



**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CENTRO TECNOLÓGICO
SISTEMAS DE INFORMAÇÃO**

JOÃO VÍCTOR BAIERLE ARANTES

**WILD.IO: SISTEMA DE MONITORAMENTO DE AVISTAMENTO DE ANIMAIS
SILVESTRES NO ESPAÇO URBANO**

Florianópolis
2023

JOÃO VÍCTOR BAIERLE ARANTES

**SISTEMA DE MONITORAMENTO DE AVISTAMENTO DE ANIMAIS
SILVESTRES NO ESPAÇO URBANO**

Proposta de Trabalho de Conclusão de Curso
apresentado à Universidade Federal de Santa
Catarina (UFSC) como parte dos requisitos
necessários para a obtenção do título de
Bacharel em Sistemas de Informação.
Orientador: Prof. José Eduardo de Lucca

Florianópolis

2023

**SISTEMA DE MONITORAMENTO DE AVISTAMENTO DE ANIMAIS
SILVESTRES NO ESPAÇO URBANO**

Trabalho de conclusão de curso submetido ao Departamento de Informática e Estatística da Universidade Federal de Santa Catarina para a obtenção do Grau de Bacharelado em Sistemas de Informação.

Orientador:

Prof. José Eduardo De Lucca
UFSC

Banca Examinadora:

Prof. Dr. Jean Carlo Rossa Hauck
UFSC

Profa. Dra. Lúcia Helena Martins Pacheco
UFSC

RESUMO

Este projeto irá trabalhar com o conceito da ciência cidadã, ou seja, vai contar com a participação ativa da comunidade para registrar e documentar as aparições e avistamentos de animais silvestres em um sistema especialmente construído para tal. No caso, a área explorada será a de avistamentos de animais silvestres nos variados espaços de cidades. Os dados coletados por meio desta plataforma podem ser utilizados para os mais diversos fins, seja para educação ambiental, conscientização para cuidados do meio ambiente, gerar atratividade turística ou mesmo tomada de decisões políticas, como por exemplo trazer mais atratividade para parques públicos que contêm uma maior fauna; saber em que áreas da cidade com vias de automóveis ocorre a maior presença de animais aos arredores, podendo criar meios de evitar acidentes, etc. Com esse TCC, o objetivo é o desenvolvimento de um aplicativo mobile que permita esses registros e a visualização dos mesmos.

Palavras-chave: Ciência cidadã; Desenvolvimento mobile; Ocupação de espaços públicos; Sistemas de Informação.

ABSTRACT

This project will work with the concept of citizen science, in other words, it will count on active community participation to register and document appearances and sightings of wild animals in a system built specially for this purpose. In this case, the explored area will be the wild animals sightings in the various areas of cities. The data collected through this platform can be used for the most varied purposes, whether it is for environmental education, environmental care awareness, create tourism attractiveness or even political decision-making such divulging public parks that contain more fauna; finding out in which city region with car traffic occurs more animal presence in the surroundings, enabling the development of approaches to avoid accidents, etc. The goal of this project is to develop a mobile application which allows these registers and its visualization.

Keywords: Citizen science; Mobile development; Public space occupation; Information systems.

LISTA DE FIGURAS

- Figura 1** - Tela inicial da página do projeto Mückenatlas
- Figura 2** - Casos de uso da plataforma
- Figura 3** - Representação da arquitetura do sistema
- Figura 4** - Tela de login do sistema
- Figuras 5 e 6** - Tela inicial do aplicativo sem e com login
- Figura 7** - Tela inicial com o alerta de localização não fornecida
- Figuras 8 e 9** - Instruções de concessão de permissão para Android e iOS
- Figura 10** - Tela de avistamentos de animais
- Figura 11** - Tela de registro de avistamentos
- Figuras 12 e 13** - Modal de informação do avistamento (com e sem imagem)
- Figura 14** - Tela de avistamentos de flora
- Figura 15** - Tela de registro de avistamentos de flora
- Figuras 16 e 17** - Modal de avistamentos de flora (com e sem imagem)
- Figuras 18 e 19** - Tela de registros de parques e modal de cada registro
- Figura 20** - Tela de registro de parques
- Figura 21** - Aviso de função exclusiva
- Figura 22** - Tela “Sobre o Aplicativo”

LISTA DE TRECHOS DE CÓDIGO

Trecho de Código 1 - Função de registro de flora

Trecho de Código 2 - Configuração das rotas da API

Trecho de Código 3 - Configuração do Mongoose

Trecho de Código 4 - Configuração do Express

Trecho de Código 5 - Função de buscar todos avistamentos de fauna

LISTA DE ABREVIações

API - Application Programming Interface

REST - Representational State Transfer

JSON - JavaScript Object Notation

UC - Unidade de Conservação

UFSC - Universidade Federal de Santa Catarina

HTTP - Hypertext Transfer Protocol

SUMÁRIO

1 INTRODUÇÃO	12
1.2 JUSTIFICATIVA	12
1.3 OBJETIVOS	13
1.3.1 Objetivo geral	13
1.3.2 Objetivos específicos	13
1.4 MÉTODO DE PESQUISA	13
1.5 RESTRIÇÕES	14
2 FUNDAMENTAÇÃO TEÓRICA	15
2.1 CIÊNCIA CIDADÃ	15
2.2 OCUPAÇÃO DE ESPAÇOS PÚBLICOS PELA SOCIEDADE	15
2.3 MORTALIDADE DE FAUNA EM REGIÕES DE AUTO ESTRADAS BRASILEIRAS	16
3 TRABALHOS CORRELATOS	17
3.1 SISTEMA URUBU	18
3.2 RENOVAR	19
3.3 MÜCKENATLAS	20
3.4 LIFELINES	23
4 DESENVOLVIMENTO DA APLICAÇÃO	25
4.1 DESCRIÇÃO DA PLATAFORMA	25
4.2 DEFINIÇÃO DE REQUISITOS	25
4.2.1 Requisitos não-funcionais	25
4.2.2 Requisitos funcionais	26
4.3 CASOS DE USO	28
4.4 TECNOLOGIAS UTILIZADAS	28
4.4.1 React Native	29
4.4.2 Expo	30
4.4.3 NodeJS	30
4.4.4 Express	31

4.4.5 Axios	31
4.4.6 MongoDB	32
4.4.7 Mongoose	32
4.4.8 React Native Maps	33
4.4.9 Firebase	34
4.4.10 CORS	34
4.5 ARQUITETURA DA PLATAFORMA	35
4.6 VISÃO GERAL DA APLICAÇÃO	37
4.6.1 Front-end	37
4.6.1.1 Tela de login	37
4.6.1.2 Tela principal da plataforma	38
4.6.1.3 Tela de avistamentos de animais	41
4.6.1.4 Tela de avistamentos de flora	43
4.6.1.5 Tela de parques	46
4.6.1.6 Aviso de login necessário	48
4.6.1.7 Tela de “Sobre o Aplicativo”	49
4.6.1.8 Envio dos dados dos formulários	49
4.6.2 Back-end	51
4.6.2.1 Criação dos modelos de cada entidade	51
4.6.2.2 Rotas da API	52
4.6.2.3 Configurações	53
4.6.2.4 Controllers da API	54
4.7 RELATOS DO DESENVOLVIMENTO	55
4.7.1 Registro de flora	56
4.7.2 Implementação do login	56
4.7.3 Intervalo de duração de exibição dos avistamentos	56
5 CONCLUSÃO	58
5.1 POSSÍVEIS DESDOBRAMENTOS	59
5.1.1 Implementação do papel de administrador	59
5.1.2 Parceria com órgãos governamentais	59

5.1.3 Gamificação	60
REFERÊNCIAS	61
APÊNDICES	64

1 INTRODUÇÃO

No ano de 2020 foi registrada a morte de 475 milhões de animais silvestres, vítimas de atropelamentos nas estradas pelo Brasil. Isso significa cerca de 17 óbitos por segundo (SISTEMA URUBU, 2020). Com um melhor mapeamento de avistamento de animais nos espaços com maior volume de trânsito, acontecimentos como esses podem ser melhor previstos e conseqüentemente evitados.

Uma pesquisa realizada em diferentes regiões metropolitanas do país pelo Instituto Semeia no ano de 2022, mostra que cerca de 70% da população da amostra frequenta parques urbanos com baixa frequência ou não frequenta. A mesma pesquisa mostra também que a motivação de “contemplar a natureza e suas belezas naturais” foi citada por 35% dos entrevistados e “mostrar a natureza para os filhos”, por 22% como motivos para visitar esses espaços (INSTITUTO SEMEIA, 2022).

Pensando em dados como esses apresentados acima e outras possíveis aplicações, a proposta deste trabalho é praticar o conceito de ciência cidadã, que é uma coparticipação de pesquisadores ou autores dos projetos, com a população como um todo através de uma metodologia participativa para a coleta dos dados utilizados no projeto (SISTEMA DE INFORMAÇÃO SOBRE A BIODIVERSIDADE BRASILEIRA, 2020).

O objetivo deste trabalho é o desenvolvimento de um aplicativo mobile que permita que os usuários façam o registro de avistamento de animais silvestres nos espaços da cidade. Com esses registros, a plataforma poderá, por exemplo, apresentar aos usuários dados como: quais regiões com faixas de automóveis ocorre a maior quantidade de avistamentos de animais silvestres, podendo assim ajudar a evitar atropelamentos; ou apontar quais parques urbanos apresentam uma maior variedade de fauna, o que pode se tornar um atrativo para o local, tendo em vista que se mostra como uma atividade muito pouco praticada nos centros metropolitanos.

1.2 JUSTIFICATIVA

Nos dias de hoje, a grande maioria das pessoas tem um *smartphone* em seu alcance a qualquer momento do dia, de acordo com a Pesquisa Anual do Uso de TI (MEIRELLES, 2022), existem cerca de 242 milhões de dispositivos móveis inteligentes no Brasil, mais que um *smartphone* por habitante. Com esse dado em mente, agregado ao elevadíssimo volume de atropelamentos de animais silvestres nas rodovias brasileiras (SISTEMA URUBU, 2020), se torna cada vez mais pertinente um aplicativo *mobile* que permita o registro do avistamento de animais silvestres em espaços urbanos a fim de mapear os pontos de maior ocorrência de determinadas espécies para que, a partir desses registros possam ser desenvolvidas estratégias de mitigação desses milhões de fatalidades.

Além de buscar a atenuação nos números de atropelamentos dos animais silvestres através desses registros, a plataforma poderá também apresentar os parques urbanos da cidade que possuam uma maior concentração de animais silvestres. De acordo com a pesquisa Parques do Brasil: Percepções da População (INSTITUTO SEMEIA, 2022), a segunda motivação mais mencionada pelos entrevistados para visitar parques urbanos é “descansar, relaxar e contemplar a natureza”. Usuários da plataforma vão saber localizar os parques com uma fauna mais farta e abundante em suas cidades, incentivando assim a ocupação desses espaços públicos pela sociedade.

