar o cadastro de crianças ou animais de estimação e atribuí-los a um *beacon*; (e) Disparar alertas quando a distância estabelecida na aplicação para o *beacon* for ultrapassada; (f) Criar uma rede colaborativa que auxilie na localização de um utilizador perdido; (g) Manter um histórico de localização de cada *beacon*; (h) Criar um chat para a comunicação entre quem solicitou o “rastreio global” de um utilizador e quem o localizou.

Em se tratando de estrutura organizacional, o presente artigo está organizado como segue: a seção 2 apresenta um referencial teórico referente aos temas abordados no trabalho. A seção 3 apresenta alguns dos trabalhos já realizados acerca deste tema. A seção 4 aborda a metodologia empregada, apresentando o método ágil utilizado para concepção, elaboração e construção do trabalho. A seção 5 trata do processo de desenvolvimento e testes do aplicativo. A seção 6 apresenta os resultados obtidos, relatando as mudanças que ocorreram no layout até a aplicação final. E, por fim, a seção 7 aborda a conclusão do trabalho, retomando o tema e metodologia abordados neste trabalho, bem como a apresentação dos principais resultados obtidos e a sugestão de possíveis trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

O objetivo desta seção é demonstrar o estado da arte nos temas envolvidos nesta pesquisa.

### 2.1 Desaparecimento de pessoas

Desaparecimento é o sumiço repentino de uma pessoa, sem que nenhum familiar ou terceiro tenha conhecimento. Para que uma pessoa seja considerada como desaparecida é necessário que ela não seja localizada nos lugares que costumava frequentar ou em outros locais, ou seja, a família deve primeiramente realizar a tentativa de localização da mesma, e não se faz necessário o aguardo de algum intervalo de tempo para isto (MPSP, 2016).

O desaparecimento de pessoas é um problema no mundo todo, mas no Brasil esse caso é ainda pior. Dentre as reclamações de quem já teve um parente desaparecido estão: os cadastros nacionais de desaparecidos para adultos e crianças estão defasados e desatualizados; uma aparente falta de priorização do poder público em relação aos desaparecidos registrados, mesmo sendo considerados como caráter de urgência pelo mesmo; falta de cuidado e amparo com as famílias que possuem uma pessoa desaparecida, tanto por parte do governo e autoridades de segurança pública quanto da própria sociedade. Parte dos problemas enfrentados também se dá pela falta de conhecimento por parte da população sobre a necessidade de registro do desaparecimento na delegacia de polícia, gerando assim dados subestimados da real situação (COSTA, 2019).

Segundo o Fórum Brasileiro de Segurança Pública (2019), foram registrados mais de 82 mil casos de pessoas desaparecidas somente no ano de 2018, enquanto o número de pessoas localizadas no mesmo ano chegou a 52.328, apenas 63,7% em relação aos número de desaparecidos.

## 2.2 Desaparecimento de crianças e adolescentes

Embora a grande parte dos desaparecimentos de crianças e adolescentes sejam solucionados nas primeiras 48 horas após a ocorrência, existe ainda um percentual de 15% a 20% dos casos que não são localizadas por um longo período de tempo. Estima-se que quase 250 mil menores de idade ainda estejam desaparecidos no Brasil (VIEIRA, 2018).

O motivo para que esse percentual de crianças e adolescentes ainda continuem desaparecidos mesmo após as 48 horas de buscas se dá, em parte, por conta da falta de integração entre os sistemas de registro e notificação de desaparecimento. Essa falta de comunicação entre as autoridades e estados dificulta as buscas e prejudica "[...] que um ponto final seja colocado em histórias embebidas em lágrimas e desespero" (VINAGRE, 2017).

O Brasil é um país com muitos problemas sociais e o desaparecimento de crianças e adolescentes é um destes problemas graves que carece de medidas para seu enfrentamento. Ele exige respostas concretas dos órgãos governamentais e demais autoridades, bem como uma maior participação das famílias, que podem ajudar efetivamente na prevenção por meio do diálogo entre pais e filhos ou até mesmo a adoção de normas de segurança mais restritivas (VINAGRE, 2017).

## 2.3 Beacon

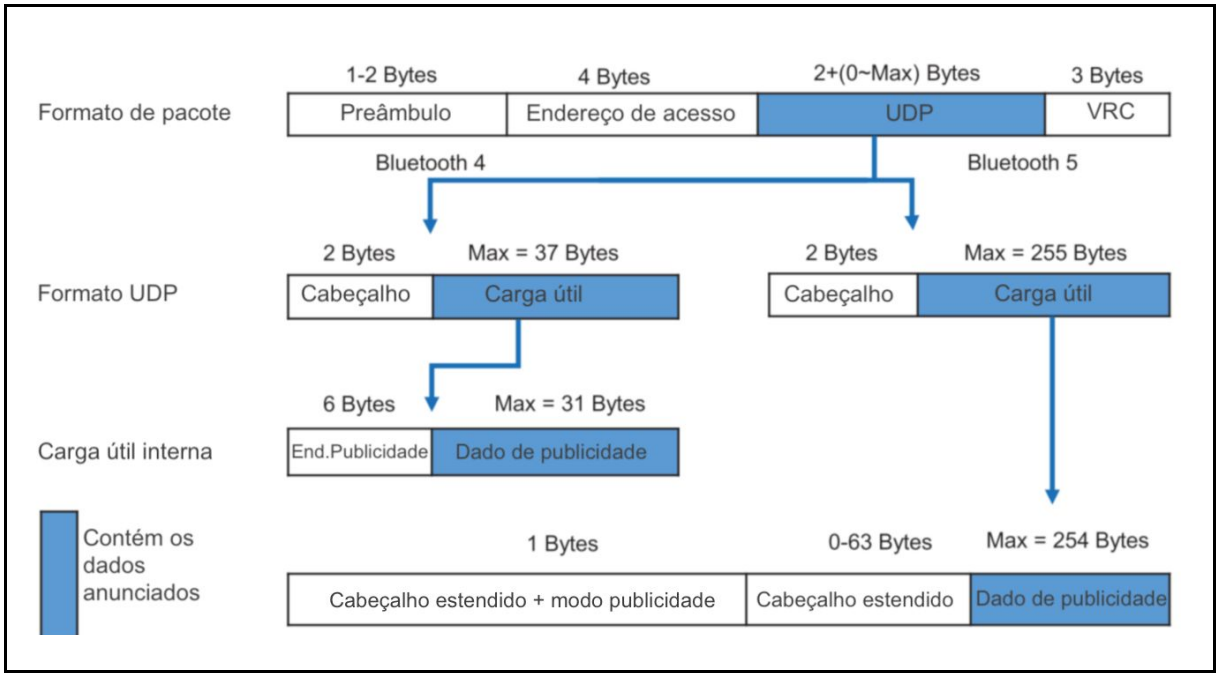
Os *beacons* são pequenos dispositivos móveis alimentados por bateria que emitem sinal *Bluetooth Low Energy* (BLE), variação do Bluetooth surgida junto com a versão 4.0, e qualquer aparelho com a tecnologia Bluetooth a partir desta versão pode enviar e receber sinais, como por exemplo um *smartphone*. Os *beacons,* em geral, são produzidos com o objetivo de "publicidade não direcionada", isso significa que eles transmitem seus dados para todos os dispositivos que estiverem no seu raio de alcance, sem necessidade de emparelhamento ou estabelecimento de conexão prévia (ESTIMOTE, 2020a).

Para realizar essa transmissão de dados, os *beacons* utilizam um protocolo de comunicação, sendo os dois principais protocolos do mercado o iBeacon e o Eddystone. O iBeacon, desenvolvido pela Apple em 2013, é o primeiro e mais utilizado protocolo. Já o Eddystone, foi desenvolvido pelo Google em 2015, para fazer frente ao protocolo da Apple. Eles possuem comportamento semelhante, e boa parte dos *beacons* do mercado possui suporte para ambos (KONTAKT, 2020).

Estes dispositivos não são projetados para uma transmissão rica de dados. Os *beacons* que utilizam a tecnologia Bluetooth 4.0 a 4.2 possuem um pacote de transmissão de dados equivalente a 31 bytes e velocidade de transmissão de 1 Mbps (Megabits por segundo). Já os dispositivos compatíveis com a versão Bluetooth 5.0 possuem um pacote de transmissão de dados de 255 bytes, e velocidade de transmissão de 2 Mbps. Essa maior capacidade do pacote na versão mais recente permite que sejam trafegados pequenos trechos de texto, como URLs, por exemplo, do *beacon* para o *smartphone* (RAHMAN, 2019). A estrutura deste pacote de dados é mostrado na Figura 1.

Os *beacons*, palavra que numa tradução literal significa farol, são conhecidos por emitirem periodicamente o mesmo pacote de dados, composto por um preâmbulo, um endereço de acesso, uma Unidade de Dados de Protocolo (UDP) e um Verificador de Redundância Cíclica (VRC). No pacote UDP, composto por 31 ou 255 bytes, é onde são inseridas informações do *beacon*, contendo neste, além de um cabeçalho: um Identificador Único Universal UUID (*Universally Unique Identifier*), um indicador do nível de potência de recepção (RSSI) e dois parâmetros denominados Major e Minor (RECK, 2016).

**Figura 1 - Estrutura do pacote de dados do *beacon***




Fonte: Adaptada de Yin *et al* (2019).

A tecnologia BLE utilizada por estes dispositivos *beacons* possui uma melhoria considerável em relação à versão clássica do Bluetooth no que diz respeito ao comportamento energético e eficiência na transmissão de dados (BLUETOOTH, 2020). Essa melhora energética permite que os *beacons* tenham uma vida útil média de 4 a 5 anos, dependendo do caso de uso (ESTIMOTE, 2020b).

