

UNIVERSIDADE DE SÃO PAULO - USP
ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

UGO TOMIO FURUKO SAKAI
THOMAS KENZO TOMIDA

BEEAROUND

App Para Monitoramento De Abelhas e Ninhos

São Paulo - 2018

UGO TOMIO FURUKO SAKAI
THOMAS KENZO TOMIDA

BEEAROUND

App Para Monitoramento De Abelhas e Ninhos

Trabalho de Formatura do Curso de
Engenharia Elétrica apresentado à
Escola Politécnica da Universidade de São Paulo
Orientador: Prof. Antonio Mauro Saraiva

São Paulo - 2018

Lista de Imagens

- Figura 1: Arquitetura Básica do projeto;
- Figura 2: Diagrama de Casos de Uso, Aplicativo;
- Figura 3: Diagrama de Casos de Uso, BackEnd;
- Figura 4: Diagrama de casos de uso do backend
- Figura 5: Tela Inicial
- Figura 6: Tela de Login
- Figura 7: Tela de Cadastro
- Figura 8: Tela de Observações
- Figura 9: Tela Adicionar Observação
- Figura 10: Tela Espécies Encontradas:
- Figura 11: Tela Explorar Mapa
- Figura 12: Tela de Filtros
- Figura 13: Tela ao Selecionar Observação
- Figura 14: Tela ao Selecionar Observação [continuação]
- Figura 15: Tela de Sugerir Identificação
- Figura 16: Tela de Identificações
- Figura 17: LoopBack Explorer
- Figura 18: Cadastro Usuários
- Figura 19: Get Cadastro
- Figura 20: Informações sobre os registros
- Figura 21: Informações sobre as fotos

LISTA DE NOTAÇÕES, ABREVIATURAS

CS - Citizen Science

EBV - Essential Biodiversity Variables

API - Application Programming Interface

JSON - JavaScript Object Notation

LB - LoopBack

BD - Banco de Dados

SUMÁRIO

1	Introdução	09
1.1	Objetivo	10
1.2	Motivação	10
1.3	Organização do Trabalho	10
1.4	Justificativa	11
2	Aspectos conceituais	12
2.1	Referências na Literatura	13
3	Tecnologias Utilizadas	14
4	Especificação de Requisitos dos Sistema	15
4.1	Partes Interessadas	15
4.2	Requisitos dos Sistema	15
4.2.1	Requisitos Funcionais	15
4.2.2	Requisitos Não-Funcionais	16
5	Metodologia do Trabalho	17
5.1	Casos de Uso	18
5.1.1	Login	19
5.1.2	Sign Up	19
5.1.3	Registrar Observação	20
5.1.4	Postar Foto Ninho	21
5.1.5	Postar Foto Abelha	22

5.1.6 Visualizar Foto de Outros Usuários	22
5.1.7 Identificar Foto	23
5.1.8 Gerir Atividade	24
5.1.9 Armazenamento de Dados do App	25
5.1.10 Requisição de Dados do App	25
5.1.11 Resgate dos Dados	26
6 Implementação	27
6.1 FrontEnd	28
6.1.1 Início	28
6.1.2 Adicionar Observação	30
6.1.3 Explorar Mapa	31
6.2 BackEnd	34
6.2.1 LoopBack	34
6.2.2 Cadastro de Usuários	35
6.2.3 GET Cadastro	36
6.2.4 Registros e Informações das Fotos	37
7 Referências	38

1. Introdução

As abelhas são responsáveis por uma parcela muito alta da polinização de plantas por animais, sendo um dos animais mais bem adaptados para esse fim e representam cerca de 80% de todos animais responsáveis pela polinização.(A.B.E.L.H.A., 2018)

Atualmente a população de abelhas sem ferrão, como a Jataí, Araí e Uruçu está sendo ameaçada por espécies invasoras acidentalmente introduzidas por seres humanos. Com a iniciativa de coletar dados, conscientizar a população sobre abelhas e toda a biodiversidade que ela compõe e envolvê-la em alguns processos científicos, como a coleta de registros ou tentando identificar os registros, houve por parte dos biólogos Celso Barbieri e Sheina Koffler, a ideia de criação de um aplicativo mobile que por meio do conceito de CS(Citizen Science) haja a coleta de dados referentes às abelhas, sejam elas criadas ou nativas/selvagens.

O termo Citizen Science, é muito utilizado pela comunidade científica e representa a coleta de dados, dentre eles representando dados sobre as ocorrências das espécies em determinadas regiões, ou geografia de sua distribuição, entre outros fatores que podem influenciar em um ou mais aspectos estudados, os EBV("Variáveis Essenciais de Biodiversidade", em português) sendo esse padrão muito utilizado em CS pela comunidade(CHANDLER. 2017).

1.1. Objetivo

O objetivo principal desse projeto é criar um aplicativo mobile que possa realizar identificações de abelhas e seus ninhos para que eles possam ser identificados e mapeados. Com esses dados seria possível rastrear as populações de suas espécies e se tomar os devidos cuidados para a preservação dessas espécies.

Nesse aplicativo mobile será possível tirar fotos das abelhas e compartilhá-la. Outros usuários poderão então visualizar a foto e sugerir a qual espécie o animal pertence. Essa identificação estará ligada à localização e com isso seria possível mapear as populações dessas abelhas no país.

1.2. Motivação

Visando um monitoramento melhor das abelhas e utilizando o conceito de CS será realizado um projeto de aplicativo mobile e este será um grande auxiliador na preservação e monitoramento das abelhas assim como um instrumento de educação ambiental e biológico para as pessoas.

Esse app tem como principal motivação aumentar o engajamento dos criadores amadores e profissionais de abelha bem como uma melhor interação entre eles e os cidadãos, com o objetivo em comum de se obter mais dados sobre abelhas e melhorar a sua conservação.