A principal razão da plataforma ser desenvolvida para dispositivos móveis, além da parcela cada vez maior da internet que vem sendo ocupada por tráfego *mobile* de acordo com *web trackers* como o StatCounter (STATCOUNTER, 2022), é a compatibilidade da proposta com a possibilidade de o usuário conseguir fazer o registro na hora que avistar o animal, sendo assim possível já capturar a localização de onde o registro foi feito e eventualmente uma foto do animal, quando o usuário conseguir.

1.3 OBJETIVOS

1.3.1 Objetivo geral

Desenvolver um aplicativo mobile que contribua para a educação e conscientização ambiental relacionada a animais silvestres por meio de registro, publicação e identificação de avistamentos.

1.3.2 Objetivos específicos

- Investigar todas as possíveis aplicações dos dados e registros coletados para criar os casos de uso;
- Levantamento das melhores tecnologias e bibliotecas a serem utilizadas para a plataforma proposta;
- Desenvolver um aplicativo mobile que permita o registro do avistamento de animais silvestres em espaços urbanos;
- Publicar o código do aplicativo em um repositório público sob a licença de *open source*.

1.4 MÉTODO DE PESQUISA

O desenho metodológico que será adotado no projeto é do tipo exploratório, o que será analisado é o atual estado da arte sobre plataformas de gestão e controle do avistamento de fauna silvestre, tanto avistados em vida quanto após acidentes em sistemas com propostas similares.

Será brevemente analisado e discutido o desenvolvimento de aplicativos com a biblioteca React Native, uma biblioteca Javascript. Considerando que este tópico conta com uma comunidade significativa, soluções existentes serão mencionadas para proporcionar embasamento técnico e teórico. Por conta deste segundo ser um tema com uma grande comunidade, soluções existentes serão analisadas para fundamentação técnica e teórica.

Depois do período de análise e estudo, será iniciado o período designado para o desenvolvimento do aplicativo e do desenvolvimento da monografia.

Finalizado o desenvolvimento do aplicativo, inicia-se um período para testar o aplicativo de maneira cotidiana para encontrar possíveis pontos de melhorias, sejam elas de usabilidade ou *bugs* propriamente ditos.

1.5 RESTRIÇÕES

A plataforma concentra-se no registro de fauna e flora exclusivamente em áreas urbanas, como vias públicas, jardins e parques urbanos. Não tem como objetivo ser uma ferramenta técnica, seja para profissionais ou amadores, em ambientes selvagens ou de acesso restrito. Essa delimitação evita possíveis implicações em situações que envolvam transgressões, como invasões de áreas ambientais protegidas ou nativas.

Como o próprio nome da plataforma diz, o objetivo é documentar avistamentos exclusivamente em áreas urbanas, focando em situações nas quais os animais adentram o espaço comunitário urbano. A plataforma não tem a intenção de estimular que os usuários se desloquem até o habitat natural dos animais.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 CIÊNCIA CIDADÃ

De acordo com Mamede et al. (2017), a ciência cidadã é a contribuição voluntária da sociedade ou comunidade como um todo na coleta de dados para projetos que tenham como objetivo o desenvolvimento de novos conhecimentos científicos. A partir dos dados coletados por indivíduos voluntários, cientistas qualificados na área de domínio do projeto podem ter uma maior facilidade em encontrar respostas para os mais diversos tipos de pesquisas.

É um conceito que transcende uma área específica de estudo, pode ser utilizado nos mais diversos tipos de iniciativas, desde projetos que buscam novos conhecimentos e descobertas tecnológicas até mesmo descobertas químicas, físicas ou biológicas, por exemplo.

Na página de projetos do SiBBr (SiBBr, 2023 a), são apresentados diversos projetos de ciência cidadã capazes de exemplificar de forma prática alguns projetos de ciência cidadã. Como o SiBBr trata da biodiversidade brasileira, o foco dos projetos acaba sendo nessa área.

Dentre as plataformas utilizadas como exemplo no site, podem ser destacados projetos como o Anfíbios de Bromélia (SiBBr, 2023 b), que busca entender a composição e distribuição de anfíbios que habitam as bromélias em Santa Teresa, Espírito Santo, através da participação comunitária no envio de fotos dos espécimes encontrados; o Onde Estão as Baleias e Golfinhos?, que busca mapear a distribuição de cetáceos nas águas costeiras do Rio de Janeiro, através de relatos de avistamentos feitos pela comunidade, aprimorando assim as medidas de conservação desses animais (SiBBr, 2023 c); e o Táxeus, que é plataforma de gestão de listas de espécies que aumenta o conhecimento da biodiversidade brasileira através do registro e consulta da distribuição de animais no Brasil (SiBBr, 2023 d).

2.2 OCUPAÇÃO DE ESPAÇOS PÚBLICOS PELA SOCIEDADE

De acordo com a pesquisa realizada pelo Instituto Semeia em grandes capitais do Brasil no ano de 2022, cerca de 20% dos entrevistados não sabiam sequer

mencionar o nome de qualquer parque urbano, enquanto apenas 3% afirmou não conhecer nenhum parque natural (distante de centros urbanos), o que parece ser bastante incoerente, tendo em vista que parques urbanos são ambientes em regiões de maior circulação da população. Um dos motivos para essa discrepância é devido ao fato de que alguns parques naturais (como o Parque Nacional da Chapada Diamantina, ou o Parque Nacional da Chapada dos Veadeiros, que estão entre os mais citados na pesquisa) são amplamente divulgados em novelas ou programas de TV ou até mesmo em redes sociais por influenciadores digitais como alvos turísticos. Tendo em vista essa grande discrepância, infere-se que faltam meios de divulgação de parques urbanos, seja nas redes de televisão ou redes sociais (INSTITUTO SEMEIA, 2022).

Além do grande número de indivíduos que não sabiam mencionar nenhum parque urbano, seja de sua cidade ou de outras grandes capitais do país, o estudo apresenta também o dado que cerca de 70% da população visita parques urbanos com nenhuma ou baixíssima frequência (1 vez a cada seis meses ou menos), ou seja, mesmo entre aqueles indivíduos que conhecem os parques urbanos, a maioria não vê razões suficientes para visitar o local. Dentre as barreiras para a visita citadas pelos entrevistados, 13% alegou “não ter muita informação sobre os parques e o que oferecem”; e outros 5% afirmaram “não saber que tipo de atividade poderia ser feita em um parque desse tipo”, mostrando mais uma vez a falta de divulgação dos atrativos destes espaços.

2.3 MORTALIDADE DE FAUNA EM REGIÕES DE AUTO ESTRADAS BRASILEIRAS

No Brasil, mais de 475 milhões de vertebrados são atropelados por ano pelas autoestradas do país (BAGER et al., 2016). Como se esse número já não fosse suficientemente preocupante, evidentemente o progresso e desenvolvimento econômico do país exige malhas rodoviárias cada vez mais extensas e inevitavelmente ecologicamente invasivas, principalmente levando-se em conta que hoje em dia no Brasil, não exista uma normativa única que defina claramente quais limites de unidades de conservação devem ser respeitados pelas rodovias e estradas. Quando são construídas estradas em áreas de UCs, o mais comum é encontrar placas de sinalização de velocidade ou travessia de animais, o que evidentemente se mostra

como uma medida ineficaz.

Segundo um levantamento realizado na última década (BOTELHO, 2012), 72% das 313 UCs do Brasil são afetadas pela malha rodoviária nacional, sendo a principal vítima dessa intervenção antrópica, a Caatinga, representando 21,6% da área afetada. Evidenciando ainda mais a falta de controle legal sobre essa imprudente e inexorável invasão das áreas de preservação brasileiras.

3 TRABALHOS CORRELATOS

Este capítulo apresenta alguns outros trabalhos que foram utilizados como base para o desenvolvimento e compreensão do atual estado da arte, onde foram buscados projetos tecnológicos voltados para o meio ambiental que trabalhassem com ciência cidadã e/ou projetos que abordassem o impacto da ação antrópica sobre a fauna e flora das regiões afetadas. No caso de projetos tecnológicos, detalhes de caráter técnico das aplicações desenvolvidas também foram brevemente mencionados.

A pesquisa de trabalhos correlatos foi realizada por meio da plataforma Google Scholar¹, empregando palavras-chave pertinentes ao escopo do projeto. As principais palavras-chave utilizadas incluíram "ciência cidadã", "plataforma móvel" e "meio ambiente". Além disso, com base na literatura mais diretamente relacionada ao projeto, como o exemplo do Sistema Urubu e do projeto Renovar, foi possível aprofundar-se nas referências presentes nos próprios artigos, servindo como base para novas investigações na plataforma Google Scholar.

3.1 SISTEMA URUBU

O artigo (CASTRO; BAGER, 2019) tem como objetivo fazer breves comparações com plataformas similares e apresentar de modo geral o Sistema Urubu, plataforma desenvolvida com a finalidade de, por meio de ciência cidadã, reunir registros de: atropelamentos de animais selvagens, animais vivos cruzando vias e medidas de mitigação, como placas de aviso de passagens de fauna na estrada, dispositivos de redução de velocidade, entre outros.

Um dos maiores motivos de debate da efetividade de projetos de ciência cidadã é a confiabilidade dos dados, por isso, a maioria das atividades da área são apenas contributivas, ou seja, a sociedade participa apenas da fase da coleta, esse é um dos diferenciais do Sistema Urubu. Além dos voluntários para registros das observações, existem também os validadores voluntários da plataforma, que são especialistas da área e responsáveis por um rigoroso processo de identificação taxonômica dos animais registrados a partir das fotos enviadas pelos usuários.

¹ Saiba mais em <https://scholar.google.com>

A plataforma é dividida em alguns módulos principais, sendo eles: os de coleta e envio de dados (Urubu Mobile e a Ferramenta de Importação); os de gestão, tanto de usuários (Portal da Segurança), quanto de registros (Urubu Web) e os de análise e visualização de dados (Ferramenta de Análise e Urubu Map, respectivamente).

No Urubu Web, que é a plataforma *desktop* do projeto, existem três diferentes tipos de usuários, que são os usuários do Urubu Mobile, que também podem acessar seus registros via *desktop*; os validadores, que são os especialistas voluntários responsáveis por classificar taxonomicamente os registros; e também os gestores e administradores, que fazem a gestão dos registros recebidos tanto dos usuários (evitam registros incorretos, inadequados ou duplicados), quanto dos validadores. Cada registro pode ser avaliado por até cinco validadores e ao menos três devem chegar em um consenso na classificação taxonômica.