## 2.4 Metodologia, tecnologias e ferramentas

### 2.4.1 Método Kanban

O Kanban foi desenvolvido por Taiichi Ohno como um meio de implementar o Just-In-Time (JIT) no sistema de produção da montadora de veículos Toyota (TPS). O objetivo foi desenvolver um método que pudesse atender as necessidades do mercado na época, limitando a produção de determinada peça apenas quando houvesse a capacidade de processá-la, e de maneira que as demais tarefas a serem realizadas pudessem ser organizadas em fila, antes de iniciarem o processo de fabricação (AHMAD *et al*, 2018).

O Kanban possui um método aberto e adaptável que permite os integrantes: visualizarem o fluxo de trabalho, mantendo-o sempre priorizado; limitar o trabalho em

progresso, com limites explícitos de itens que podem estar sendo produzidos (conforme a capacidade de processamento da equipe); e acompanhar o tempo de execução de cada tarefa, facilitando assim a previsibilidade e otimização do processo (KNIBERG; SKARIN, 2009).

### 2.4.2 React Native

O *framework* React Native (RN) foi desenvolvido para a construção de aplicativos Android e iOS utilizando o mesmo código-fonte com os componentes React, uma biblioteca JavaScript de código aberto para criação de interfaces de usuário interativas de forma fácil. Os componentes do RN são apoiados pelas mesmas visualizações do Android e iOS, e quando em tempo de execução o RN cria as visualizações dos componentes em ambas as plataformas, conforme os componentes nativos correspondentes, mantendo a aparência, comportamento e desempenho de um aplicativo nativo (REACT NATIVE, 2020).

Para executar o aplicativo, o React Native utiliza duas *threads* principais: a *thread* JavaScript (JS) e outra de módulos nativos (principal). A *thread* de JS é responsável por executar a lógica de negócio do aplicativo, com seus atributos, estados e valores. Ela também é responsável por enviar e receber mensagens para a *thread* principal. O módulo principal é quem recebe os comandos da *thread* JS e converte em interfaces de usuário na plataforma. Ela também é responsável por qualquer interação necessária entre o aplicativo e as funcionalidades nativas, como por exemplo acessar o Bluetooth do dispositivo (ESKOLA, 2018).

### 2.4.3 NodeJS

O NodeJS é um interpretador de linguagem JavaScript de software livre, com tempo de execução assíncrono, baseado em eventos e projetado para a criação de aplicativos escaláveis, com baixa possibilidade de travamentos do processo em execução (NODEJS, 2020).

Construído em cima do motor V8 da Google, o NodeJS possui arquitetura assíncrona, *single-thread* e JavaScript *server-side,* ou seja, projetado para executar códigos na linguagem JavaScript no lado do servidor, sendo considerada uma linguagem de programação mais dinâmica e performática quando comparada a linguagens fortemente tipadas (HERRON, 2018).

### 2.4.4 Firebase

A seguir estão descritas as ferramentas do Google Firebase utilizadas no desenvolvimento da aplicação.

- Authentication: Firebase Authentication é uma ferramenta do Google Firebase que conta com recursos como: realizar o registro do usuário na sua base de dados, garantindo a unicidade do mesmo; diversos métodos de *login*; controle de recuperação e troca de senha; bem como manter a sessão do usuário ativa na aplicação mesmo após o fechamento e/ou encerramento do mesmo (AUTHENTICATION, 2020).

- Firestore: O Cloud Firestore é um banco de dados NoSQL desenvolvido especialmente para aplicações móveis e web com foco na velocidade de consulta dos dados e estrutura flexível. Ele ainda possui fácil implementação através de SDKs, escalabilidade automática com forte garantia de consistência e suporte a sincronização em tempo real dos dados nos dispositivos conectados (FIRESTORE, 2020).

- Functions: Cloud Functions é um *framework* responsável por executar códigos *back-end* em resposta a eventos de outros serviços do Firebase ou requisições HTTPS. De fácil implantação, ele não possui manutenções e conforme a carga de requisições aumenta ou diminui, o Google escala rapidamente e de forma automática os recursos computacionais para atender as necessidades atuais da aplicação (FUNCTIONS, 2020).

- Hosting: Firebase Hosting é um serviço para hospedagem de conteúdo web estático e/ou dinâmico e que possui também integração com outros serviços do Firebase, como por exemplo o Cloud Functions, para a hospedagem de microsserviços. Provisionando automaticamente certificados SSL, todo conteúdo é exibido com segurança no Firebase Hosting (HOSTING, 2020).

## 3 TRABALHOS RELACIONADOS

Existem soluções no mercado que apresentam funções semelhantes às propostas pelo aplicativo Safe Family, com o uso de *beacons* ou GPS. O site Spyzie apresentou uma reportagem sobre o assunto, na qual realizou o levantamento de dez dispositivos mais conhecidos para o rastreamento infantil (SILVA, 2019), dos quais três foram selecionados para o comparativo.

### 3.1 Lineable

A Lineable (2020a) é uma empresa coreana que desenvolve pulseiras, relógios e aplicativos para o rastreio de seus produtos, utilizando as tecnologias Bluetooth, GPS e LPWAN. A ideia dos seus produtos é manter todos os membros da família conectados através dos seus dispositivos *wearables*.

Seu aplicativo propõe algumas soluções parecidas com o Safe Family, como: cerca geográfica, que é utilizado para definir uma área ao qual o dispositivo pode permanecer, e no caso de sair dela os *smartphones* conectados são avisados; e o aviso de perda de conexão com o *wearable* (LINEABLE, 2020b).

A aplicação apresenta um ponto negativo quanto à dependência do uso de *wearable* de sua própria produção/marca, não permitindo o rastreio de qualquer *beacon* disponível no mercado (LINEABLE, 2020b).

### 3.2 My Buddy Tag

A My Buddy Tag é uma fabricante de pulseiras Bluetooth desenvolvidas especialmente para o uso de crianças. O aplicativo da empresa, usado juntamente com as suas pulseiras, conta com funcionalidades parecidas com as propostas pelo Safe Family, como por exemplo a cerca geográfica, e outras funcionalidades extras como: botão de pânico na pulseira para que a criança possa apertar e alertar os dispositivos *smartphones* conectados; e sensor de água na pulseira, para alertar os responsáveis e prevenir possíveis afogamentos (MY BUDDY TAG, 2020).

A aplicação apresenta o mesmo ponto negativo do aplicativo da Lineable: segundo a My Buddy Tag (2020), seu aplicativo só permite o emparelhamento com pulseiras da própria fabricante.

### 3.3 Find My Kids

A aplicação Find My Kids foi desenvolvida especialmente para o acompanhamento infantil, disponível para as duas principais plataformas *mobile*, Android e iOS. Permite acompanhar a localização de uma criança através do sinal de GPS do celular da mesma, ou

via relógio com rastreamento GPS, caso a criança não possua *smartphone*. O aplicativo possui algumas funcionalidades como: cercamento eletrônico utilizando GPS, gravação de sons ao redor do dispositivo e histórico de localização (ÁLVARES, 2019).

O aplicativo Find My Kids possui a mesma proposta da aplicação Safe Family, mas utilizando diferentes recursos. Segundo Álvares (2019), para o funcionamento da aplicação é necessário que a criança possua um *smartphone* ou algum relógio com GPS com compatibilidade com a aplicação mais conexão de internet nos dois dispositivos: o do responsável e o da criança.

### 3.4 Diferenciais do aplicativo Safe Family

O aplicativo desenvolvido possui alguns diferenciais em relação aos outros que foram pesquisados neste trabalho, os quais são apresentados no Quadro 1.

**Quadro 1 - Diferenciais do aplicativo Safe Family**

| Funcionalidade | Safe Family | Lineable | My Buddy Tag | Find My Kids |
|---|---|---|---|---|
| Cercamento eletrônico | Sim | Sim | Sim | Sim |
| Tecnologia Bluetooth com rastreio de *beacons* | Sim | Sim | Sim | Não |
| Aplicativo em português | Sim | Não | Não | Sim |
| Funcionamento com outros *beacons* do mercado | Sim | Não | Não | Não |
| Manter um cadastro prévio de utilizadores | Sim | Não | Não | Sim |
| Visualizar demais usuários da aplicação no mapa | Sim | Não | Não | Não |
| Rede de rastreio colaborativa | Sim | Não | Não | Não |
| Chat entre usuários da aplicação para ajudar na localização de um utilizador perdido | Sim | Não | Não | Não |

Fonte: Autor (2020).

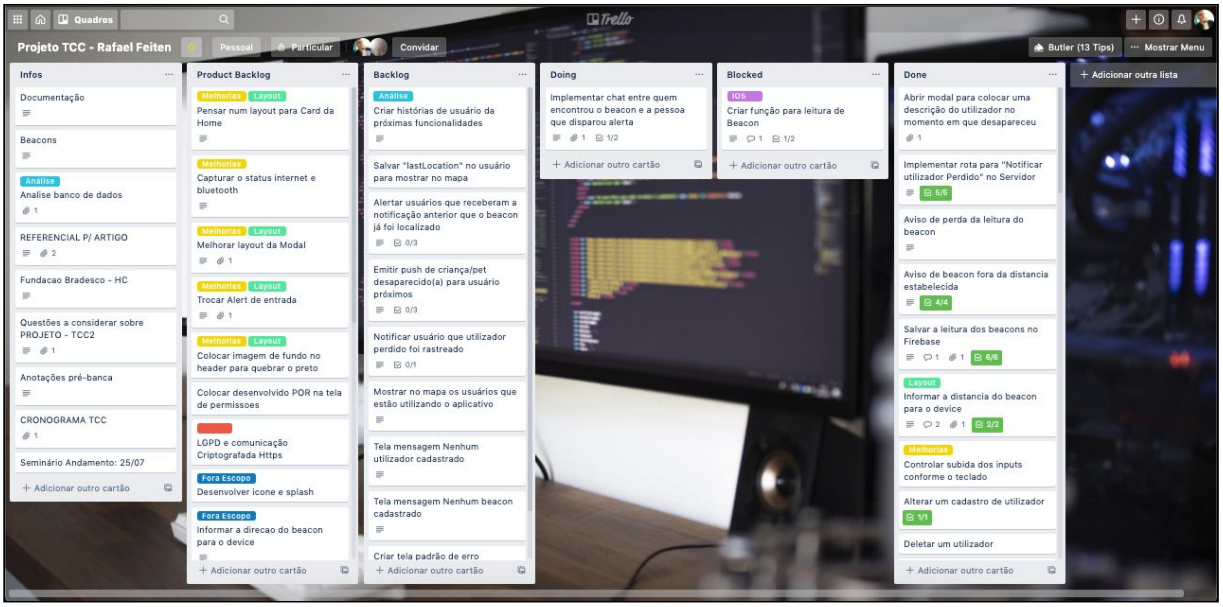
# 4 METODOLOGIA

Nesta seção são apresentados os processos de construção do aplicativo, partindo da concepção da idéia, a elaboração e construção da aplicação com padrões de desenvolvimento.