1.3 Organização do Trabalho

Neste trabalho foram considerados as seguintes etapas:

- Esclarecimento de metas a serem alcançadas com o aplicativo;
- Esclarecimento de funcionalidades necessárias para entrega coerente;
- Estudo sobre as linguagens de programação e frameworks “React Native”, “Javascript+Node.js” e “MongoDB”;
- Definição de casos de uso a respeito do aplicativo Front End e Back End com o servidor de banco de dados;
- Revisão e estudo do aplicativo base já existente iNaturalist modificado;

- Implementação e adaptação de novas funcionalidades no app, bem como sua apresentação no aplicativo;

1.4 Justificativa

Com o constante desmatamento, poluição e crescente avanço da agropecuária por todo o território nacional, houve uma redução gradual da fauna e flora locais, o que inclui abelhas sem ferrão. Para aumentar o estudo sobre o monitoramento e conservação de espécies nativas de abelhas sem ferrão é necessária a participação não só de cientistas e entidades governamentais, como também da própria população local, pois esta é quem está mais próxima dos ninhos além de ser impactada diretamente pelas populações de abelhas locais, uma vez que estas conseguem polinizar plantas nativas que são bastante utilizadas na alimentação local e produção de remédios e loções medicinais.

Além de impactar a própria biodiversidade uma vez que toda a cadeia produtiva das espécies nativas estão em equilíbrio e com a introdução de espécies invasivas esse equilíbrio acaba se desfazendo, podendo levar a extinção de algumas espécies.

Levando tudo isso em conta, o projeto desenvolvido visou a aproximação do cidadão comum com o meio científico por meio do CS e por um aplicativo que consiga aumentar o interesse dos usuários.

2. Aspectos Conceituais

Com os conceitos anteriores já apresentados, o projeto utilizou o conceito de Citizen Science como principal pilar para a coleta de dados pelos seus usuários, e por isso contém aspectos lúdicos de forma a atrair um maior público não acadêmico, esses aspectos podem se apresentar forma de se engajar o público utilizando competições com o objetivo de se gerar várias observações de abelhas.

Algumas maneiras de se aumentar o engajamento do público não acadêmico poderão ser realizadas em uma segunda etapa do projeto, após a verificação de que o aplicativo está com todas as suas funcionalidades básicas e com a validação do aplicativo pelos stakeholders.

O projeto apresenta uma arquitetura básica apresentada pela figura 1, na qual o aplicativo é a parte visível pelos usuários do aplicativo mobile, que poderão enviar fotos relacionadas à temática do projeto e identificá-las, opinar sobre outras identificações. Esse módulo será inteiramente construído na framework React Native, uma linguagem em JavaScript desenvolvida pelo facebook, utilizada para criar aplicativos mobile nativos.

O BackEnd funciona como uma espécie de centralizador e difusor de dados. Com ele, dados específicos da plataforma iNaturalist (*aplicativo já existente de Citizen Science utilizado como base*) são enviados para o mesmo com sua estrutura base, e dados adicionais para nosso projeto são enviados ao nosso próprio banco de dados.

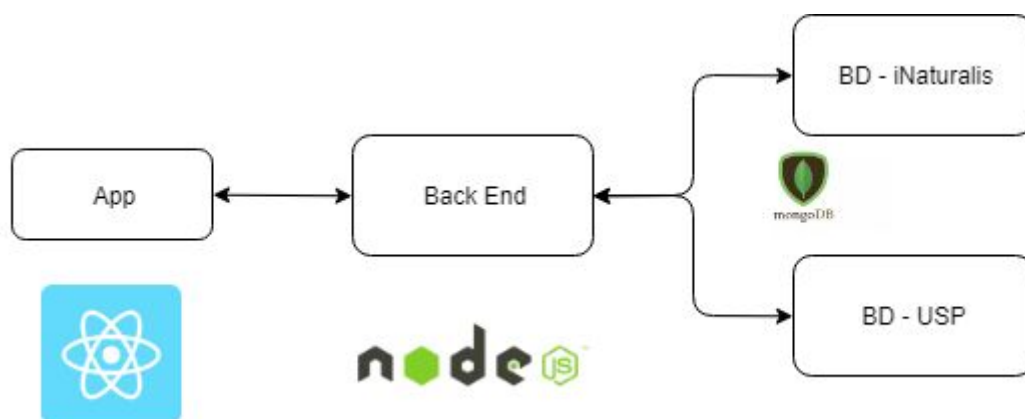


Figura 1: Arquitetura básica do projeto

2.1. Referências na literatura

Uma parte da estrutura será aproveitada do aplicativo iNaturalist, que é de autoria da Academia de Ciência da Califórnia. O projeto BeeAround foi fortemente influenciado pelo iNaturalist e por isso suas estruturas são bem influenciadas.

Uma das principais diferenças dos dois aplicativos são suas abrangências, pois, ao invés de abranger todas as espécies de animais, plantas e outros seres vivos, o BeeAround tem como foco espécies de abelhas, mais específico ainda, as sem ferrão.



Figura 2: iNaturalist, como funciona.

3. Tecnologias Utilizadas

Para o desenvolvimento do aplicativo será utilizada a ferramenta React Native, uma linguagem de programação desenvolvida pelo facebook para desenvolver aplicativos nativos com base em JavaScript.

Para o BackEnd do BeeAround serão utilizadas as ferramentas de loopback e Node.js para fazer todas as APIs necessárias para o controle dos dados recebidos e enviados do aplicativo.

O banco de dados do iNaturalist já possui uma API específica e apenas foi utilizada para armazenar a maior parte dos dados. O banco de dados do projeto, por sua vez é responsável pelo link entre as identificações de uma mesma foto e guardar dados do usuário que não possam ser armazenadas no banco de dados do iNaturalist.

4. Especificação de Requisitos do Sistema

4.1 Partes Interessadas

As partes interessadas do projeto se referem aos stakeholders, ou em outras palavras pessoas e agentes que são impactados direta ou indiretamente por processos do projeto, entre os principais stakeholders podemos citar:

- Pesquisadores, em sua grande parte biólogos;
- Cidadão comuns que tenham alguma interação com as abelhas:
 - Meliponicultores;
 - Apicultores;
 - Eletricistas;
 - Pessoas com interesse por Abelhas de modo geral.