O Portal da Segurança, responsável pela gestão dos usuários, é onde é feita a distinção das funções e permissões de cada um dos papéis disponíveis no aplicativo. O usuário do Urubu Mobile é responsável pelo envio das fotos via aplicativo; o validador deve identificar os registros enviados pelos usuários via Urubu Web; o gestor é responsável pela gestão dos dados no Urubu Web, fazendo uma breve avaliação dos dados enviados pelos usuários e mais tarde dos dados enviados pelos validadores, além de gerenciar as outras ferramentas como o Portal de Segurança, Urubu Map (módulo responsável pela visualização dos registros) etc; já a última categoria, o administrador, além de ter as atribuições do gestor, é também responsável pelo controle de todos processos do sistema, como o CRUD e permissões de cada um dos usuários/papéis.

Os dados são inseridos pela Ferramenta de Importação através de uma planilha em Excel, a fim de facilitar potenciais integrações externas com outros projetos ou organizações interessadas.

3.2 RENOVAR

O projeto Renovar apresenta como problema a má qualidade do ar em centros urbanos, com um foco específico na cidade de Florianópolis, bem como a ausência de um sistema de monitoramento da qualidade do ar na cidade no período de desenvolvimento do trabalho. Traz também dados que evidenciam o inevitável

agravamento desse quadro nos próximos anos, como por exemplo o grande fluxo de automóveis nas ruas (SACCO FLORES ALMEIDA TEIXEIRA, 2018).

Tomando essa problemática como base, o objetivo do projeto é desenvolver dispositivos de coleta e monitoramento de qualidade do ar, um módulo web para visualizar esses dados coletados e uma documentação de um modelo escalável tomando como base os módulos previamente desenvolvidos.

Por conta de apresentar uma estrutura basilar para a exibição desses dados, viabiliza também que no futuro, indivíduos possam contribuir e apoiar o projeto com os seus próprios dispositivos de coleta. Sendo assim, pode-se afirmar que o projeto trabalha com ciência cidadã, por conta da disposição de participação voluntária de entusiastas para a produção de dados e conhecimentos científicos e trabalha também com Internet das Coisas (do inglês, *Internet of Things*, ou IoT), em virtude de contar com objetos (os dispositivos de coleta) para transmitir dados para a rede mundial de computadores.

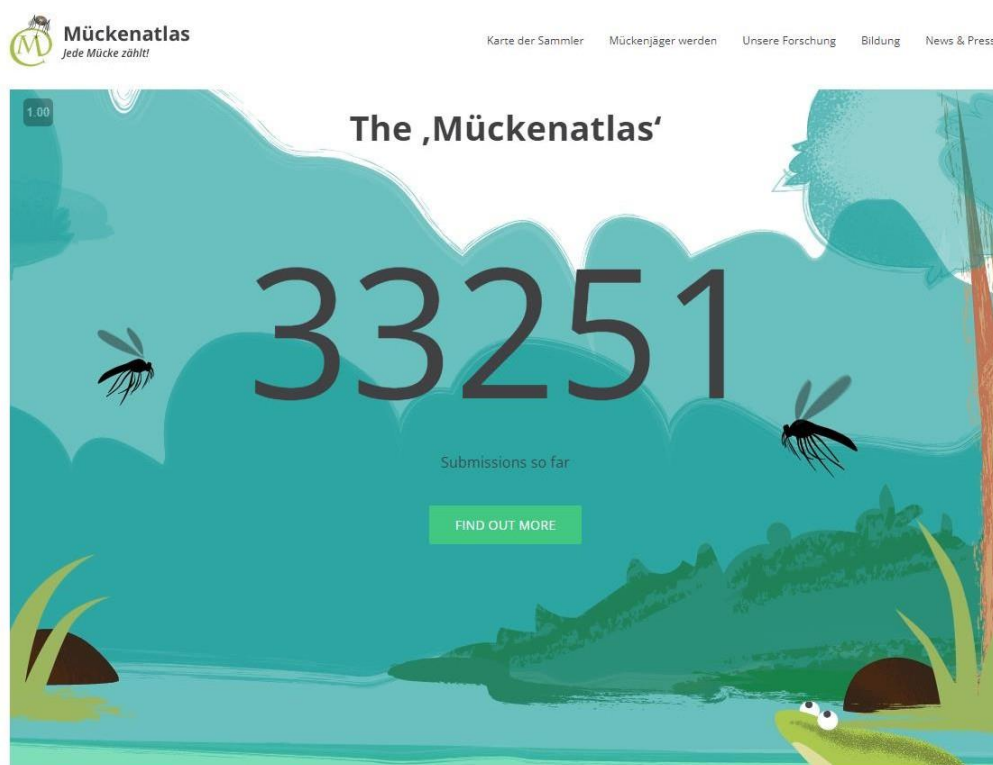
Falando do ponto de vista técnico e tecnológico, o módulo *web* foi desenvolvido usando a arquitetura RESTful, ou seja, o código-fonte do *front-end* e do *back-end* podem ser alterados de maneira independente sem interferência mútua. Para o *back-end* (ou cliente), foi utilizado o *developer kit* Java JDK para o desenvolvimento e o software brModelo para a modelagem do banco de dados, já para o *front-end* (ou servidor), o autor do projeto utilizou os *frameworks* Angular para *web* e Ionic para os híbridos para dispositivos móveis, juntamente com outras tecnologias que orbitam esses dois *frameworks*, como *Highcharts*, *Cordova* entre outros.

3.3 MÜCKENATLAS

Tendo em vista o aumento na ocorrência de espécies invasivas de mosquitos na Europa nos últimos anos, o controle desse tipo de fauna voltou à pauta nas agendas científicas europeias (PERNAT et al., 2020). Uma das importantes medidas tomadas, com o objetivo de ampliar o conhecimento científico da distribuição das espécies na Alemanha, foi iniciar um programa nacional de monitoramento de mosquitos, que é feito através de coleta profissional/científica de amostras (através de armadilhas para capturar os objetos de estudo, por exemplos) ou através de plataformas e programas de ciência cidadã.

O programa abordado no artigo é a *Mückenatlas*; as comparações feitas no artigo são: a cobertura de habitat (proporção do tipo de território onde a espécie foi capturada), a capacidade de detectar espécies invasivas (qual dos métodos capturou mais espécies exóticas) e o registro de espécies (a gama de espécies identificadas em cada método) (LEIBNIZ CENTRE FOR AGRICULTURAL LANDSCAPE RESEARCH, 2012).

Figura 1 - Tela inicial da página do projeto *Mückenatlas*



Fonte: LEIBNIZ CENTRE FOR AGRICULTURAL LANDSCAPE RESEARCH, 2023

A coleta dos mosquitos por parte dos cidadãos é feita através da captura dos indivíduos fisicamente íntegros, ou seja, sem serem esmagados. A recomendação do projeto é que os mosquitos sejam coletados em recipientes que possam ser fechados e deixados por cerca de 24 horas no freezer, essa amostra é enviada para a caixa de correios do projeto juntamente com um formulário disponível no site da iniciativa. A coleta profissional é feita através de armadilhas chamadas de BG-Sentinel Traps, que são dispositivos que liberam CO₂ para atrair e capturar as espécies.

Os resultados encontrados referente à cobertura de habitat no monitoramento passivo e ativo foram bem diferentes, enquanto o passivo abrangeu mais de 3000

municípios na Alemanha, a distribuição do tipo de território de coleta foi bastante desproporcional, sendo mais de 60% oriundas de áreas urbanas, áreas verdes urbanas e instalações de esportes e lazer; enquanto na observação ativa, apesar de abranger apenas 221 municípios, teve sua divisão em aproximadamente 48% de áreas de agricultura, 28% de áreas naturais e apenas 17% de espaços artificiais, que expressa com mais veracidade a distribuição real do solo alemão.

Relativo à espécies invasivas, o sistema de monitoramento ativo conseguiu identificar 4 indivíduos exóticos ao longo de 4 estados federais, enquanto o *Mückenatlas* recebeu registros de 6 espécies exóticas espalhadas por 10 estados federais.

Quanto à gama de espécies encontradas, a observação ativa conseguiu identificar um total de 38 espécies, enquanto a observação passiva registrou 36, porém vale ressaltar que no primeiro ano em que cada uma das observações iniciou, ambas identificaram 28 espécies e levaram mais cinco anos para alcançar seus respectivos totais de espécies encontradas, ou seja, a “taxa de identificação de novas espécies em função do tempo” foi bastante similar. As coleções compartilham mais de 70% das espécies coletadas, enquanto o monitoramento ativo identificou 4 espécies não registradas pelo passivo; o projeto de ciência cidadã registrou 2 espécies não encontradas pelas armadilhas.

Com base nos dados coletados, o estudo mostra que o *Mückenatlas*, por conta de apresentar uma mais abrangente cobertura espacial e apesar da dominância das três áreas citadas mais cedo no resumo, conseguiu apresentar um número absoluto maior de tipos de territórios que as armadilhas. Além disso, outra grande vantagem da ciência cidadã, é que consegue coletar informações dentro de propriedades privadas, que são espaços inacessíveis por cientistas, podendo ser esse um dos motivos pelo qual o monitoramento passivo conseguiu encontrar mais espécies invasivas, tendo em vista que essas espécies provavelmente vieram para o país através de ovos em plantas exóticas importadas.

Conclui-se que o monitoramento passivo tem uma efetividade comparável — em alguns pontos, até melhor — que o ativo, porém com custos reduzidos de equipamentos e menos necessidade de gerenciamento de equipe. Claro que dependendo do projeto, a proposta pode não funcionar tão bem quanto no caso objeto de estudo do artigo, que foi feito na Alemanha, um país bem desenvolvido

economicamente. Se fosse feito em um país com condições econômicas mais debilitadas, talvez a população não estivesse tão disposta a por exemplo arcar com os custos do envio das amostras ou mesmo não tivessem tempo disponível para participar de projetos como esse. Mas em se tratando de efetividade, com certeza os projetos de ciência cidadã, como o *Mückenatlas*, vêm se mostrando cada vez mais capazes de coletar dados de qualidade similares aos dados coletados profissionalmente, porém em um volume muito maior e com custo bastante reduzido.

3.4 LIFELINES

O projeto Life Lines desenvolveu um guia que tem como objetivo o desenvolvimento de uma infraestrutura verde, que promova um deslocamento e abrigo seguros para a fauna e flora por todo percurso das infraestruturas lineares (modificação de origem humanas em um ambiente florestal, como rodovias, linhas de energia etc.) (GARCIA, 2021).

O objetivo do guia é definir de maneira objetiva um processo para o planejamento, execução e manutenção de soluções que promovam maior segurança para a fauna em regiões florestais atravessadas por infraestruturas lineares.