## 4.1 Concepção

Pensando em resolver o problema descrito, foi iniciado o processo de criação do servidor *back-end* e do aplicativo Safe Family, que através do uso de *beacons* realiza o monitoramento de crianças e animais de estimação, a fim de evitar o desaparecimento dos mesmos. Em conjunto com o método Kanban, escolhido para nortear o desenvolvimento da aplicação, foi utilizada a ferramenta Trello, possibilitando a organização, priorização e acompanhamento das tarefas levantadas, como demonstrado na Figura 2.

**Figura 2 - Ferramenta Trello com o quadro de tarefas**




Fonte: Autor (2020).

A análise inicial e a identificação dos requisitos partiram da ideia de entregar um aplicativo que pudesse ser acessado pelas duas principais plataformas do mercado *mobile*, Android e iOS. Para cumprir esse objetivo, decidiu-se criar uma aplicação utilizando o

*framework* React Native, devido ao fato de o mesmo atender ao requisito de distribuição a ambas as plataformas. A criação de um servidor também se fez necessária para a implementação de algumas funcionalidades, como a *push notification*, alteração dos dados de acesso ao aplicativo e armazenamento da imagem do utilizador. Para tal, foi utilizado o interpretador de linguagem JavaScript NodeJS.

## 4.2 Elaboração

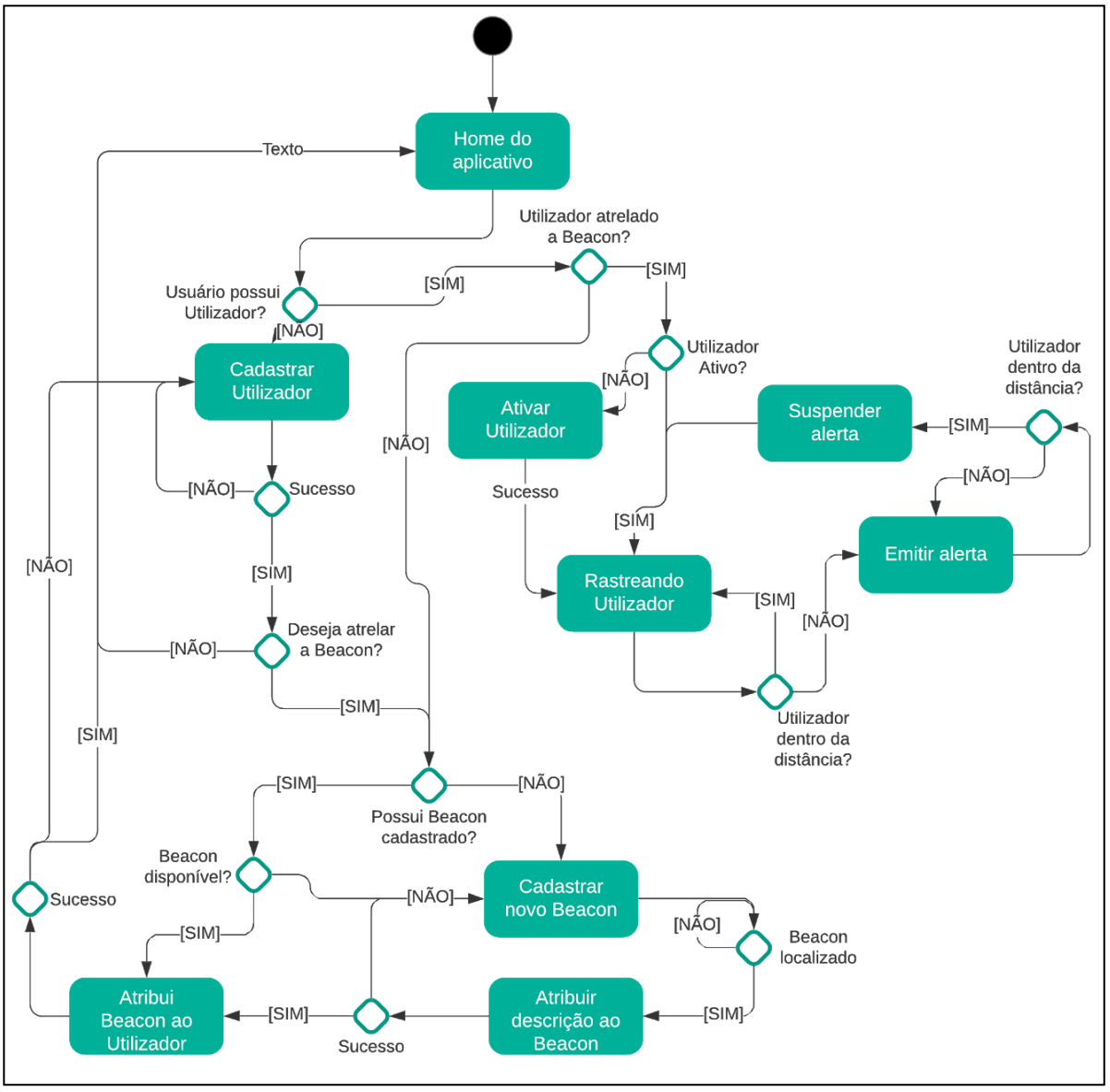
Depois de definida a aplicação a ser desenvolvida, criou-se uma lista de requisitos iniciais que a mesma deveria de conter. Estes requisitos foram quebrados e colocados de forma priorizada (de acordo com a lógica de implementação e importância da funcionalidade para a aplicação) na ferramenta Trello. Desta forma, foi possível visualizar e destacar o escopo de trabalho da aplicação.

O processo de criação das histórias de usuário se deu sempre pré-implementação de uma nova funcionalidade, ao passo em que, com as histórias de usuário, era possível mapear os pontos que a funcionalidade deveria atender, tornando a implementação da mesma mais simples de ser iniciada e executada.

Para as funcionalidades com maior grau de complexidade, antes da implementação da mesma, foram utilizados diagramas da *Unified Modeling Language* (UML). A UML é composta de diversos diagramas, com o objetivo de fornecer múltiplas visões, sob diversos aspectos, do sistema a ser modelado. O uso de diagramas para a análise do sistema permite que falhas sejam descobertas, diminuindo a possibilidade de erros futuros, durante a implementação (GUEDES, 2018).

Foram desenvolvidos dois diagramas de atividades, o primeiro, para analisar o funcionamento da parte inicial do aplicativo, envolvendo as funcionalidades de cadastro de usuário, recuperação de senha e *login* no aplicativo, e o segundo, foi desenvolvido para a análise do rastreio do *beacon* (Figura 3), envolvendo a sua relação com o utilizador atrelado, o cercamento eletrônico e o sistema de notificação, no caso da distância máxima do cercamento ser excedido pelo utilizador.

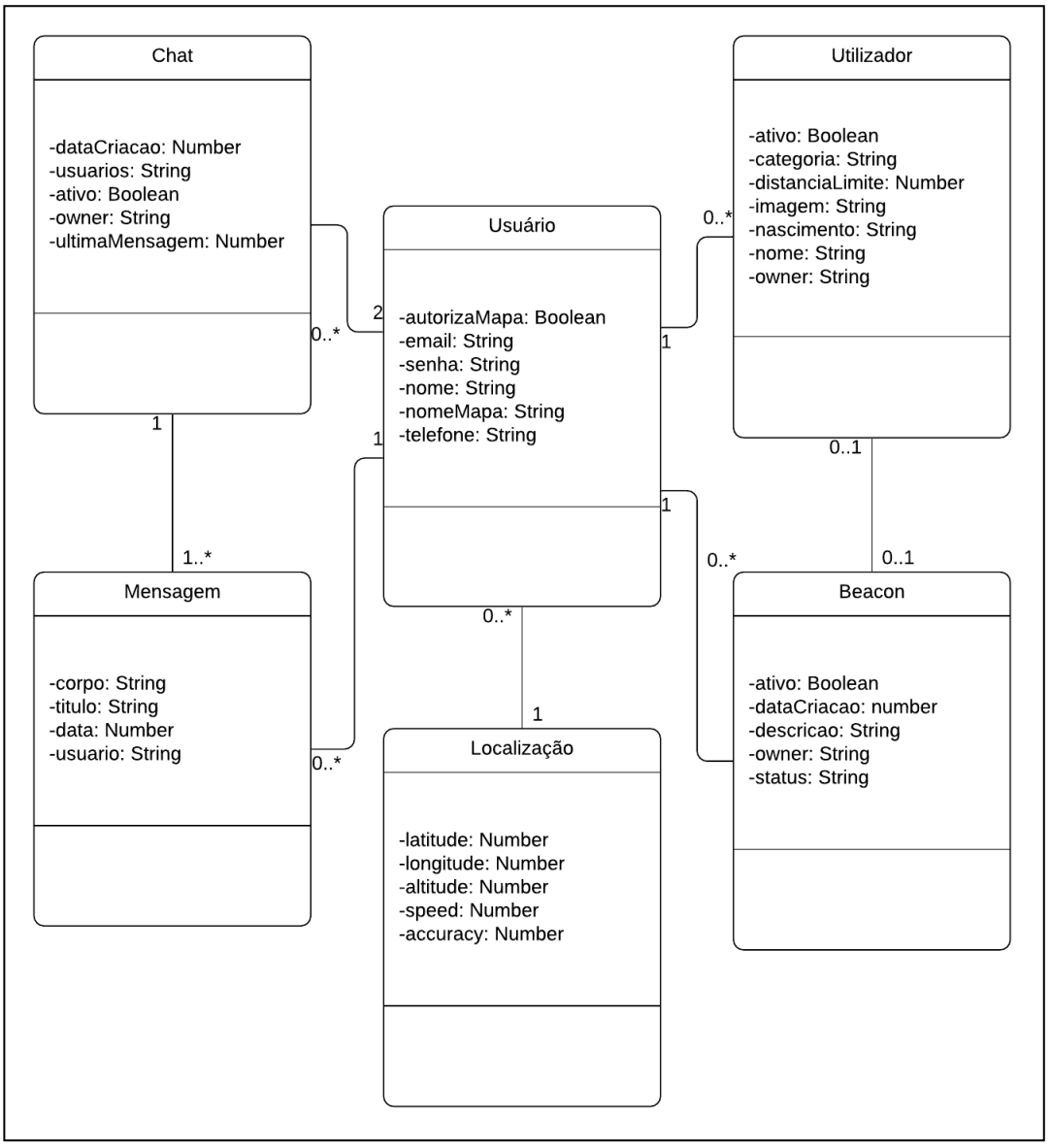
**Figura 3 - Diagrama de atividades - Rastreio de *beacons***