4.2 Requisitos Do Sistema

O levantamento de requisitos do sistema estão listados abaixo, a nomeação dos requisitos seguem a seguinte característica para os requisitos funcionais, RF-X onde X é um número indicando qual é o requisito e os requisitos não funcionais serão identificados por RNF-X.

4.2.1 Requisitos Funcionais

- RF-1: O Sistema precisa conseguir sincronizar as informações após entrada do usuário na rede;
- RF-2: O Aplicativo tem que conseguir completar ou dar possibilidades de nomes(científicos/populares) para o usuário;
- RF-3: O usuário receberá e-mail para confirmação de cadastro;
- RF-4: O Sistema precisa possibilitar o login automático;
- RF-5: O Sistema deverá possibilitar a obtenção de dados via API;
- RF-6: O Sistema tem que possibilitar a criação, a destruição ou a alteração de usuários;

- RF-7: Os dados do usuário não devem ser destruídos mesmo que o usuário seja;

4.2.2 Requisitos Não-Funcionais

- RNF-1: Os dados dos usuários precisam ser guardados em segurança, em servidores confiáveis;
- RNF-2: O Aplicativo deve se manter funcional mesmo sem a possibilidade de comunicação com a rede e por conseguinte com o resto do sistema;
- RNF-3: O Sistema deve garantir a eficácia de seus dados para os pesquisadores diretamente relacionados, e proporcionar uma base de dados para pesquisadores indiretamente relacionados ampla o suficiente para atender a todos eles;
- RNF-4: O Aplicativo deve permitir que o usuário tenha uma experiência de uso do aplicativo muito boa;
- RNF-5: O Sistema deve ser escalável a até 10 mil usuários sem perdas de desempenho consideráveis;

5.1. Casos de Uso

A seguir apresentamos os casos de uso que foram pensados desde o início da captação dos requisitos a partir do stakeholders, e as consequentes adaptações e melhorias apresentadas depois de várias conversas e trocas de informações para entendermos como seria a melhor implementação desse sistema e como os stakeholders viam como seria o sistema final.

Os casos de uso serão mais detalhados a seguir. São eles:

1. Login
2. Sign Up
3. Registrar Obervação
4. Postar Foto Ninho
5. Postar Foto Abelha
6. Visualizar Fotos de Outros Usuários
7. Identificar Foto
8. Gerir Atividade
9. Armazenamento de Dados do App
10. Requisição de Dados do App
11. Resgate dos Dados

Caso De Uso 1: Login

Este caso de uso se refere ao usuário querendo entrar no aplicativo por meio de uma conta já existente.

O usuário na tela de login escolhe a opção de login, onde ele indica o usuário ou o email e depois a senha para entrar no aplicativo, que são enviados para o backend para a verificação, podendo ser feita no próprio backend, ou feita em verificação com o banco de dados, este requerendo do usuário estar conectado a uma rede e demandando mais tempo para a verificação.

Existem alguns fluxos alternativos que precisam ser levados em conta, como preferência por entrar no aplicativo por meio de outros autenticadores, como facebook/gmail, usuário esquece ou erra senha, ou ainda se o usuário tentar burlar o sistema de alguma forma.

Pré requisito: Usuário já possui uma conta

Fluxo Principal

1. - Usuário clica em "Fazer login";
2. - Usuário preenche E-mail/Senha;
3. - Dispositivo necessita estar conectado a internet;
4. - Esses dados são enviados para o backend para validação;

Fluxos Alternativos:

1. Usuário se conecta por outro autenticador (Facebook/Gmail) - 1;
2. Usuário erra a senha - 2;

Caso De Uso 2: Sign Up

Este caso de uso descreve o processo do usuário querendo se cadastrar no aplicativo.

Após inicializar o aplicativo o usuário escolhe a opção de Sign Up, para se cadastrar no sistema, que o leva a uma segunda tela onde terão campos para que o usuário indique seu usuário, seu e-mail, senha, que são os campos usuais de todo cadastro, porém o aplicativo irá requerer campos adicionais, como ano de nascimento, escolaridade(Sem escolaridade, Fundamental Incompleto, Fundamental Completo, Ensino Médio, Incompleto/Cursando, Ensino Superior

Incompleto/Cursando, Ensino Superior Completo, Pós Graduação Incompleto, Pós Graduação Completa), Município de residência, Profissão, Gênero(f/m/outros, caso seja outros: indicar), Meliponicultor(S/N, caso seja ano em que começou as atividades com abelhas).

Após todo esse processo o usuário recebe um e-mail de confirmação.

Os fluxos alternativos que precisam ser levados em conta são do usuário não preencher um campo, ou preencher um campo inválido, o usuário não confirmar o e-mail.

Fluxo Principal:

1. Usuário clica em “Cadastrar”;
2. Usuário preenche as informações solicitadas;
3. Usuário clica em “Finalizar Cadastro”;
4. Usuário recebe e-mail de confirmação;
5. Usuário confirma o e-mail;

Fluxos Alternativos:

1. Usuário não preenche um campo ou preenche de forma inválida - 2;
2. Usuário não confirma e-mail - 5;

Caso De Uso 3: Registrar Observação

Este caso de uso descreve o comportamento mais superficial e referente apenas ao processo de querer postar uma foto por parte do usuário, para isso ele deve indicar que deseja postar uma nova observação, via botão para registrar nova observação.

Com a indicação de registrar nova observação o usuário é desviado para um nova janela onde haverá uma janela para colocar uma ou mais fotos, um gps/localizador geográfico via API google maps, o usuário então identifica se as fotos que são de um ninho e as fotos que são das abelha, após essa separação são abertos novos campos para identificação do usuário.