O primeiro passo do processo é a angariação de informação preliminar, onde se identifica os locais com risco potencial a partir de registros de mortalidade, mapas de ocorrências de espécies ameaçadas; em seguida é feita a identificação das espécies mais suscetíveis ao dano a ser mitigado; então é feita a identificação das características do local (declive da estrada, sinuosidade, permeabilidade etc.).

O segundo passo é a seleção e planejamento das ações, onde primeiro seleciona-se as medidas adequadas (impedir travessias, proporcionar passagens seguras, dissuadir a fauna com refletores ou redes, etc.) levando em conta a espécie-alvo; em seguida é feito desenho e planejamento da solução, considerando a tipologia do local, extensão etc.; e então a elaboração do projeto propriamente dito.

O terceiro passo é o de execução, onde o planejamento é rigorosamente cumprido. A fase seguinte é a de manutenção, que envolve a verificação regular de integridade das soluções, assim como a realização de eventuais reparações e a gestão da vegetação para que a eficácia da implementação não seja prejudicada. O último passo é um processo iterativo de monitoramento, gestão adaptativa e

ajustamento, ou seja, monitora-se os efeitos sobre a mortalidade, abundância e deslocamento da fauna; avalia-se a eficácia e ajusta-se as soluções conforme necessário.

Outro importante artefato proveniente do guia em questão, é uma definição clara e objetiva das fichas de solução, que são literalmente documentos técnicos descrevendo a implementação. As fichas compreendem as seguintes propriedades: nome; grupo-alvo (espécies); enquadramento ecológico, que informa as condições e premissas específicas do grupo-alvo para efetividade da solução; impacto a ser minimizado, considerações técnicas, como detalhamento específico para a implantação, como tamanho, instrumentos, equipamentos etc.; requisitos de manutenção (ações necessárias e seus períodos); complementaridade, ou seja, medidas complementares que podem ser futuramente implementadas para potencialização da eficácia; alternativas; eficácia geral, que determina a capacidade da solução com base na avaliação decorrida; avaliação da solução, que tem 5 critérios específicos (dificuldade, custos de manutenção, necessidade de reabilitação, custo geral e eficácia em seus objetivos); custo-benefício (em cinco classes que variam de muito desfavorável até muito favorável); e esquemas e fotografias, para fazer uma ilustração do projeto de maneira visual.

4 DESENVOLVIMENTO DA APLICAÇÃO

Nesta seção é discutido todo o processo de desenvolvimento do projeto, desde uma descrição da plataforma pretendida, apresentação dos requisitos (funcionais e não funcionais) e dos casos de uso até quais tecnologias foram consideradas e quais as efetivamente utilizadas e uma visão geral da estrutura do código-fonte e do próprio aplicativo.

4.1 DESCRIÇÃO DA PLATAFORMA

A plataforma desenvolvida no presente projeto será uma aplicação *mobile* que vai permitir que os usuários da plataforma vejam um mapa da cidade que apresente as faixas de automóveis da região, a localização dos parques públicos urbanos e a localização dos registros de avistamentos realizados por eles mesmos e pelos demais utilizadores.

Será também possível que os usuários participem ativa e colaborativamente da plataforma, realizando esses registros dos avistamentos de animais silvestres no espaço urbano através de seus *smartphones*. Cada registro será vinculado à localização onde o mesmo foi realizado, à espécie do animal ou flora (quando identificada) com uma imagem (opcional) e descrição.

4.2 DEFINIÇÃO DE REQUISITOS

Para o levantamento de requisitos da plataforma, foi utilizada como base principalmente o Sistema Urubu, projeto de ciência cidadã para o registro do atropelamento de animais; assim como, evidentemente, os objetivos da plataforma, que foram previamente apresentados.

4.2.1 Requisitos não-funcionais

A explicação da razão da definição de cada uma das tecnologias está presente na seção 4.4.

Código	Identificação	Classificação	Descrição
RNF01	Compatibilidade com o Expo	Obrigatório	O sistema deve ser compatível com o pacote npm Expo, permitindo sua execução com a utilização deste pacote
RNF02	Disponível para Android e iOS	Obrigatório	O sistema deve ser compatível com dispositivos iOS e Android, garantindo seu funcionamento adequado em ambas as plataforma
RNF03	Utilização do React Native	Obrigatório	O sistema deve ser desenvolvido utilizando o framework React Native, baseado em JavaScript

4.2.2 Requisitos funcionais

A presente seção apresenta os requisitos funcionais do aplicativo, que foram definidos ao longo do trabalho de fundamentação teórica, mais especificamente na parte de análise de trabalhos correlatos, buscando preencher lacunas focadas no propósito da aplicação.

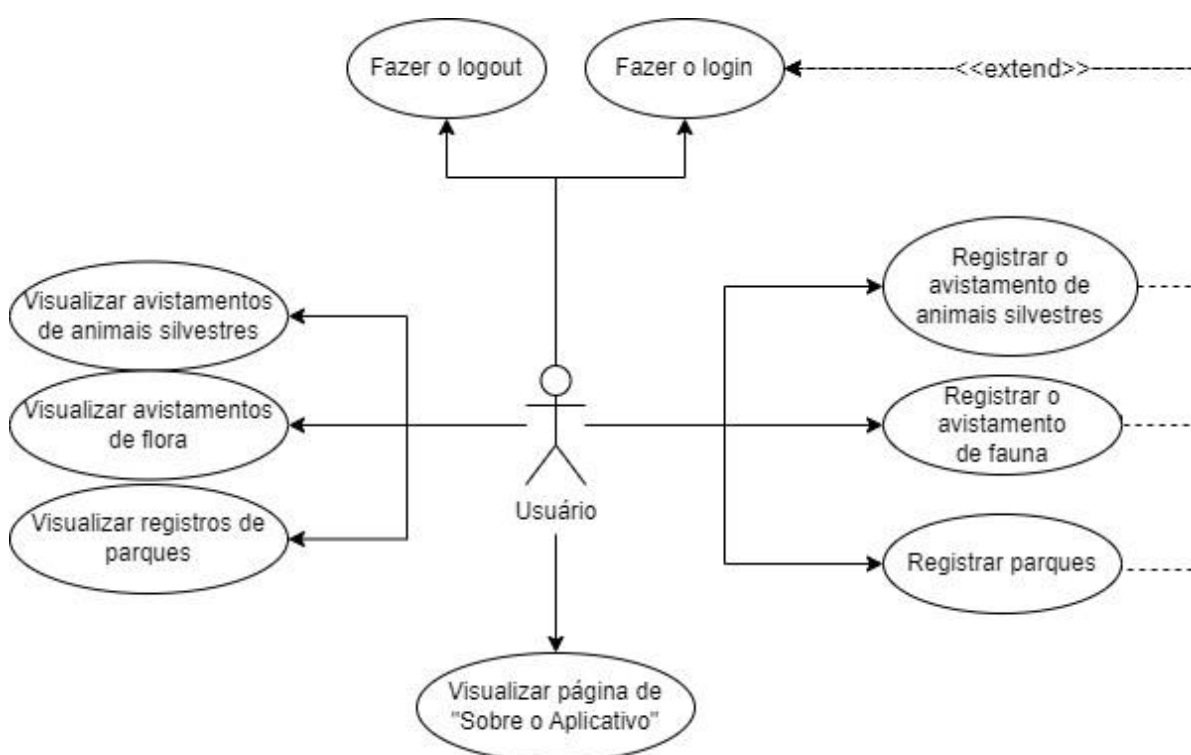
Código	Identificação	Classificação	Descrição
RF001	Fazer o login	Obrigatório	O sistema deve permitir que o usuário faça o login social via Google
RF002	Fazer o logout	Obrigatório	O sistema deve permitir que o usuário se deslogue do aplicativo
RF003	Realizar registros	Obrigatório	O sistema deve permitir que o usuário logado realize registros de parques e de avistamentos de fauna e flora
RF004	Informar se a espécie foi identificada	Obrigatório	O sistema deve permitir que o usuário logado informe, durante o registro de animais e de flora, se a espécie foi identificada ou não

RF005	Informar a espécie, quando identificada	Obrigatório	O sistema deve permitir que o usuário logado informe a espécie do animal ou planta, quando identificada, durante o registro de animais e de flora
RF006	Permitir o envio de fotos dos avistamentos de fauna e flora	Obrigatório	O sistema deve permitir que o usuário logado insira, opcionalmente, uma foto do animal ou planta durante o registro de animais e de flora
RF007	Exigir uma descrição durante o registro de fauna e flora	Obrigatório	O sistema deve exigir que o usuário logado insira uma descrição do animal ou planta durante o registro de animais e de flora
RF008	Exigir uma descrição durante o registro de parques	Obrigatório	O sistema deve exigir que o usuário logado insira uma descrição do parque sendo registrado
RF009	Exigir a localização do avistamento de fauna e flora	Obrigatório	O sistema deve exigir que o usuário logado defina o local do avistamento durante o registro de animais e de flora
RF010	Exigir a localização do registro de parques	Obrigatório	O sistema deve exigir que o usuário logado defina o local do parque durante o registro
RF011	Permitir que o usuário defina o nome do parque ao fazer um registro	Obrigatório	O sistema deve permitir que o usuário logado informe o nome do parque, se tiver a informação, durante o registro do parque
RF012	Exigir a descrição dos parques registrados	Obrigatório	O sistema deve exigir que o usuário logado insira uma descrição do parque durante o registro
RF013	Apresentar os registros realizados pelos demais usuários	Obrigatório	O sistema deve permitir que qualquer usuário visualize os registros realizados por outros usuários
RF014	Apresentar as vias da cidade no mapa	Obrigatório	O sistema deve permitir que o usuário identifique as rodovias da cidade no mapa do aplicativo
RF015	Apresentar tela de "Sobre o Aplicativo"	Desejável	O sistema deve exibir uma tela de "Sobre o Aplicativo", que contextualiza o usuário sobre o que é a plataforma

4.3 CASOS DE USO

Nesta seção, é apresentado um diagrama de casos de uso da plataforma Wild.io, que destaca as principais funcionalidades do sistema sob a perspectiva do usuário. A Figura 2 evidencia alguns aspectos determinados nos requisitos funcionais, tais como a capacidade de todos os usuários visualizarem qualquer tipo de registro, seja de avistamentos de fauna e flora ou de parques. Entretanto, para efetuar um novo registro de avistamento, o usuário deve obrigatoriamente estar autenticado na plataforma.