Fonte: Autor (2020).

Durante o desenvolvimento inicial do aplicativo, sentiu-se a necessidade da criação de um modelo de domínio (Figura 4), para que fossem mapeados os conceitos envolvidos, bem como os atributos pertencentes a cada um. Isso permitiu um maior entendimento da aplicação, facilitando o desenvolvimento de todas as funcionalidades que envolviam esses atributos.

**Figura 4 - Modelo de domínio**




Fonte: Autor (2020).

## 4.3 Construção

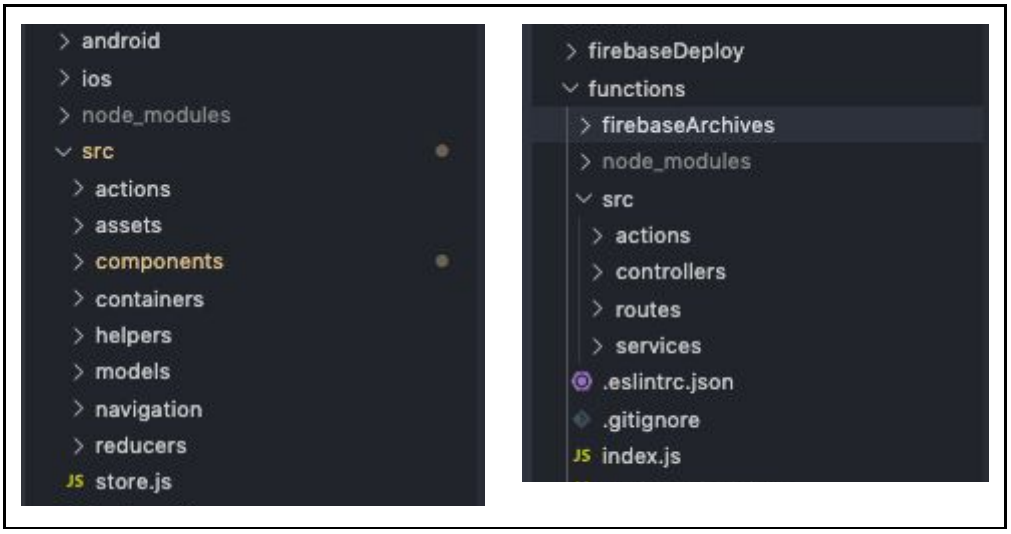
O método Kanban norteou a construção da aplicação. No desenvolvimento do aplicativo e servidor, foram utilizadas versões adaptadas do padrão *Model-View-Controller* (MVC).

No aplicativo, a camada *Container*, encontrada dentro da pasta “containers”, é a responsável por receber as chamadas de outras funções e repassar para a camada *Controller*. A camada *Controller* é a responsável por executar todas as regras de negócio pertencentes a

função, e por fim, chamar a camada *Component*, que é responsável por receber todos os dados da *Controller* e apresentá-los na tela. As camadas *Controller* e *Component* são encontradas dentro da pasta "components", onde estão separadas por pastas correspondentes a cada funcionalidade.

Enquanto isso, no desenvolvimento do servidor, criou-se uma camada denominada *Route*, inserida dentro da pasta "routes", ao qual, é responsável pela entrada das requisições, validar o método (*GET, POST, PUT* ou *DELETE*) e repassar a chamada para a camada *Controller*. A *Controller* é responsável executar todas as regras de negócio pertencentes a função, responder a requisição e consultar a camada *Action,* que possui a função de consulta ao banco de dados e demais APIS externas, e devolver a resposta para *Controller*.

**Figura 5 - MVC *front-end* e MVC *back-end***




Fonte: Autor (2020).

## 5 DESENVOLVIMENTO E TESTES

O objetivo desta seção é apresentar os processos de desenvolvimento do aplicativo e servidor, bem como os testes realizados na aplicação.

### 5.1 Desenvolvimento

No desenvolvimento, foi utilizado o editor de código-fonte Visual Studio Code. Essa ferramenta possui recursos próprios e extensões que auxiliam o desenvolvedor em diversos

aspectos, como por exemplo: integração com sistema GIT, que permite total interação com o repositório através de atalhos visuais; ESLint, ferramenta para análise de código-fonte, que permite identificar possíveis erros de codificação; Identação do código de forma automática, permitindo padronização da formatação e maior produtividade, uma vez que isso se torna transparente para o desenvolvedor.

O *framework* React Native foi utilizado para a construção do aplicativo. Ele permite a criação de toda a parte gráfica, bem como interação com os recursos nativos do *smartphone,* como o Bluetooth, para a leitura dos *beacons*.

Para o *back-end,* foi utilizado o NodeJS, que junto com o serviço do Firebase Functions, permite a criação de um servidor com arquitetura *serverless*, abstraindo o gerenciamento de máquinas, disponibilidade e escalabilidade das mesmas.

Através do SDK do Firebase, utilizado no *front-end* e *back-end*, foi possível realizar, de forma segura, o acesso aos serviços de: autenticação de usuário, banco de dados, *storage* das imagens e hospedagem do servidor.

Ao longo do desenvolvimento, novas funcionalidades foram mapeadas, bem como mudanças nas funcionalidades já propostas. Esse gerenciamento se deu com a ferramenta Trello, norteada pelo método Kanban. Com ela, foi possível realizar toda a gestão das tarefas de forma prática e visual.

## 5.2 Testes

A fim de construir uma aplicação com maior qualidade, foram utilizados dois tipos de testes: teste caixa-branca e teste exploratório.

Os testes caixa-branca (aqueles ao qual o testador possui acesso/conhecimento do código-fonte), foram realizados sempre ao final da implementação de cada funcionalidade, de forma individual. Isso permitiu que todas as funcionalidades pudessem ser validadas, avaliando sempre o fluxo principal, subfluxos e fluxos alternativos (exceções). Quando a ocorrência de algum comportamento inesperado foi detectado, uma tarefa para a correção do mesmo era criada de forma priorizada e sua implementação realizada logo ao final do teste.

Ao final, quando todas as funcionalidades mapeadas foram implementadas, iniciou-se a fase de testes exploratórios no aplicativo. Eles foram realizados primeiramente pelo próprio desenvolvedor, a fim de testar todo o escopo da aplicação e validar o fluxo completo do

mesmo. Após, alguns convidados foram selecionados para também realizarem o teste exploratório. Foi passado para eles o objetivo do aplicativo e uma breve descrição de algumas das funcionalidades. Com isso, o objetivo foi, além de validar o fluxo completo, também validar a usabilidade e experiência do usuário ao usá-lo.