Os fluxos alternativos para este caso de uso são, a postagem de uma foto fora de contexto, podendo ser a foto de uma espécie que mimifica a abelha em forma e características ou podendo ser fotos totalmente fora de contexto, sem haver

nenhum valor científico para o projeto em si, ambos casos seriam levados para um moderador que faria um posterior controle dos dados.

Fluxo Principal:

1. Usuário clica em “Nova Observação”
2. Usuário então seleciona quais fotos deseja registrar
3. Usuário marca quais fotos são referentes a ninhos e quais são abelhas
4. Usuário então verifica se a localização via GPS está correta ou não;
5. Então ele confirma as ações para os próximos setores;

Caso De Uso 4: Postar Foto Ninho

Este caso de uso é uma extensão do **Caso de Uso 3: Postar Foto**, após o usuário ter colocado as foto e confirmado o grupo de fotos é aberta duas sessões na mesma página que a postagem da foto, uma delas é para identificação dos ninhos.

Com isso é criado uma lista de checkpoints desejáveis para obtenção de dados, como local de apoio, se está fixado em construção humana, natural ou se é um apiário, identificação do local de apoio, se for humana, se é muro, poste e etc. se for um for natural uma possível identificação da planta usada como ponto de apoio, ou do terreno utilizado, rochas/subterrâneo.

Alguns dos fluxos alternativos são o usuário não souber/querer responder algum dos campos selecionados, onde tirando o campo de fixação o resto não seria obrigatório.

Pré - Condição: Usuário selecionou fotos de ninhos

Fluxo Principal:

1. É aberto uma sessão de edição de entrada sobre os ninhos (Substrato, Terreno);
2. Usuário preenche as informações;
3. Usuário confirma;

Fluxos Alternativos:

1. Usuário não sabe ou não deseja responder as informações - 2;

Caso De Uso 5: Postar Foto Abelha

Este caso de uso é a outro extensão do **Caso de Uso 3: Postar Foto**, onde assim como na identificação do ninho é aberto um sessão com campos para serem preenchidos pelo usuário.

Nos campos aberto para preenchimento estão a identificação da abelha e nomes sugeridos, além da identificação da flor e nomes sugeridos também.

Os fluxos alternativos nesse caso de uso estão relacionados ao usuário errar a identificação que ele fez ou não souber como identificar a espécie desejada e serão tratadas por outros casos de uso.

Pré - Condição: Usuário selecionou fotos de abelhas

Fluxo Principal:

1. É aberto uma sessão de edição de entrada sobre as abelhas;
2. Usuário preenche as informações;
3. Usuário confirma;

Fluxos Alternativos:

1. Usuário não sabe ou não deseja responder as informações - 2;

Caso de Uso 6: Visualizar fotos de outros usuários

Este caso de uso se refere ao feed de postagens feitas outros usuários.

Nele o usuário pode ver postagens de outros usuários nele estarão à mostra os conjuntos de fotos e identificações, com essas postagem será possível para o usuário verificar se ele concorda com a identificação do dono da postagem e se não estiver de acordo ele pode contestar essa identificação colocando uma identificação que ele mesmo ache plausível.

Entre os fluxos alternativos estão trocar de tela sem ver as postagens, dando continuidade para outros casos de uso.

Pré - Condição: Usuário entrou na aba de visualização;

Fluxo Principal:

1. Usuário clica em um registro de outro usuário;
2. Usuário recebe as informações do registro (foto, identificação);
3. Usuário pode avaliar o registro(Positivo, negativo);

Fluxos Alternativos:

1. Usuário não selecionar nenhuma identificação e abre mais registros - 1;

Caso de Uso 7: Identificar Foto

Derivando do caso de uso anterior, **Caso de Uso 6: Visualizar Fotos de Outros Usuários**, neste caso de uso é específico para a identificação da foto que o usuário quer contestar a identificação original.

Aqui o usuário após verificar que a identificação do dono da foto pode estar errada sugere um nome, científico ou popular, mais adequado para aquela espécie a identificação é então colocada a prova para outros usuários decidirem se ela é ou não pertinente à espécie analisada.

Identificações com muitas negações são escondidas/desativadas, enquanto as identificações positivas são mostradas com maior importância.

Entre os fluxos alternativos existe a possibilidade do usuário contestar um nome científico utilizando um nome popular ou vice e versa, neste caso deixaremos a cargo dos moderadores a política de escolha, podendo ter as duas identificações ativas, outro fluxo alternativo é a possibilidade do usuário colocar algum nome totalmente fora do contexto, nesse caso o usuário acaba perdendo credibilidade com relação às futuras identificações que ele fizer.

Pré - Condição: Usuário avaliou negativamente um registro

Fluxo Principal:

1. Após avaliar negativamente um registro, usuário sugere nova identificação
2. Nova identificação entra como identificação secundária
3. Identificação com votos negativos são excluídos

Caso de Uso 8: Gerir Atividade

Este caso de uso representa a possibilidade do Administrador do sistema conseguir manusear, permissões para os usuários do aplicativo, deixando alguns monitorando as ações gerais do aplicativo

Pré - Condição: Usuário possui a hierarquia de Administrador do aplicativo

Fluxo Principal:

1. Administrador pode 'contratar' novos moderadores a partir de usuários que ele julgue ser aptos para a tarefa;
2. Assim como ele pode contratar novos moderadores ele pode demitir moderadores que não estejam fazendo o trabalho correto;

Casos de Uso 9: Armazenamento de dados do App:

Os próximos casos de uso descrevem o funcionamento do BackEnd visto tanto pelo pesquisador como pelo aplicativo.

Esse caso de uso descreve o processo de armazenamento dos dados enviados pelo aplicativo para os banco de dados.

Após o usuário terminar de preencher os dados necessários do registro, o App envia esses dados para o BackEnd fazer o tratamento desses dados, ou seja, separa os dados que irão para o banco de dados do servidor do iNaturalist e dos servidores do projeto.

Os dados recebidos que são enviados para os Servidores do projeto recebem uma tag de ligação para apontar para os dados armazenados no Servidor iNaturalist.