Figura 2 - Casos de uso da plataforma



Fonte: Elaborada pelo autor

4.4 TECNOLOGIAS UTILIZADAS

Nesta seção, será brevemente descrito quais tecnologias foram utilizadas em cada parte da arquitetura do projeto, assim como a justificativa para o uso das ferramentas. O desenho e explicação da arquitetura completa serão apresentados na próxima seção, que será inteiramente dedicada a isto.

4.4.1 React Native

A biblioteca selecionada para o desenvolvimento da interface de usuário da aplicação foi o React Native², devido à familiaridade prévia do autor com a tecnologia e sua ampla adoção no mercado. Embora não exista uma seção dedicada às tecnologias mais empregadas no desenvolvimento mobile, a Stack Overflow Developer Survey 2022 (STACK OVERFLOW, 2022) indica que o React Native (inserido na categoria "Outras bibliotecas e frameworks") ocupa a oitava posição entre as tecnologias mais utilizadas, sendo superado apenas pelo Flutter³ quando considerados somente os competidores diretos. O Flutter também foi avaliado, entretanto, devido à grande atividade de ambas as comunidades, o autor optou pelo React Native por estar mais familiarizado com a tecnologia.

O React Native é um framework baseado em JavaScript⁴ (a linguagem de programação mais utilizada globalmente, conforme a Stack Overflow Developer Survey 2022) e também é compatível com TypeScript⁵, o que facilita a busca por soluções de possíveis problemas durante o desenvolvimento. Neste projeto, o TypeScript foi empregado pois apresenta bastante semelhanças com o JavaScript, entretanto com uma tipagem mais rigorosa, evitando que erros relacionados a tipos de variáveis ou objetos sejam compilados, forçando o desenvolvedor a corrigi-los.

Um dos principais atributos do React Native é sua habilidade de renderizar o código desenvolvido de maneira nativa tanto para iOS quanto para Android, conforme a plataforma do dispositivo, sem a necessidade de empregar diferentes tecnologias no desenvolvimento de aplicativos multiplataforma, como o Wild.io.

4.4.2 Expo

² Documentação e descrição completa da biblioteca em <https://reactnative.dev>

³ Documentação e descrição completa da biblioteca em <https://flutter.dev>

⁴ Documentação e descrição completa da linguagem em <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>

⁵ Documentação e descrição completa da linguagem em <https://www.typescriptlang.org>

O Expo⁶ é um framework cujo propósito é gerenciar eficientemente o ambiente de desenvolvimento, permitindo que o desenvolvedor se concentre na lógica do código do aplicativo. Adicionalmente, oferece diversas funcionalidades que simplificam o processo de desenvolvimento, como o Hot Reloading, que permite a renderização imediata das modificações no código no dispositivo ou emulador em que a aplicação está sendo executada.

Além disso, o Expo inclui uma ampla gama de módulos pré-construídos que proporcionam acesso a recursos nativos do dispositivo, como câmera, geolocalização e notificações push, promovendo uma maior facilidade e agilidade no desenvolvimento de aplicativos. Sua compatibilidade com a biblioteca React Native permite a criação de soluções multiplataforma, otimizando a utilização de uma única base de código para desenvolver aplicativos tanto para iOS quanto para Android. Essa abordagem resulta em uma economia significativa de tempo e recursos, tornando o Expo uma escolha atrativa para desenvolvedores e equipes de desenvolvimento em todo o mundo.

4.4.3 NodeJS

O Node.js⁷ é uma plataforma de tempo de execução baseada no motor JavaScript V8 do Google Chrome, desenvolvida para permitir a execução de código JavaScript fora do contexto de navegadores. Sua principal função é facilitar o desenvolvimento de aplicações escaláveis e de alta performance, principalmente do lado do servidor (back-end). Além disso, a plataforma Node.js oferece uma vasta coleção de módulos e bibliotecas disponíveis por meio de seu gerenciador de pacotes, o npm⁸ (Node Package Manager), fomentando a reutilização de código e a colaboração entre desenvolvedores na construção de soluções robustas e sofisticadas.

Optar pelo Node.js foi uma escolha muito simples, não apenas para o autor do projeto como também para uma significativa parcela de desenvolvedores quando se fala de back-end, conforme demonstrado pela Stack Overflow Developer Survey 2022.

⁶ Documentação e descrição completa da linguagem em <https://expo.dev>

⁷ Documentação e descrição completa em <https://nodejs.org/en>

⁸ Documentação e descrição completa do gerenciador em <https://www.npmjs.com>

Segundo os resultados, o Node.js se destaca como a tecnologia mais empregada em sua categoria, representando a opção de quase 50% dos participantes que responderam ao levantamento. Criando inevitavelmente uma comunidade muito ativa, o que facilita muito na resolução de *bugs* ao longo do desenvolvimento.

4.4.4 Express

O Express⁹ facilita a construção de aplicações web e APIs robustas e eficientes, com uma estrutura simplificada e modular para lidar com rotas, solicitações e respostas no ambiente Node.js. Essa abordagem contribui para um desenvolvimento mais ágil, tirando uma carga do desenvolvedor e conseqüentemente permitindo uma maior produtividade.

Assim como as outras tecnologias selecionadas para o desenvolvimento do trabalho, o Express também se mostra muito bem posicionado entre os frameworks mais utilizados na Stack Overflow Developer Survey 2022. Apesar de ser colocado como concorrente do Node.js na pesquisa (quando na verdade, ambos são comumente empregados de maneira complementar, como no caso do Wild.io), ocupa a quarta posição entre as tecnologias mais utilizadas por desenvolvedores do segmento, o que proporciona um suporte sólido para solucionar dúvidas e enfrentar desafios durante o desenvolvimento.

4.4.5 Axios

O Axios¹⁰ proporciona uma API clara e intuitiva para realizar chamadas HTTP, facilitando a integração e comunicação entre a aplicação móvel e os serviços de back-end, o que contribui para um desenvolvimento mais ágil e produtivo.

Outro fator que fundamenta a escolha do Axios, é sua capacidade de simplificar o manuseamento de dados no formato JSON ao tratar requisições e respostas, o que está de acordo com as práticas do mercado de desenvolvimento de aplicações móveis, inclusive no Wild.io, agilizando a troca de informações entre cliente e servidor.

⁹ Documentação e descrição completa do *framework* em <https://expressjs.com>

¹⁰ Documentação e descrição completa em <https://axios-http.com>

4.4.6 MongoDB

O MongoDB¹¹ é um banco de dados NoSQL orientado a documentos, o que facilita a modelagem e armazenamento de dados em formato JSON. Então, assim como o Axios, se mantém alinhado às práticas modernas de desenvolvimento móvel no mercado (a ao próprio projeto), garantindo maior flexibilidade na manipulação das informações entre cliente e servidor.

Outro aspecto a ser considerado é a facilidade de integração do MongoDB com diversas linguagens e tecnologias, incluindo o ecossistema do React Native. Através do uso de *drivers* e bibliotecas específicas, como o Mongoose¹² (que será definido na próxima seção), os desenvolvedores podem estabelecer uma comunicação ágil entre a plataforma móvel e o banco de dados.

Além disso, o MongoDB, ocupa a quarta posição na categoria de banco de dados na Stack Overflow Developer Survey 2022, sendo o primeiro do tipo NoSQL. Assim, conta com uma comunidade expressiva e engajada, proporcionando apoio e recursos para solucionar questionamentos e superar obstáculos durante o processo de desenvolvimento.

4.4.7 Mongoose

O Mongoose é uma biblioteca que facilita a integração entre o MongoDB e o ambiente de desenvolvimento Node.js, criando mais uma camada de abstração no gerenciamento do banco de dados.

Um grande benefício do Mongoose é a capacidade de definir *schemas* para os modelos de dados, garantindo a consistência e a validação dos dados armazenados no MongoDB. Essa abordagem orientada a esquemas facilita a manutenção da estrutura do banco de dados, tornando o banco de dados mais robusto e confiável.

Apesar de não ser apresentado no Stack Overflow Developer Survey 2022, um importante ponto para a utilização dessa tecnologia é a facilidade em encontrar uma comunidade ativa utilizando a ferramenta, criando um maior suporte e recursos para solucionar dúvidas ou *bugs* durante o desenvolvimento.

¹¹ Mais informações em <https://www.mongodb.com>

¹² Mais informações em <https://mongoosejs.com>

4.4.8 React Native Maps

O React Native Maps¹³ é um módulo específico para o ambiente React Native que utiliza componentes nativos para renderizar mapas. Sendo assim, a implementação é simplificada, já que o desenvolvedor não precisa lidar com a complexidade das APIs nativas e pode utilizar a mesma base de código para ambas as plataformas. Além disso, o React Native Maps fornece uma série de funcionalidades avançadas, como geolocalização, traçado de rotas e gestos personalizados, que podem ser facilmente acessadas por meio de sua interface.

A outra opção analisada foi o OpenStreetMap¹⁴, que além de fornecer o mapa em si, fornece também dados geográficos abertos. Porém, justamente por não ser uma biblioteca desenvolvida para o ambiente React Native, a implementação da biblioteca geralmente requer a combinação de algumas outras, como o react-native-mapbox-gl¹⁵ ou o react-leaflet¹⁶, o que aumentaria o grau de complexidade da integração e manutenção do projeto.

Por consequência de utilizar os componentes nativos, o React Native Maps acaba tendo um desempenho mais fluido e otimizado, o que é muito relevante principalmente em dispositivos móveis, onde há maior limitação de hardware.

Além disso, o React Native Maps possui uma comunidade de desenvolvedores ativa e crescente, o que se traduz em uma maior facilidade na resolução de *bugs* que podem vir a surgir durante o desenvolvimento. Embora o OpenStreetMap também conte com uma comunidade engajada, seu foco está nos dados geográficos em si, e não necessariamente na integração com a plataforma React Native.

4.4.9 Firebase

O Firebase¹⁷, desenvolvido pelo Google, é uma plataforma de desenvolvimento de aplicações, tanto para Web quanto para dispositivos móveis, que oferece diversos

¹³ Mais informações em <https://github.com/react-native-maps/react-native-maps>

¹⁴ Mais informações em <https://www.openstreetmap.org>

¹⁵ Mais informações em <https://github.com/rnmapbox/maps>

¹⁶ Mais informações em <https://react-leaflet.js.org>

¹⁷ Mais informações em <https://firebase.google.com>

serviços e recursos para facilitar e acelerar o desenvolvimento, implantação e gerenciamento de aplicações. Dentre os serviços disponíveis, destaca-se o armazenamento de dados, que permite aos desenvolvedores gerenciar e armazenar arquivos, como imagens, de forma eficiente e escalável.

O Firebase Storage foi a solução adequada para o armazenamento de fotos dos animais e plantas registrados pelos usuários. A plataforma oferece um serviço de armazenamento em nuvem, otimizado para armazenar e servir arquivos multimídia de maneira rápida e segura. Além disso, o Firebase Storage é compatível com o React Native, o que facilita sua integração com a aplicação desenvolvida.