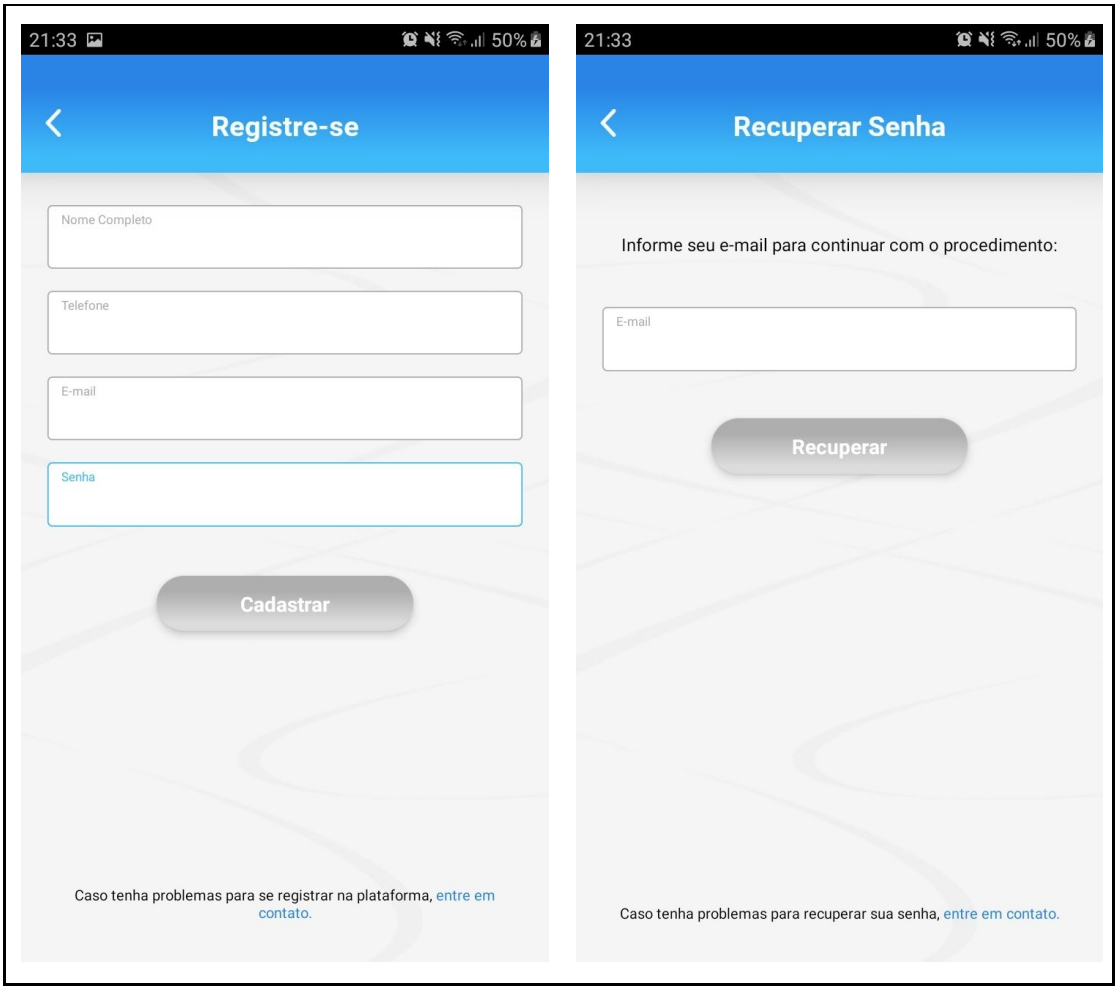
# 6 RESULTADOS

Como resultado, foi obtida uma aplicação *mobile*, para as plataformas Android e iOS, onde os usuários podem cadastrar seus filhos ou animais de estimação, e atrelá-los a *beacons*, possibilitando assim rastreá-los através da conexão entre o *smartphone,* executando a aplicação, e o *beacon*.

## 6.1 Aplicativo Safe Family

A parte inicial do aplicativo é a tela de "login". Nesta tela, o usuário que já possui um cadastro no aplicativo pode informar seus dados e realizar o *login* na aplicação. Caso o usuário ainda não possua uma conta, ele pode acessar a tela de "cadastro de usuário" e criar uma conta para acessar o mesmo (Figura 6, à esquerda). Porém, caso o usuário já possua uma conta, mas tenha esquecido sua senha, ele pode realizar a recuperação dela através da tela de "Recuperar senha" (Figura 6, à direita). Em ambas as telas, caso o usuário tenha problemas com o aplicativo, é possível entrar em contato com o autor, através de e-mail, clicando no link localizado na parte inferior da tela.

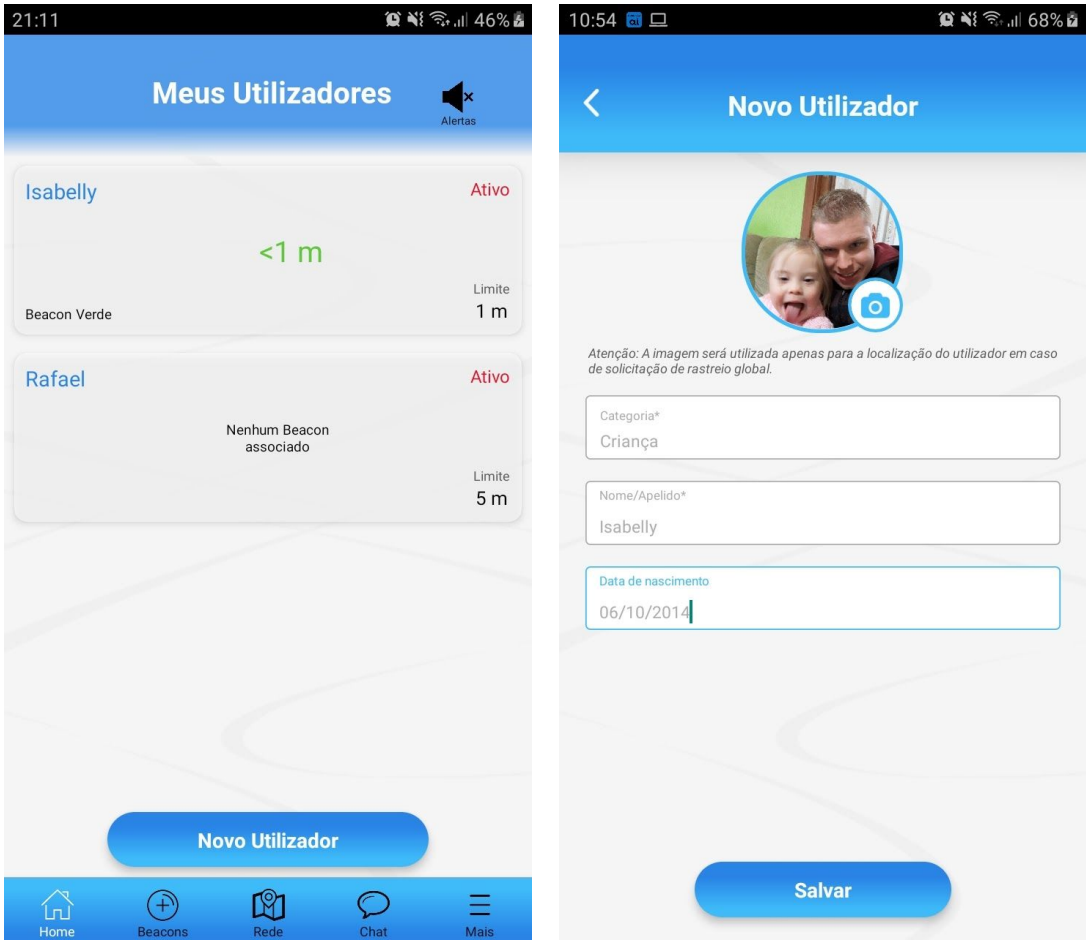
**Figura 6 - Tela de Cadastro de usuário e Recuperar senha**




Fonte: Autor (2020).

Após logar no aplicativo, o usuário tem acesso à tela principal da aplicação (Figura 7, à esquerda). Nesta tela, são apresentados os utilizadores cadastrados por ele. Também é possível cadastrar um novo clicando em “Novo utilizador”. A tela de cadastro de utilizador (Figura 7, à direita) possui dois campos obrigatórios, que são: Categoria (que pode ser uma criança ou um animal de estimação) e Nome/Apelido; o usuário também pode completar o cadastro informando a data de nascimento do utilizador, e se desejar, tem a opção de enviar uma foto que será usada para auxiliar os demais usuários a localizar a criança ou animal, caso o mesmo se perca.

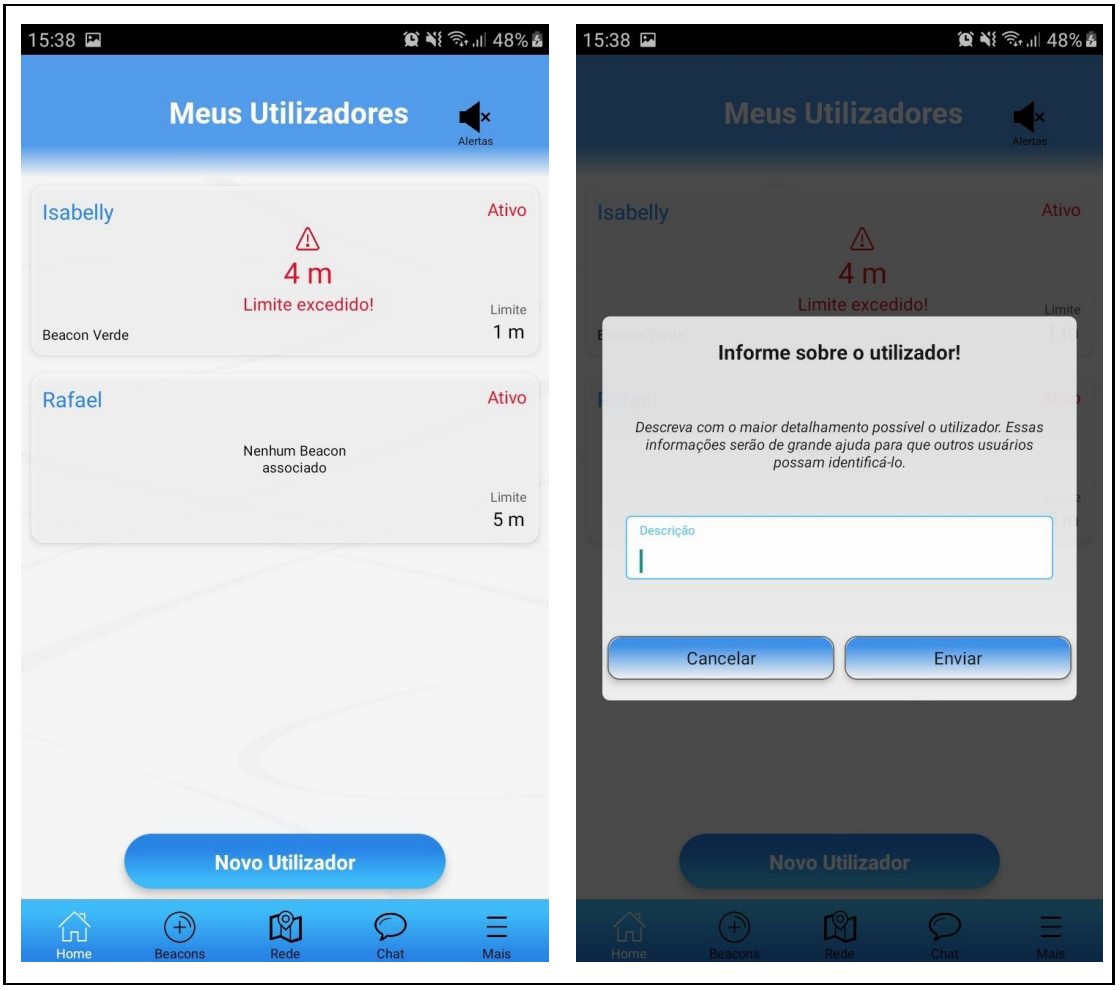
**Figura 7 - Tela de Meus utilizadores (*home*) e Novo utilizador**




Fonte: Autor (2020).