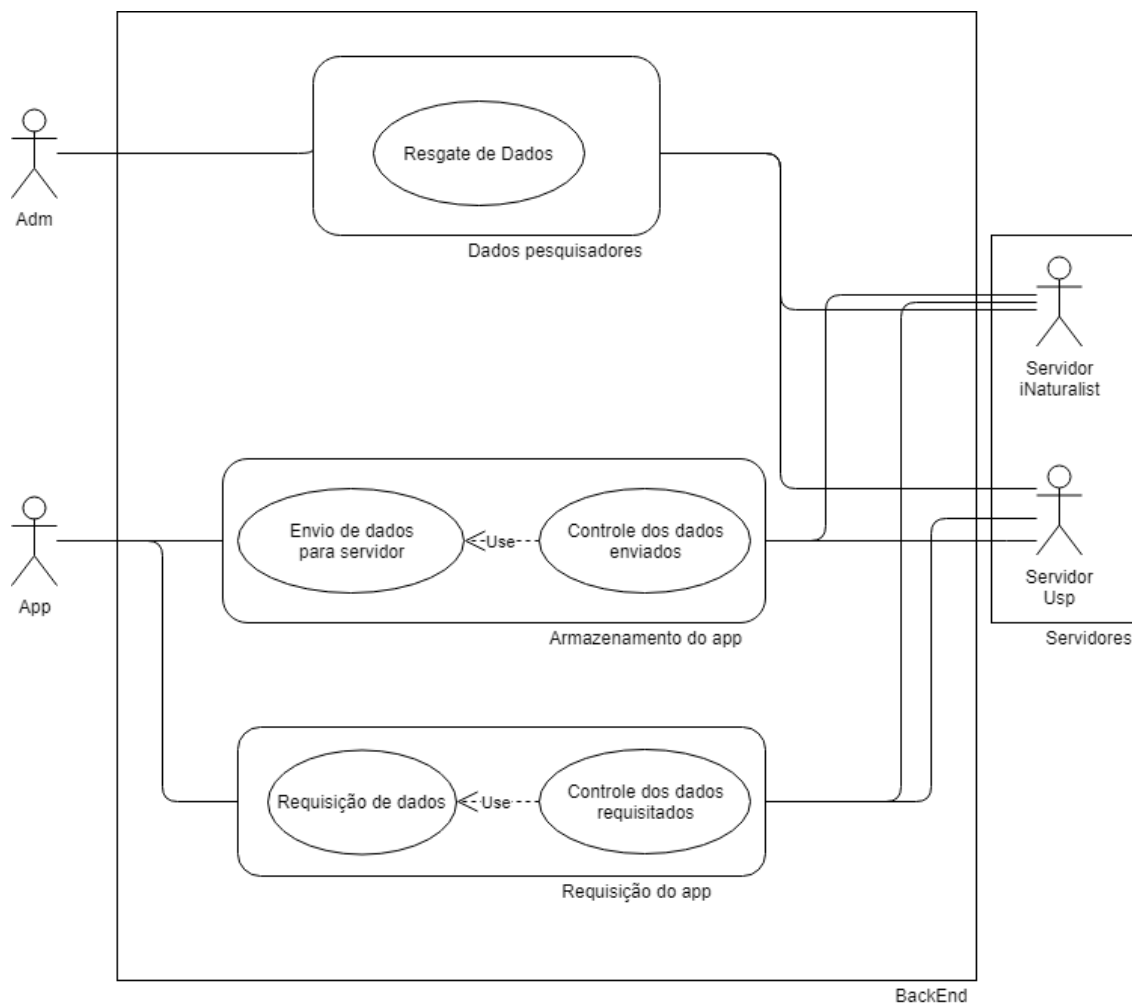


Figura 4: Diagrama de casos de uso do backend

Casos de Uso 10: Requisição de dados do App:

Esse caso descreve o processo do app requisitar informações de um registro.

O usuário ao selecionar um registro recebe as informações de ambos os servidores. Para isso, o Backend deverá fazer o tratamento desses dados e os compilar para enviar para o aplicativo.

Casos de Uso 11:Resgate dos dados:

Esse caso descreve o processo de resgate dos dados registrados no aplicativo para análise.

Um administrador externo, por via web, e não pelo aplicativo, pode selecionar os dados de ambos os servidores para análise desses dados. Podendo filtrar quais informações deseja e importá-los para um computador.

6. Implementação

A Implementação do projeto foi dividida em duas grandes partes assim como muitos projetos de desenvolvimento de softwares sendo eles o FrontEnd e o BackEnd.

O FrontEnd foi desenvolvido, como dito anteriormente em React-Native, não ocorrendo alterações muito significativas desde sua concepção na especificação do projeto.

Já para o BackEnd, foram feitos estudos e pesquisas para se descobrir como implementar e efetivá-lo no projeto para que se tenha um projeto completo de uma aplicação mobile com o controle feito em sua maior carga no BackEnd, após esses estudos e decisões tomadas foi decidido a implementação na framework do LoopBack, que utiliza a ferramenta de desenvolvimento Node.JS, o LB realizará um serviço de barramento entre o FrontEnd, o nosso BD em MongoDB e o BD do iNaturalist, onde é feita a comunicação via RESTAPI que eles disponibilizam.

6.1 FrontEnd

6.1.1 Início

O FrontEnd é composto por várias telas. Começando pela tela inicial, o aplicativo oferece 3 opções: Entrar, Login e Cadastro.