Outra opção considerada foi o Amazon S3, que possui atributos semelhantes aos já mencionados em relação ao Firebase. Entretanto, a plataforma do Google oferece vantagens no que diz respeito à política de cobrança para os desenvolvedores. Embora a AWS disponibilize um limite gratuito razoável, que dificilmente seria excedido pelo projeto em questão, ela adota o modelo "Pay-as-you-go", ou seja, caso a cota gratuita seja ultrapassada, o cartão de crédito associado ao projeto seria automaticamente cobrado para suprir as necessidades da plataforma.

Por outro lado, no caso do Firebase, caso o espaço disponível seja atingido, não haveria cobranças adicionais, e sim apenas o esgotamento do espaço alocado. Essa abordagem proporciona maior previsibilidade e controle de custos para o projeto, sem o risco de encargos inesperados, tornando-se assim um fator decisivo na escolha do Firebase em detrimento do Amazon S3.

4.4.10 CORS

O CORS¹⁸ (Cross-Origin Resource Sharing) é um mecanismo de segurança implementado pelos navegadores que permite o compartilhamento controlado de recursos entre diferentes domínios. Ele possibilita que páginas web solicitem recursos como APIs a um servidor de um domínio diferente do qual a página foi carregada.

No contexto desta aplicação, o CORS é fundamental, uma vez que a aplicação front-end e back-end (API REST) estarão hospedados em domínios distintos. Ao habilitar o CORS no back-end, é possível estabelecer uma comunicação segura entre

18

Mais informações em <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/CORS>

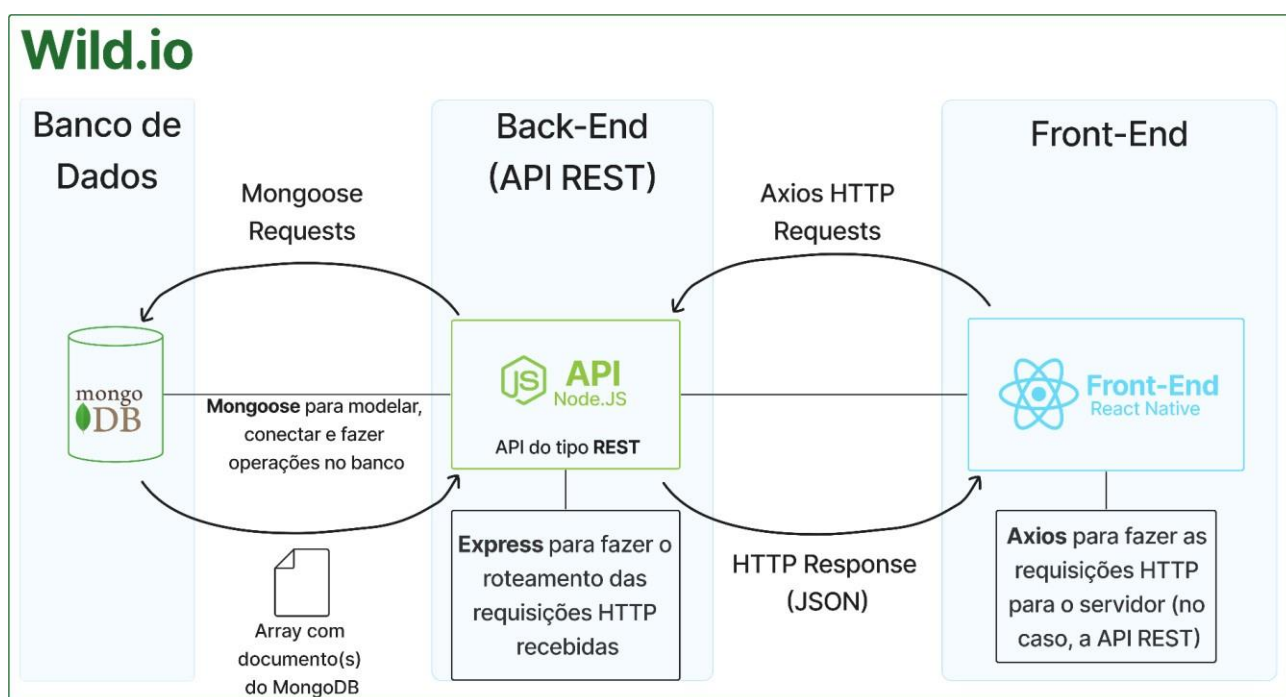
as duas partes, permitindo que o front-end acesse os recursos e serviços disponibilizados pelo back-end.

Além disso, o CORS oferece a capacidade de especificar quais domínios, métodos e cabeçalhos são permitidos nas solicitações, garantindo um maior controle e flexibilidade na definição das políticas de segurança. Dessa forma, é possível proteger a aplicação contra acessos indevidos e garantir que somente as fontes autorizadas possam interagir com os recursos providos pelo back-end.

4.5 ARQUITETURA DA PLATAFORMA

Nesta seção é explicado em mais detalhes o fluxo geral do aplicativo, mostrando como o front-end e o back-end se comunicam tanto entre si, quanto com o banco de dados.

Figura 3 - Representação da arquitetura do sistema



Fonte: elaborada pelo autor

Como apresentado na Figura 2, a plataforma é composta por três grandes seções, sendo elas: o banco de dados, o back-end e o front-end.

O banco de dados armazena todos os avistamentos de fauna silvestre e flora, assim como os parques registrados, a sua conexão e modelagem do banco na API é gerenciada pelo Mongoose.

Quanto ao back-end da aplicação, foi desenvolvido utilizando-se de uma arquitetura RESTful, sendo assim, seu funcionamento não está vinculado ao funcionamento ou até mesmo à existência de um front-end (MASSE, 2011, p. 5). Foi utilizado o Express para criar as rotas de requisição (define onde buscar os dados requisitados e tratamento de exceções).

O front-end, desenvolvido com React Native, renderiza as telas do aplicativo a partir dos dados recebidos através de um JSON. Foi utilizado o Axios para realizar as requisições HTTP para o servidor que contém as informações que serão consumidas, no caso a própria API REST descrita acima.

Para o funcionamento completo da plataforma, essas três estruturas têm que estar ativas e disponíveis ao mesmo tempo. Se o banco de dados não estiver funcionando, os demais componentes não têm dados para consumir, nem local para armazenar novos avistamentos ou parques; se a API não estiver ativa, o front-end pode ser executado, mas não conseguirá se comunicar diretamente com o banco, então não terá dados para consumir; e, finalmente, se o front-end não estiver sendo executado, os dados estarão no banco, a API vai ter as rotas ativas para receber requisições, porém nunca será acionada.

4.6 VISÃO GERAL DA APLICAÇÃO

Nesta seção, são apresentados destaques do fluxo geral da aplicação, abordando as principais funcionalidades e explicando seu funcionamento, tanto no código, quanto no aplicativo.

4.6.1 Front-end

Aqui são apresentados os aspectos relacionados à implementação do front-end em React Native do aplicativo, mostrando capturas da tela do produto final e passando por eventuais trechos relevantes de código.

4.6.1.1 Tela de login

A tela inicial da aplicação, exibida após o carregamento, permite que o usuário faça seu login. Há também a opção de prosseguir sem autenticação, restringindo as ações disponíveis à simples visualização, sem a possibilidade de realizar registros.

Figura 4 - Tela de login do sistema



Fonte: Elaborada pelo autor

A implementação do sistema de login foi realizada utilizando a autenticação social fornecida pelo Google. O processo envolve o envio de uma requisição à rota de autenticação do Google e passa alguns parâmetros na URL, sendo esses parâmetros o *client_id*, *redirect_uri*, *scope* e *response_type*.

O *client_id* consiste no identificador do projeto na plataforma Google Cloud Console, que pode ser obtido nas configurações de credenciais, na seção de APIs e Serviços. Já o *redirect_uri* especifica o local para o qual o serviço deve encaminhar o usuário após a autenticação. No contexto da aplicação desenvolvida com o Expo, o

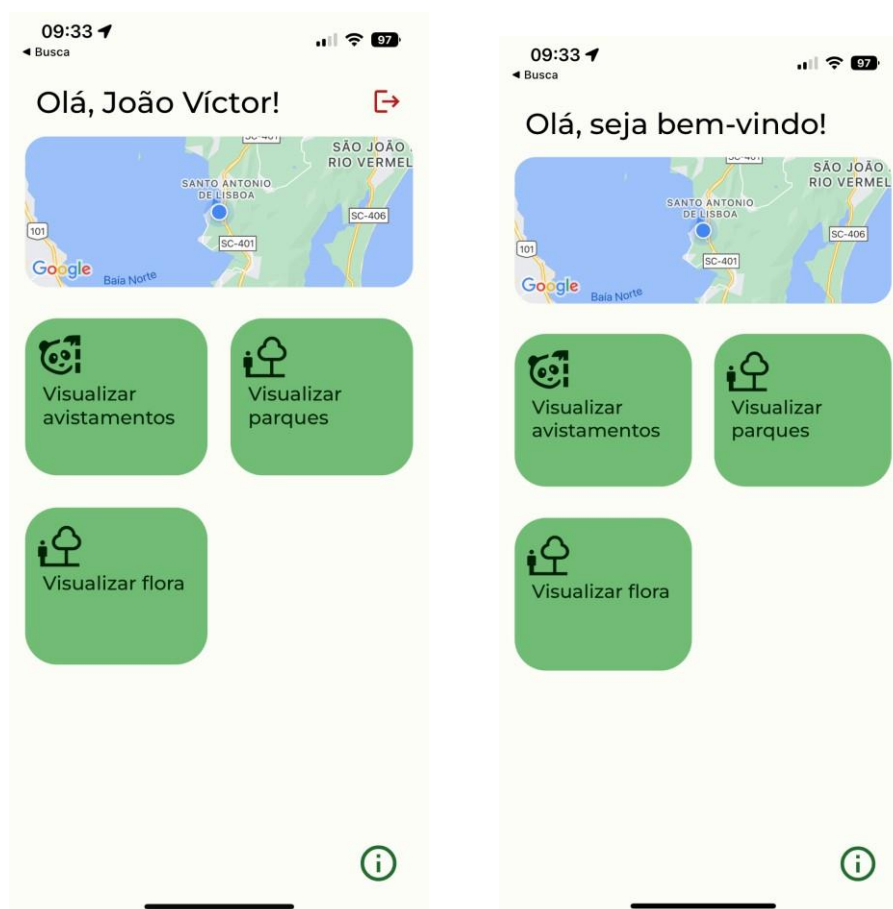
valor utilizado corresponde ao endereço do aplicativo no perfil do desenvolvedor, nas configurações do Expo.