Quando a distância estabelecida para o utilizador (configurável na tela de edição do utilizador) for ultrapassada (Figura 8, à esquerda) a aplicação irá disparar um alerta, de forma sonora e vibração, para o usuário. Caso o responsável não consiga localizar a criança ou animal de estimação nas proximidades, ele pode notificar a perda do mesmo, informando uma descrição opcional que possa ajudar os demais usuários da aplicação a localizá-lo(a), como mostrado na Figura 8, à direita.

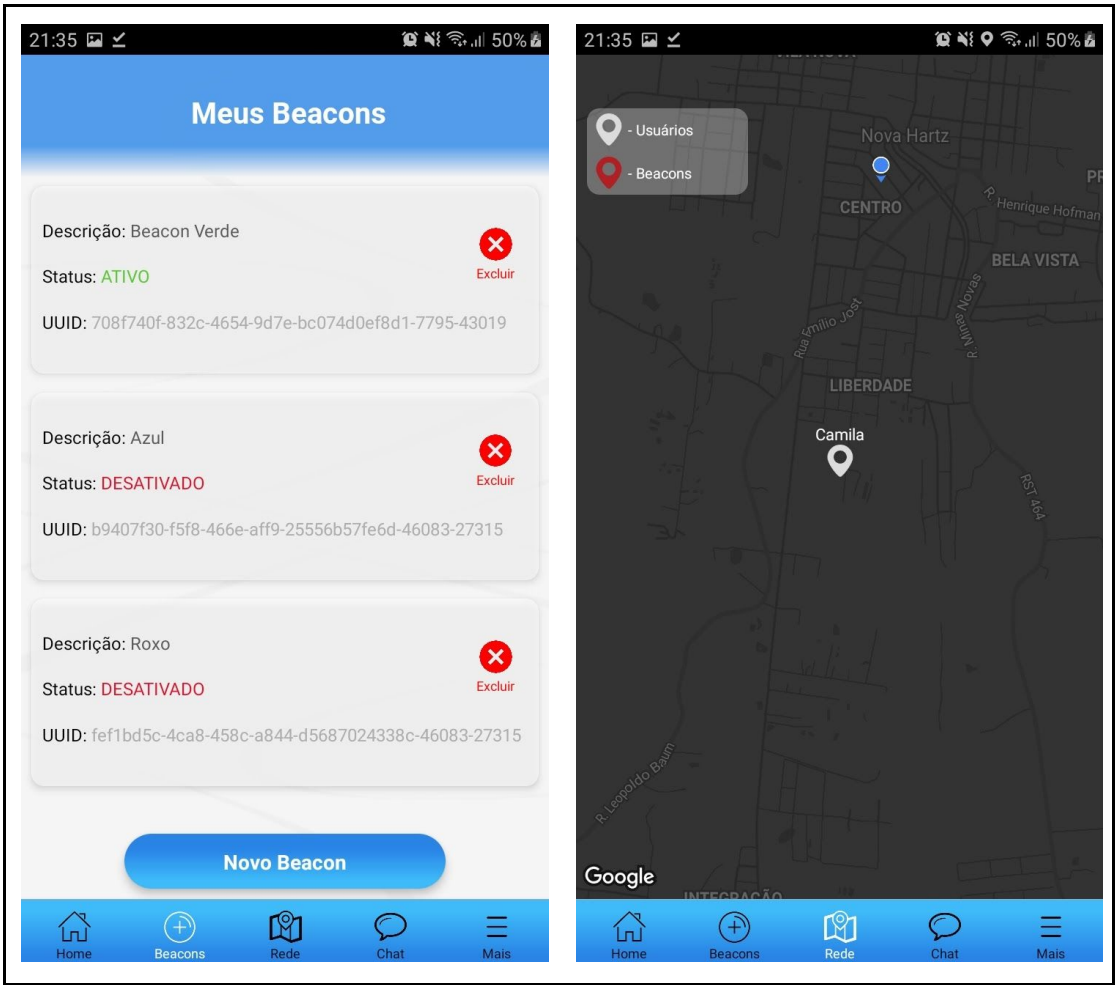
**Figura 8 - Tela de Meus utilizadores e Descrição do utilizador perdido**




Fonte: Autor (2020).

A Figura 9 apresenta as telas de “Meus Beacons” e “Mapa”. A primeira (localizada à esquerda), apresenta uma lista com todos os *beacons* cadastrados pelo usuário. Cada *beacon* da lista possui uma descrição cadastrada pelo usuário, e um estado, que pode ser “ativo” (informando que o *beacon* está em uso por um utilizador) ou “desativado” (quando o mesmo não está em uso). A segunda tela (localizada à direita), apresenta um mapa onde é possível visualizar a localização atual do usuário (representado por um ponto azul), a última localização dos demais usuários (representado por um *pin* cinza) que permitiram compartilhar sua localização, e a localização de um utilizador perdido quando rastreado por um outro usuário (representado por um pin vermelho).

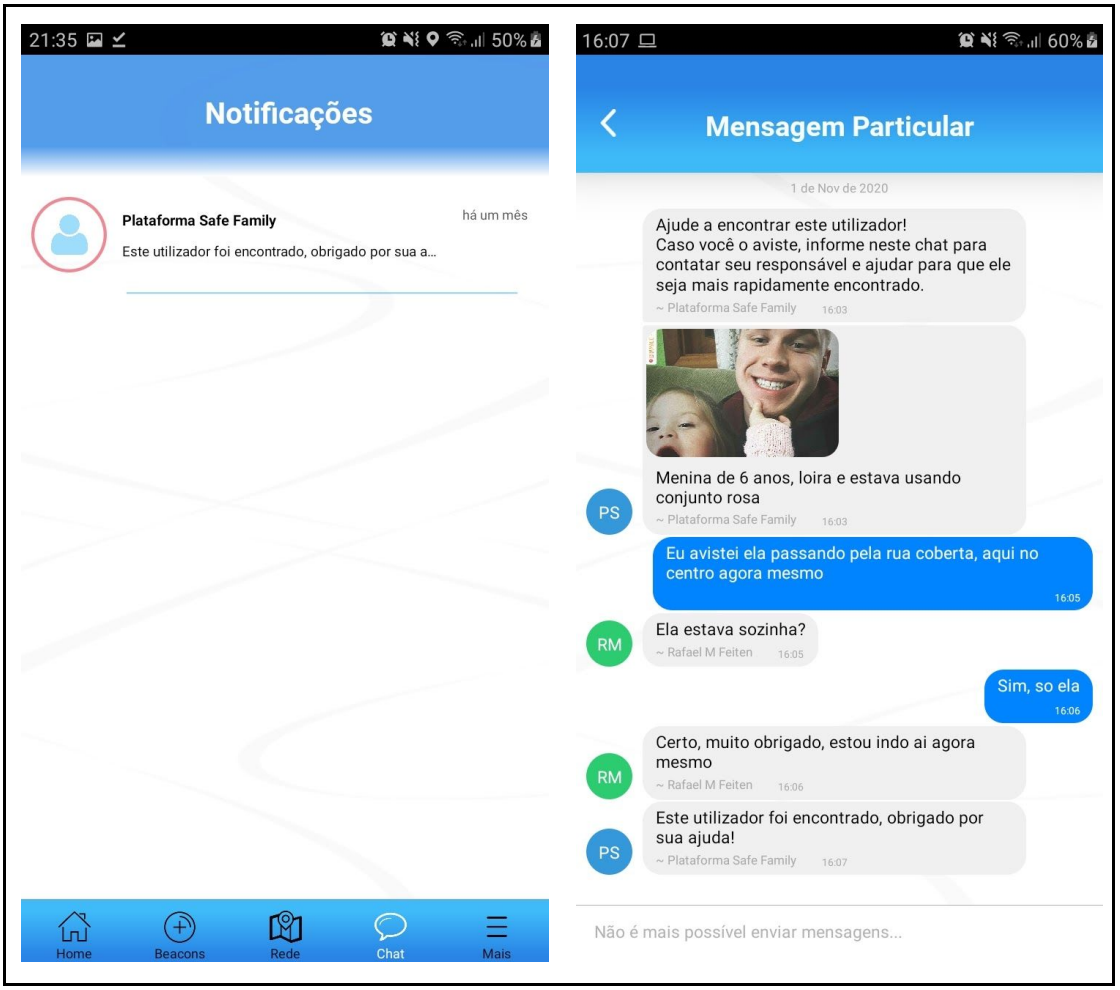
**Figura 9 - Tela de Meus *beacons* e Mapa (Rede)**




Fonte: Autor (2020).

A Figura 10 apresenta as telas de “últimas mensagens” e “chat”, que podem ser usadas pelo usuário para ajudar na localização de um utilizador perdido. O chat permanece ativo até que a criança ou animal de estimação, que havia se perdido, seja declarado(a) como encontrado(a) pelo responsável.