Figura 5: Tela Inicial

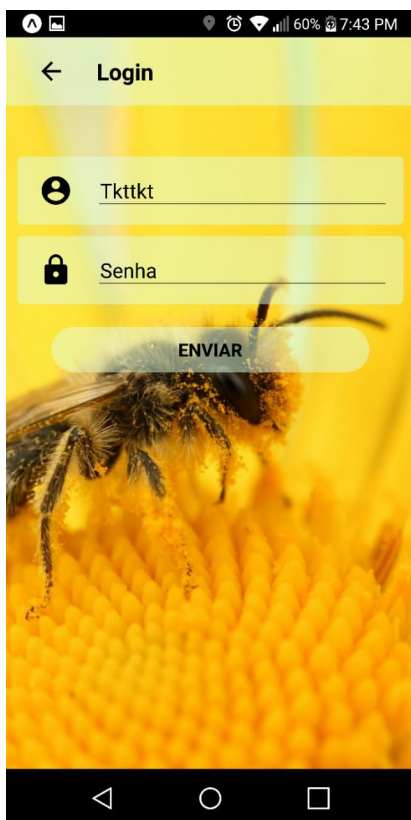


Figura 6: Login

Com a opção de Login, o usuário deve completar os campos de usuário e senha. Ao fazer isso, ele entra no aplicativo imediatamente.

Caso seja a primeira vez que o usuário entra no aplicativo, ele tem a opção de fazer o cadastro no aplicativo. Para isso ele deve fornecer um e-mail válido, um nome de usuário e uma senha.

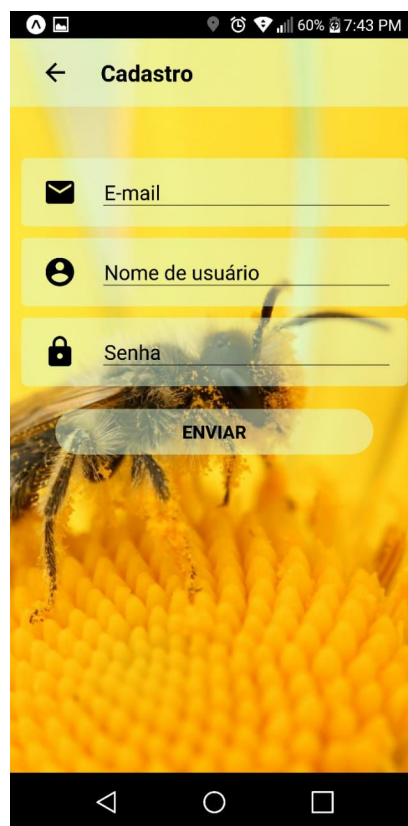


Figura 7: Cadastro

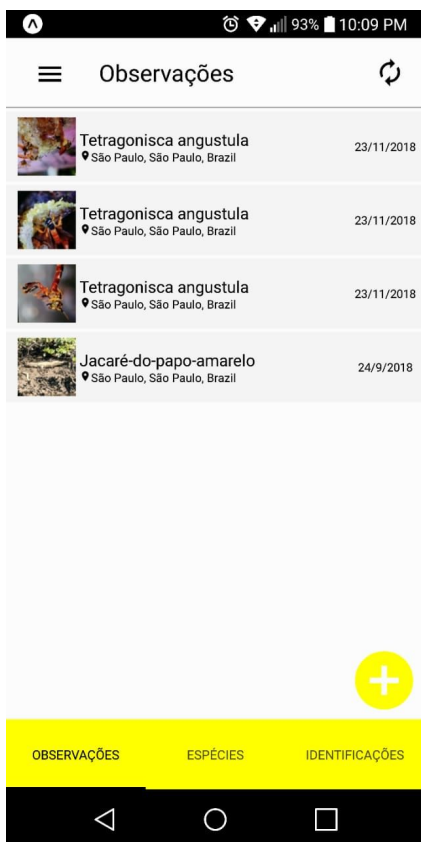


Figura 8: Observações

Feito o cadastro e o Login, o usuário entra no aplicativo e já pode começar a utilizá-lo. Logo ao iniciar, o aplicativo mostra uma tela com 3 abas. Observações, Espécies e Identificações. Nessa primeira aba ficam as observações do usuário após elas serem adicionadas

6.1.2 - Adicionar observação

Para fazer uma observação, é necessário primeiramente tirar uma ou mais fotos para a identificação da espécie. Tirada a foto, o usuário pode ou não dar um palpite de nome da espécie encontrada, informar se existe na foto abelhas, ninhos de abelhas e/ou com plantas e é possível também fazer um comentário. Por se tratar de espécies visadas, a geolocalização é por padrão obscurecida, sendo o ponto mostrado aleatoriamente em uma área com cerca de 22kmx22km da localização real.



Figura 9: Adicionar observação

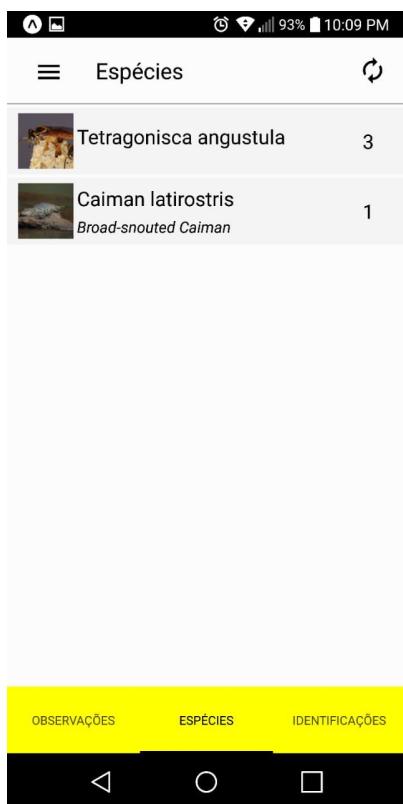


Figura 10: Espécies encontradas

Nessa tela se encontram as espécies já catalogadas pelo usuário. Sempre que é feito uma nova observação, a espécie dela é catalogada nesta seção.

6.1.3 Explorar mapa

É possível acessar o mapa de observações pelo menu lateral. Nesse mapa, o usuário pode procurar por observações de outros usuários, selecioná-lo e se desejar, propor uma nova identificação para a observação.



Figura 11 : Mapa



Figura 12: Filtros

Também existe a função de filtrar informações para se ter uma melhor busca ao explorar o mapa.