O *scope* determina quais informações o Google deve fornecer durante a autenticação. Neste caso, solicita-se o perfil (contendo foto, nome do usuário e outros dados básicos) e o endereço de e-mail do usuário. Por fim, o *response_type* estabelece o tipo de resposta desejada do Google, sendo, neste caso, o token de autorização de acesso aos dados da conta.

4.6.1.2 Tela principal da plataforma

A tela principal da aplicação apresenta uma mensagem de boas-vindas ao aplicativo (com o nome do usuário, quando estiver logado), com um mapa com a localização do usuário, assim como os botões para visualizar e registrar avistamentos.

Figuras 5 e 6 - Tela inicial do aplicativo sem e com login



Fonte: Elaboradas pelo autor

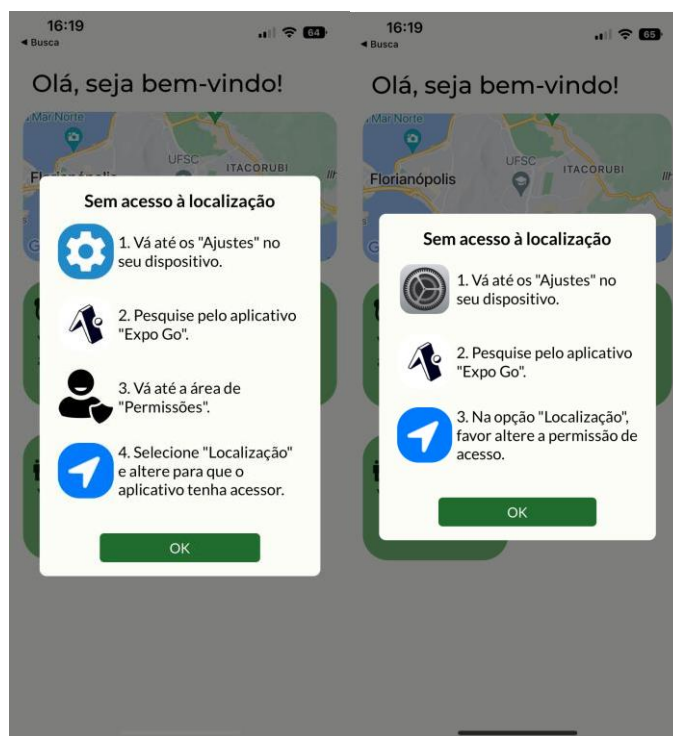
Um ponto interessante a ser comentado é a permissão de acesso à localização do usuário por parte da aplicação, que foi uma das ocasiões onde a abstração permitida pelo Expo foi fundamental. O Expo disponibiliza um pacote chamado *expo-location*, responsável por gerenciar o pedido de acesso à localização do dispositivo. Dentre os métodos disponíveis, como *getCurrentPositionAsync* e o *getLastKnownPositionAsync*, o Wild.io optou pelo segundo em virtude de sua performance, ainda que essa função possa apresentar uma ligeira variação entre a última posição conhecida e a posição real, atual. Contudo, é importante ressaltar que essa pequena variação não influencia o local do avistamento, que é informado manualmente pelo usuário. Esta diferença afeta apenas a localização do usuário identificada a cada instante.

Se o usuário não conceder a permissão de compartilhamento da localização com o aplicativo, o aviso ilustrado na Figura 6 será exibido sobre o mapa. Ao clicar neste aviso, instruções específicas para alterar a permissão, baseadas no sistema operacional do dispositivo do usuário, serão apresentadas, conforme representado nas Figuras 8 e 9.

O aplicativo informado para que o usuário acesse as configurações no passo-a-passo das Figuras 8 e 9 é o Expo Go, não o Wild.io em si. Isso ocorre por conta do Wild.io ainda não estar disponível nem na Apple Store nem na Play Store, então, de momento, a maneira de executar a plataforma é através do Expo Go, que é um aplicativo que permite que os desenvolvedores executem e testem projetos do Expo diretamente em dispositivos físicos sem a necessidade de um emulador ou a exigência de compilar o aplicativo nativo.

Figura 7 - Tela inicial com o alerta de localização não fornecida

Fonte: Elaborada pelo autor

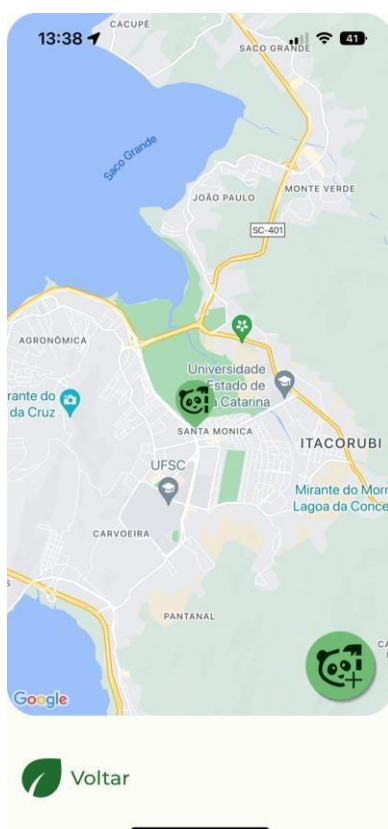
Figuras 8 e 9 - Instruções de concessão de permissão para Android e iOS

Fonte: Elaboradas pelo autor

4.6.1.3 Tela de avistamentos de animais

A tela geral de avistamentos de animais é exibida quando o usuário seleciona a opção "Visualizar avistamentos" na tela inicial do aplicativo. Neste exemplo, apresenta-se um único registro de avistamento para ilustrar de maneira clara a representação de cada avistamento no mapa.

Figura 10 - Tela de avistamentos de animais



Fonte: Elaborada pelo autor

Ao clicar no botão do canto inferior direito, o usuário é direcionado à tela de registro de avistamentos, ilustrada na Figura 11. Nessa tela, o usuário pode enviar uma foto do avistamento e deve fornecer informações sobre a identificação ou não da espécie, incluir uma descrição do evento e especificar a localização do avistamento.

Figura 11 - Tela de registro de avistamentos

13:39

Adicionar foto

Espécie identificada

Qual é a espécie?

Como é o animal que foi avistado? Busque descrever dados relevantes para identificar o animal, como tamanho, tipo de pele, coloração, presença de cauda ou asas etc.

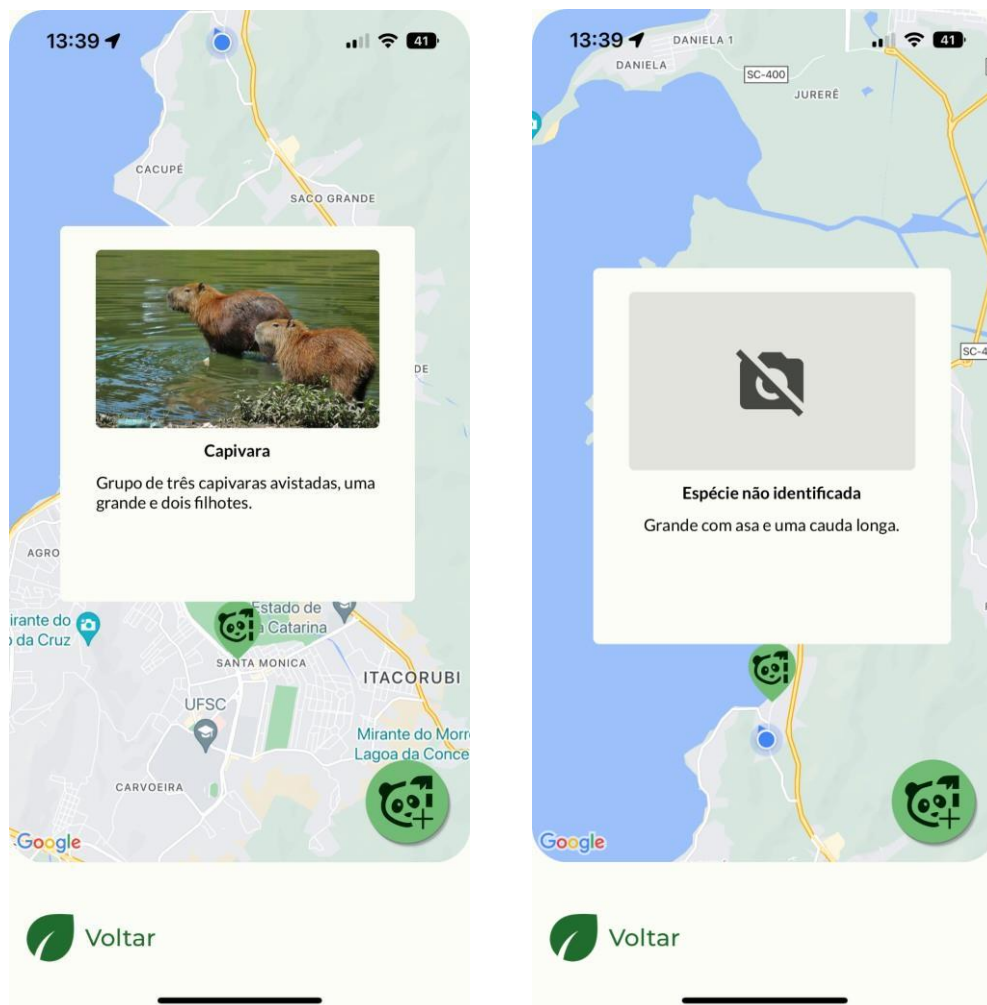
SELECIONAR LOCALIZAÇÃO DO AVISTAMENTO

REGISTRAR AVISTAMENTO

Voltar

Fonte: Elaborada pelo autor

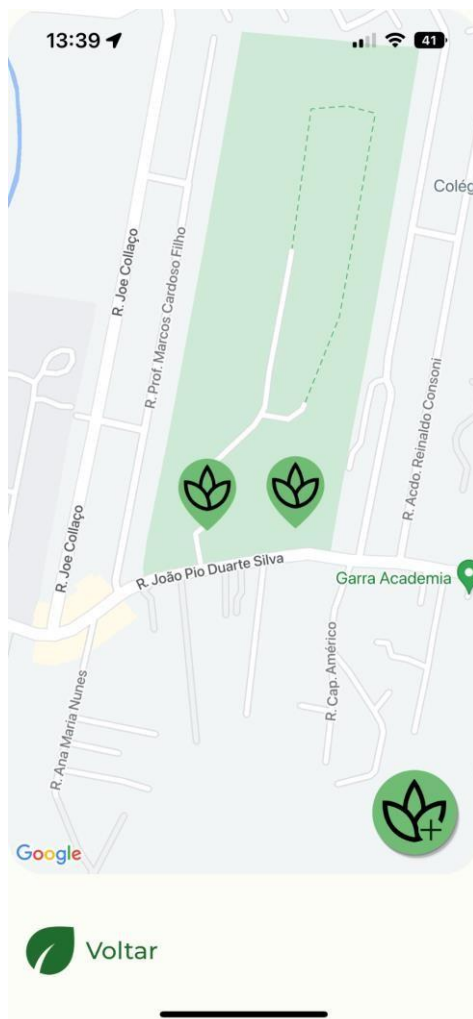
Ao selecionar um dos marcadores de avistamento no mapa, uma janela tipo modal é exibida com as informações correspondentes ao evento. Este modal apresentará a imagem capturada (caso exista), a espécie (se identificada) e a descrição fornecida pelo usuário que fez o registro.