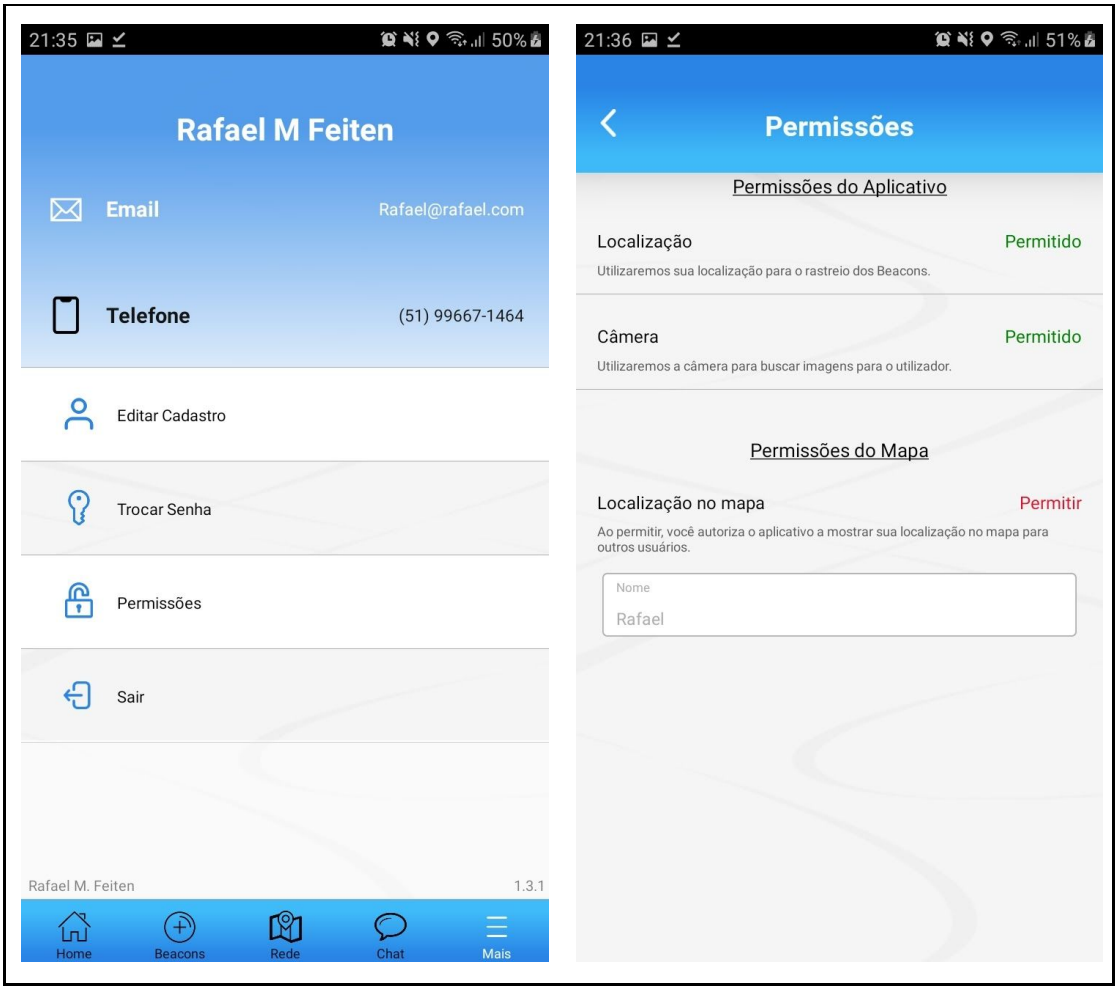
**Figura 10 - Tela de Notificações e Chat**




Fonte: Autor (2020).

Na tela de “Mais” (Figura 11, à esquerda), estão localizadas as ações de editar cadastro, trocar a senha de acesso, alterar permissões e sair da aplicação. Na tela de permissões (Figura 11, à direita), é possível visualizar as permissões do aplicativo para os recursos do *smartphone*, bem como a permissão de localização no mapa, que permite que outros usuário visualizem a sua última localização compartilhada.

**Figura 11 - Tela de Mais e Permissões**



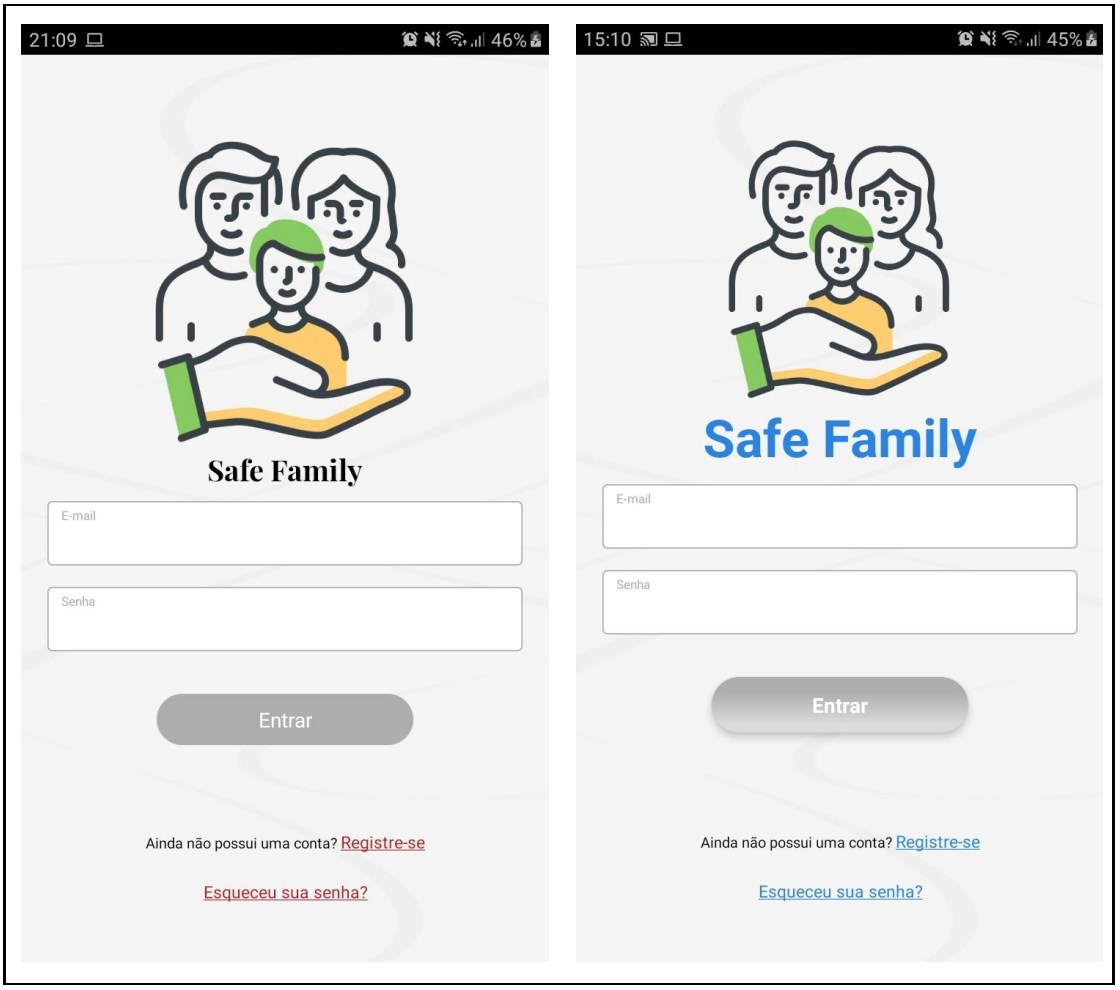

Fonte: Autor (2020).

## 6.2 Mudanças de layout

Ao longo do desenvolvimento do aplicativo, surgiu a necessidade de se realizar algumas melhorias no layout, visando apresentar um resultado mais agradável ao usuário final. Com esse objetivo, foram realizadas algumas reuniões com dois profissionais da área do design, voltados para a *User Interface* (UI) e *User Experience* (UX), onde foi possível apresentar a aplicação para eles e obter dicas com padrões de mercado. Essas dicas foram coletadas e analisadas para serem implementadas no aplicativo.

A Figura 12 apresenta as alterações ocorridas no layout da tela *Login*. Nela, foram realizadas pequenas mudanças com o objetivo de proporcionar um maior destaque no nome do aplicativo, bem como a implementação da nova paleta de cores.

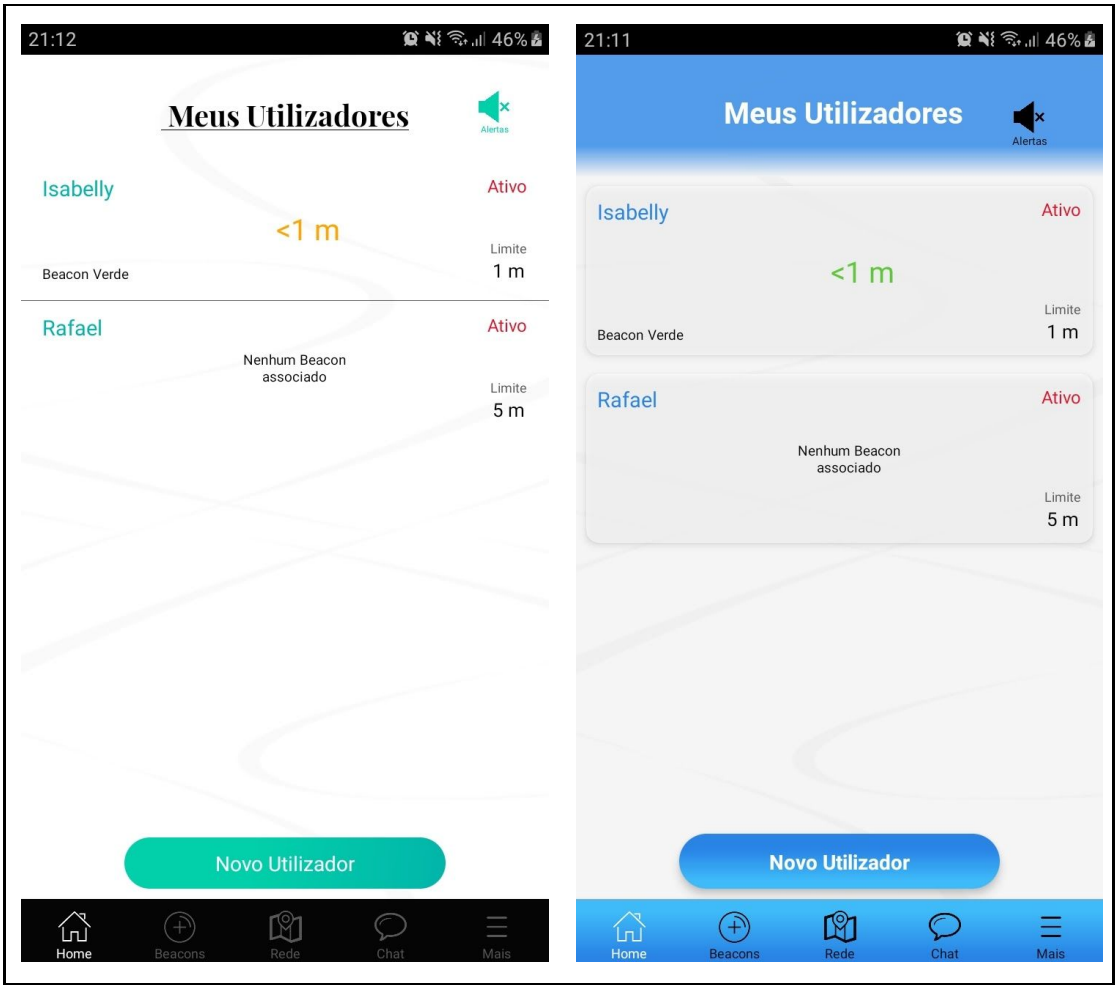
**Figura 12 - Mudança de layout na tela de *Login***