Por padrão o filtro está setado para encontrar a ordem das *Hymenoptera*, o qual as abelhas as abelhas estão contidas.

Ao ser filtrado, o mapa apenas mostra observações que se encaixam nos filtros propostos, assim sendo mais fácil de ser navegado sem se distrair com observações não relevantes para a situação.

Ao selecionar uma observação do mapa, o usuário consegue visualizar a espécie observada e o seu nível de confiança. Caso 2 ou mais pessoas concordarem com a espécie indicada, o nível de confiabilidade da marcação aumenta de Casual para Nível de pesquisa.

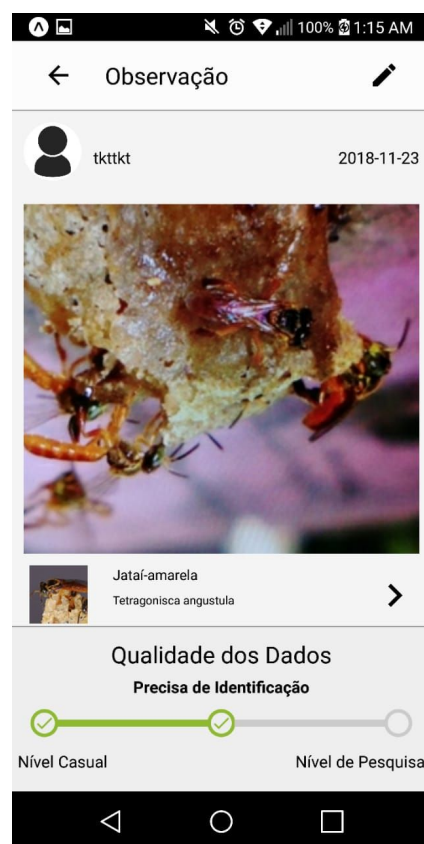


Figura 13: Selecionar observação

Em uma observação o usuário tem a possibilidade de sugerir uma identificação. Essa identificação pode fortalecer o nível da qualidade dos dados ou dar uma outra opção para aquela observação, caso o usuário ache que a espécie possa ser outra.

Figura 14: Selecionar observação[2]

O usuário vê a atual marcação da observação e pode sugerir qual espécie que a foto corresponde, seja concordando ou não com o usuário que registrou originalmente a foto.



Figura 15: Sugerir identificação

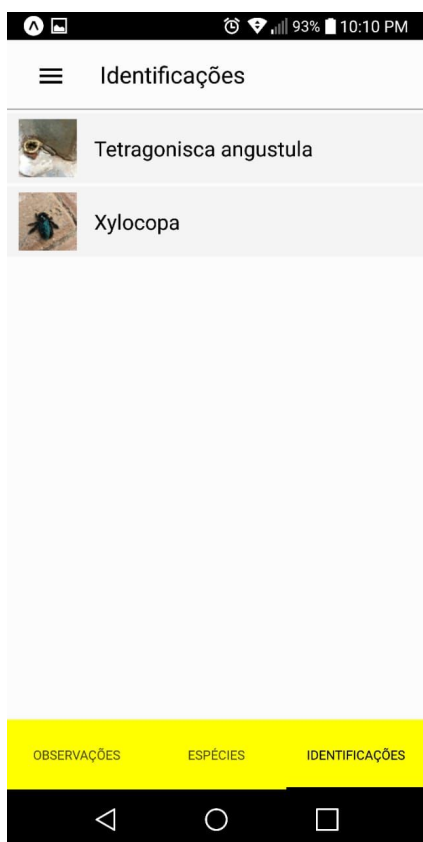


Figura 16: Identificações

Todas as identificações do usuário se concentram então na aba de “Identificação” da primeira página do aplicativo.

6.2 BackEnd

O BackEnd como dito anteriormente é um conjunto de serviços realizados tanto por um BD próprio quanto pelo BD do iNaturalist e um serviço de REST api que integra tudo.

6.2.1 LoopBack

O LB foi a framework escolhida para trabalho nesse projeto, esta framework é algo simples e que permite uma implementação de uma RESTapi simples em questão de minutos, além disso sua linguagem facilita a comunicação com os diversos access points presentes no projeto, como a RESTapi disponibilizada pelo iNaturalist.

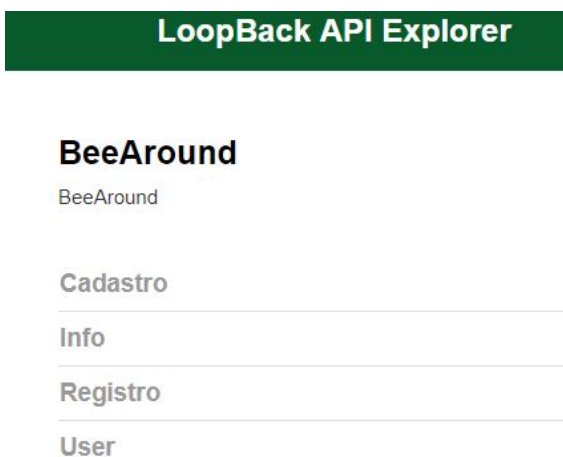


Figura 17: LoopBack Explorer

6.2.2 Cadastro de Usuários

O Cadastro de usuários, deve ser feito de uma forma a obter informações pertinentes ao proposto pelos stakeholders, para coleta de informações relacionadas às pessoas que queiram participar deste meio e utilizar este aplicativo.

The image shows a screenshot of an API client interface. It is divided into three sections: Request URL, Response Body, and Response Code. The Request URL is `http://localhost:3000/api/Cadastros`. The Response Body contains a JSON object with the following fields: `"UserID": 1010`, `"AnoNascimento": "1990-01-01T00:00:00.000Z"`, `"Escolaridade": "Ensino Superior"`, `"Residencia": ["BR", "SP", "Angatuba"]`, `"Profissao": "Estagiário"`, `"Genero": "F"`, `"Meliponicultor": true`, `"Experiencia": 1`, `"Colmeias": 10`, and `"id": "5c050da9695d41395031e397"`. The Response Code is `200`.

```
Request URL
http://localhost:3000/api/Cadastros

Response Body
{
  "UserID": 1010,
  "AnoNascimento": "1990-01-01T00:00:00.000Z",
  "Escolaridade": "Ensino Superior",
  "Residencia": [
    "BR",
    "SP",
    "Angatuba"
  ],
  "Profissao": "Estagiário",
  "Genero": "F",
  "Meliponicultor": true,
  "Experiencia": 1,
  "Colmeias": 10,
  "id": "5c050da9695d41395031e397"
}

Response Code
200
```

Figura18: Cadastro de Usuários

6.2.3 GET Cadastro

O Cadastro do Usuário pode ser trazido do BD via requisição query GET. Conforme a Figura 19, onde foi feita uma requisição GET * FROM Cadastro.

Pode ser especificar os campos conforme a Figura 18 que mostra cada campo necessário para o cadastro.



The screenshot displays a REST client interface with three sections: Request URL, Response Body, and Response Code. The Request URL is `http://localhost:3000/api/Cadastros`. The Response Body contains a JSON array with two user records. The first record has `"Colmeias": 0` and `"id": "5bffeec934ccdd49e030e31c"`. The second record has `"UserID": 1010`, `"AnoNascimento": "1990-01-01T00:00:00.000Z"`, `"Escolaridade": "Ensino Superior"`, `"Residencia": ["BR", "SP", "Angatuba"]`, `"Profissao": "Estagiário"`, `"Genero": "F"`, `"Meliponicultor": true`, `"Experiencia": 1`, `"Colmeias": 10`, and `"id": "5c050da9695d41395031e397"`. The Response Code is `200`.

```
Request URL
http://localhost:3000/api/Cadastros

Response Body
[
  {
    "Colmeias": 0,
    "id": "5bffeec934ccdd49e030e31c"
  },
  {
    "UserID": 1010,
    "AnoNascimento": "1990-01-01T00:00:00.000Z",
    "Escolaridade": "Ensino Superior",
    "Residencia": [
      "BR",
      "SP",
      "Angatuba"
    ],
    "Profissao": "Estagiário",
    "Genero": "F",
    "Meliponicultor": true,
    "Experiencia": 1,
    "Colmeias": 10,
    "id": "5c050da9695d41395031e397"
  }
]

Response Code
200
```

Figura19: Get Cadastro

6.2.4 Registros e Informações das fotos

O mesmo processo se repete para os registros e para guardar informações sobre esses registros.

O Registro é uma tabela necessária para armazenamento dos index de cada foto presente numa coleção de fotos pertinentes a um mesmo registro.

Já as informações das fotos se referem a espécie da colméia, e outras informações pertinentes

The screenshot displays the details of an HTTP request and its response. The Request URL is `http://localhost:3000/api/Infos`. The Response Body is a JSON object containing registration information. The Response Code is 200.

```
Request URL
http://localhost:3000/api/Infos

Response Body
{
  "RegistroID": "0104",
  "Caracteristica": [
    "1",
    "2",
    "3"
  ],
  "id": "5c051a50695d41395031e398"
}

Response Code
200
```

Figura 20: Informações sobre o registro

The screenshot displays the details of an HTTP request and its response. The Request URL is `http://localhost:3000/api/Registros`. The Response Body is a JSON object containing photo information for a specific user. The Response Code is 200.

```
Request URL
http://localhost:3000/api/Registros

Response Body
{
  "UserID": 1010,
  "Fotos": [
    1,
    2,
    3,
    4,
    5
  ],
  "id": "5c051aa2695d41395031e399"
}

Response Code
200
```

Figura 21: Informações sobre as fotos

Referências

Abelha. *Abelhas e Polinização*. 2018. Disponível em (<http://abelha.org.br/abelhas-e-a-polinizacao/>) acesso 26/06/2018.

Abelha. *Abelhas e a agricultura*. 2018. Disponível em (<http://abelha.org.br/abelhas-e-agricultura-2/>) acesso 26/06/2018.

Chandler, M. et al. *Contribution of citizen science towards international biodiversity monitoring*. 2017. Disponível em (<https://www.sciencedirect.com/journal/biological-conservation/vol/213/part/PB>) acesso 26/06/2018.

Dickinson, J. M. et al. *Frontiers in Ecology and the Environment*. 2012. Disponível em (<https://esajournals.onlinelibrary.wiley.com/toc/15409309/2012/10/6>) acesso 26/06/2018.

Potts, S. G. et al. *Safeguarding pollinators and their values to human well-being*. 2016. Disponível em (<https://www.nature.com/articles/nature20588>) acesso 26/06/2018.

Villas-Bôas, J. et al. *Manual Tecnológico: Mel de Abelhas sem Ferrão*. Brasília – DF. Instituto Sociedade, População e Natureza (ISPN). Brasil, 2012. ISBN: 978-85-63288-08-0.

iNaturalist API. Disponível em (<http://api.inaturalist.org/v1/docs/>). Acesso 30/11/2018.

Getting Started React Native. Disponível em (<https://facebook.github.io/react-native/docs/getting-started>). Acesso 30/11/2018.

Getting Started Expo XDE. Disponível em (<https://expo.io/learn>) Acesso 30/11/2018.

Designing a REST API with Node.js and MongoDB Atlas. Disponível em (<https://www.zoolearn.com/magazine/designing-a-rest-api-with-nodejs-and-mongodb-atlas>). Acesso 30/11/2018.

Metodologia Scrum e Agile – o que são e como aplicá-las?. Disponível em (<https://br.udacity.com/blog/post/metodologia-scrum-agile>). Acesso 30/11/2018

How to select the right React Native database. Disponível em (<https://www.simform.com/react-native-database-selection-guide/>) Acesso 30/11/2018.