Fonte: Autor (2020).

A Figura 13 apresenta as mudanças de layout ocorridas na tela *Home*. A tela ganhou um *header* (cabeçalho) mais destacado, com degradê em tons de azul. Os *cards* que apresentam os utilizadores cadastrados também ganharam um novo estilo, que proporciona melhor separação e visualização. A *Tab Bar* (barra de abas) inferior também passou a adotar um degradê em tons de azul.

**Figura 13 - Mudança de layout na tela de *Home***




Fonte: Autor (2020).

## 7 CONCLUSÃO

Este trabalho de pesquisa e desenvolvimento envolveu a criação de um aplicativo móvel denominado Safe Family, que tem por objetivo auxiliar os responsáveis de crianças e animais de estimação a mantê-los a uma distância segura, evitando o desaparecimento dos mesmos, mas caso ocorra o desaparecimento, também auxiliá-los a localizar o mais rápido possível esse membro da família.

A utilização do método Kanban auxiliou na concepção, elaboração e desenvolvimento da aplicação, pois como se trata de um método ágil, a interação com as tarefas que surgiram ao longo do desenvolvimento foi trabalhada de forma natural, seguindo os princípios do método.

Entre os resultados obtidos, destaca-se a possibilidade de utilização de qualquer *beacon* do mercado compatível com o protocolo iBeacon. Outro ponto de destaque é a criação da rede de rastreio colaborativa, com a possibilidade de rastreio global do *beacon*, função essa de grande importância para a localização de uma criança ou *pet* perdido. A criação de um chat entre a pessoa que solicitou o rastreio global de seu utilizador e outro usuário da aplicação que o tenha avistado também é um ponto de destaque, pois permite uma maior agilidade no encontro deste utilizador perdido.

Futuramente, poderão ser incluídas novas funcionalidades no projeto a fim de adicionar maior valor à aplicação. Dentre as funções já levantadas, que podem incorporar a aplicação, estão a ampliação das categorias disponíveis para cadastro de utilizador, adaptação da aplicação para trabalhar - além do iBeacon - com o protocolo Eddystone, o compartilhamento do perfil - incluindo *beacon* e utilizador - entre os responsáveis, e a padronização do aplicativo segundo a Lei Geral de Proteção de Dados (LGPD).

## REFERÊNCIAS

AHMAD, M. O. *et al*. *Kanban in software engineering:* A systematic mapping study. *Journal of Systems and Software,* mar. 2018, p. 96 - 113.

ÁLVARES, Pedro. *Aplicação Find My Kids para o rastreamento das crianças*. 2019. Disponível em: <https://findmykids.org/blog/pt/aplicacao-find-my-kids-para-o-rastreamento-das-criancas>. Acesso em: 6 de set. 2020.

AUTHENTICATION, Firebase. Disponível em: <https://firebase.google.com/docs/auth/>. Acesso em: 17 de ago. 2020.

BLUETOOTH. *Aprenda sobre Bluetooth*: Versões de rádio. Disponível em: <https://www.bluetooth.com/learn-about-bluetooth/bluetooth-technology/radio-versions/>. Acesso em: 30 de ago. 2020.

COSTA, Gilberto. *Mais de 82 mil pessoas desapareceram no último ano, mostra anuário.* 2019. Disponível em: <https://agenciabrasil.ebc.com.br/direitos-humanos/noticia/2019-09/mais-de-82-mil-pessoas-desapareceram-no-ultimo-ano-segundo-anuario>. Acesso em: 24 de ago. 2020.

ESKOLA, Rasmus. *React Native Performance Evaluation.* 2018. 54p. Dissertação (Mestre em Ciência em Tecnologia) - School of Science, Aalto University, Espoo, 2018.

ESTIMOTE. *Beacons, how do they work?*. 2020a. Disponível em: <https://developer.estimote.com/how-beacons-work/>. Acesso em: 30 de ago. 2020.

ESTIMOTE. *The world's most iconic Bluetooth beacon.* 2020b. Disponível em: <https://estimote.com/>. Acesso em: 8 de set. 2020.

FBSP, Fórum Brasileiro de Segurança Pública. *Anuário Brasileiro de Segurança Pública 2019*. 2019. Disponível em: <https://forumseguranca.org.br/wp-content/uploads/2019/10/Anuario-2019-FINAL_21.10.19.pdf>. Acesso em: 24 de ago. 2020.

FIRESTORE, Cloud. Disponível em: <https://firebase.google.com/docs/firestore>. Acesso em: 17 de ago. 2020.

FUNCTIONS, Cloud. *Cloud Functions para Firebase*. Disponível em: <https://firebase.google.com/docs/functions>. Acesso em: 22 de ago. 2020.

GUEDES, G. T. A. *UML 2 - Uma abordagem prática*. 3. ed. São Paulo: Novatec, 2018.

HERRON, David. *Node Web Development:* Server-side development with Node 10 mane easy. 4. ed. Birmingham: Packt Publishing Ltd, 2018.

HOSTING, Firebase. Disponível em: <https://firebase.google.com/docs/hosting>. Acesso em: 22 de ago. 2020.

KNIBERG, Henrik; SKARIN, Mattias. *Kanban e Scrum*: Obtendo o Melhor de Ambos. Estados Unidos da América: C4Media Inc, 2009.

KONTAKT. *Beacon profiles:* iBeacon vs. Eddystone. Disponível em: <https://kontakt.io/beacon-basics/ibeacon-and-eddystone/>. Acesso em: 31 de out. 2020.

LINEABLE. *About us*. 2020a. Disponível em: <https://www.lineable.net/about>. Acesso em: 5 de set. 2020.

LINEABLE. *Lineable App*. 2020b. Disponível em: <https://www.lineable.net/application>. Acesso em: 5 de set. 2020.

MPSP, Ministério Público do Estado de São Paulo. *Cartilha de enfrentamento ao desaparecimento:* orientações e direitos na busca de uma pessoa desaparecida. 2016. Disponível em: <http://www.mpsp.mp.br/portal/page/portal/Cartilhas/EnfrentamentoDesaparecimento.pdf>. Acesso em: 25 de ago. 2020.

MY BUDDY TAG. *Frequently Asked Questions.* Disponível em: <https://mybuddytag.com/pages/faq>. Acesso em: 5 de set. 2020.

NODEJS. *About Node.js*. Disponível em: <https://nodejs.org/en/about/>. Acesso em: 22 de ago. 2020.

RAHMAN, Moklesur. *Energy efficiency evaluation of BLE 5 technology.* 2019. 65 p. Dissertação (Programa de Graduação em Engenharia de Comunicações Wireless) - Faculdade de Tecnologia da Informação e Engenharia Elétrica, Universidade de Oulu, Oulu, Finlândia, 2019.

REACT NATIVE. *Core Components and Native Components*. Disponível em: <https://reactnative.dev/docs/intro-react-native-components>. Acesso em: 23 de ago. 2020.

RECK, Marcelo S. 2016. *Beacons BLE – BLUETOOTH LOW ENERGY – Design e análise de um sistema de localização indoor*. 84 p. Monografia (Graduação) - Curso de Engenharia e controle de automação da Universidade de Caxias do Sul - Caxias do Sul, 2016.

SILVA, Joana. *Os 10 Melhores Dispositivos de Rastreamento de Crianças*. 2019. Disponível em: <https://www.spyzie.com/br/mobile-tracker/child-tracking-device.html>. Acesso em: 5 de set. 2020.

VIEIRA, Anderson. *Desaparecimento de crianças:* Conselho de Medicina faz alerta na Comissão de Direitos Humanos. 2018. Disponível em: <https://www12.senado.leg.br/noticias/materias/2018/04/06/desaparecimento-de-criancas-conselho-de-medicina-faz-alerta-na-comissao-de-direitos-humanos>. Acesso em: 29 de ago. 2020.

VINAGRE, José F. M. *O desaparecimento de crianças e adolescentes*. 2017. Disponível em: <https://www.crmpr.org.br/O-desaparecimento-de-criancas-e-adolescentes-13-48429.shtml>. Acesso em: 29 de ago. 2020.

YIN, Junjie *et al*. *A survey on bluetooth 5.0 and mesh:* New milestones of IoT. 2019. Disponível em: <https://ink.library.smu.edu.sg/sis_research/4540>. Acesso em: 31 de out. 2020.