Figuras 12 e 13 - Modal de informação do avistamento (com e sem imagem)

Fonte: Elaboradas pelo autor

4.6.1.4 Tela de avistamentos de flora

A tela de avistamento de flora é exibida ao selecionar a opção "Visualizar flora" na página inicial do aplicativo. Nesta tela, os registros de avistamento de flora são apresentados. Novamente, o exemplo ilustrado na Figura 14, existem apenas dois avistamentos registrados, com o objetivo de demonstrar claramente como cada avistamento será exibido individualmente.

Figura 14 - Tela de avistamentos de flora

Fonte: Elaborada pelo autor

Ao clicar no botão do canto inferior direito da tela, o usuário é redirecionado para a tela de registro de avistamento de flora, ilustrada na Figura 15. Nessa tela, o usuário pode enviar uma foto da flora em questão e deve informar se a espécie foi ou não identificada, incluir uma descrição da flora avistada e informar a localização do avistamento.

Figura 15 - Tela de registro de avistamentos de flora

13:40

Adicionar foto

Flora identificada

Qual é a espécie?

Como é a flora que foi avistada? Busque descrever dados relevantes para identificar a planta, como tamanho, tipo de folhagem, coloração, presença de flores ou frutos, entre outros

SELECIONAR LOCALIZAÇÃO DA FLORA

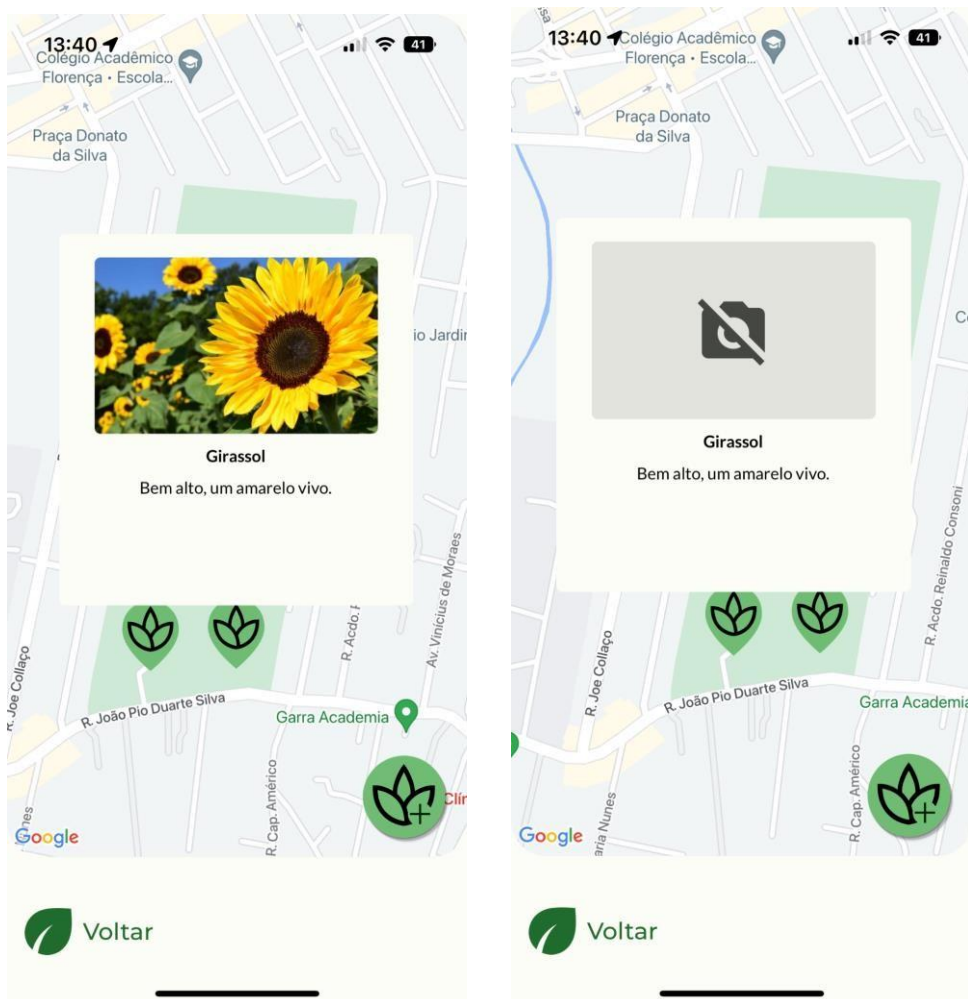
REGISTRAR FLORA

Voltar

Fonte: Elaborada pelo autor

Ao clicar em um marcador de avistamento de flora no mapa, é exibido um modal com as informações correspondentes ao registro em questão. Nesse modal será exibida a imagem capturada (quando existente), a espécie (quando identificada) e a descrição que o usuário que fez o registro forneceu.

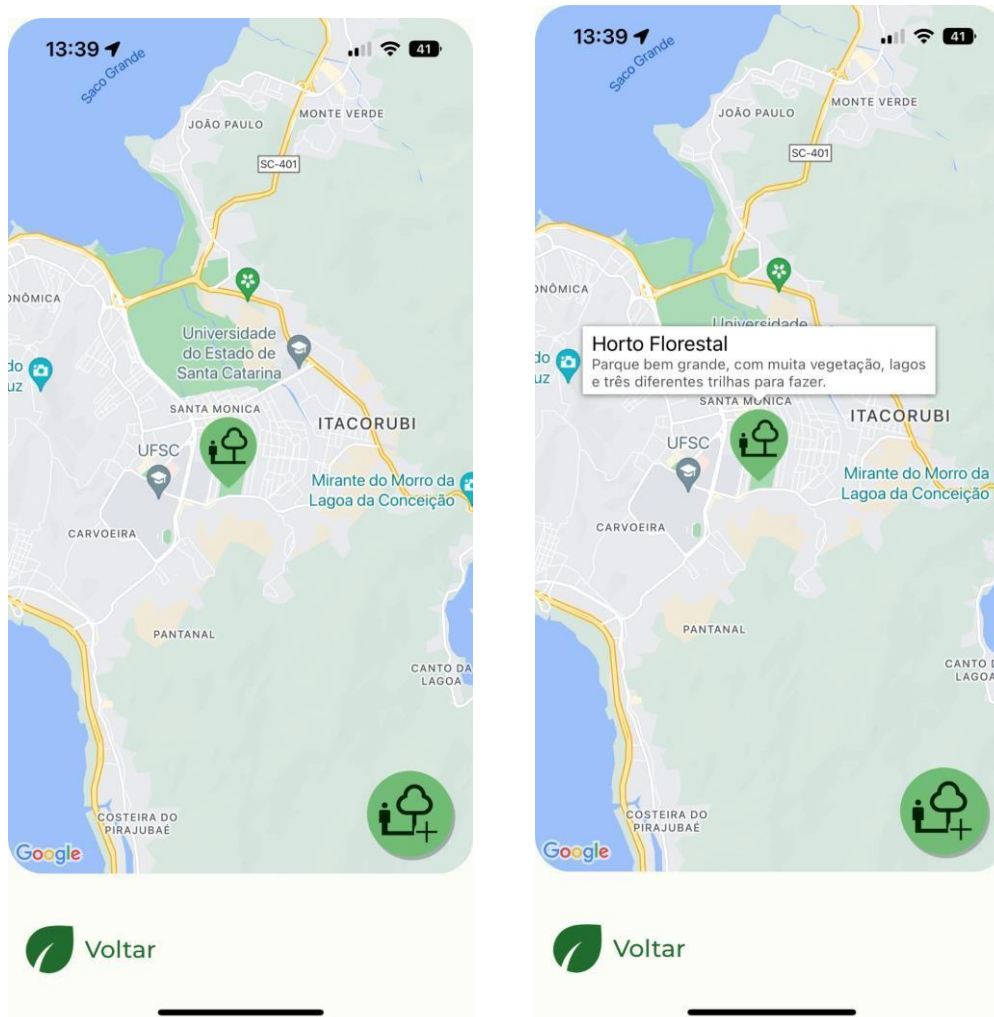
Figuras 16 e 17 - Modal de avistamentos de flora (com e sem imagem)



Fonte: Elaboradas pelo autor

4.6.1.5 Tela de parques

A tela de avistamentos de parques é apresentada ao selecionar a opção "Visualizar parques" na tela inicial do aplicativo. Nesta tela, os registros capturados de parques são exibidos. Mais uma vez, no exemplo ilustrado na Figura 18, há apenas um parque registrado, para conseguir demonstrar de forma clara a representação de cada registro no mapa. Ao clicar em um dos marcadores de parques, um modal é exibido, conforme representado na Figura 19. Este modal possui uma estrutura mais simples do que a dos avistamentos de animais, visto que não inclui imagens nem a possibilidade de não ter o nome do parque identificado.

Figuras 18 e 19 - Tela de registros de parques e modal de cada registro

Fonte: Elaboradas pelo autor

Ao selecionar o botão localizado no canto inferior direito, referente à criação de um novo registro, o usuário é direcionado à tela de registro de parques, ilustrada na Figura 20. Nessa tela, o usuário deve fornecer o nome do parque, uma descrição e a localização do mesmo.

Figura 20 - Tela de registro de parques

13:39

Qual o nome oficial do parque?

Informe aqui o bairro onde fica o parque e quaisquer detalhes relevantes que você gostaria de incluir, como a vegetação, tamanho, ano de fundação, atividades etc.

SELECIONAR LOCALIZAÇÃO DO PARQUE

REGISTRAR PARQUE

Voltar

Fonte: Elaborada pelo autor

4.6.1.6 Aviso de login necessário

O aviso representado na Figura 21 é exibido ao tentar registrar tanto um parque, quanto um avistamento de flora, quanto um avistamento de fauna. Ao invés do usuário ser redirecionado para a tela de registro da identidade, ele apenas recebe esse aviso e não é redirecionado para a tela de registro.

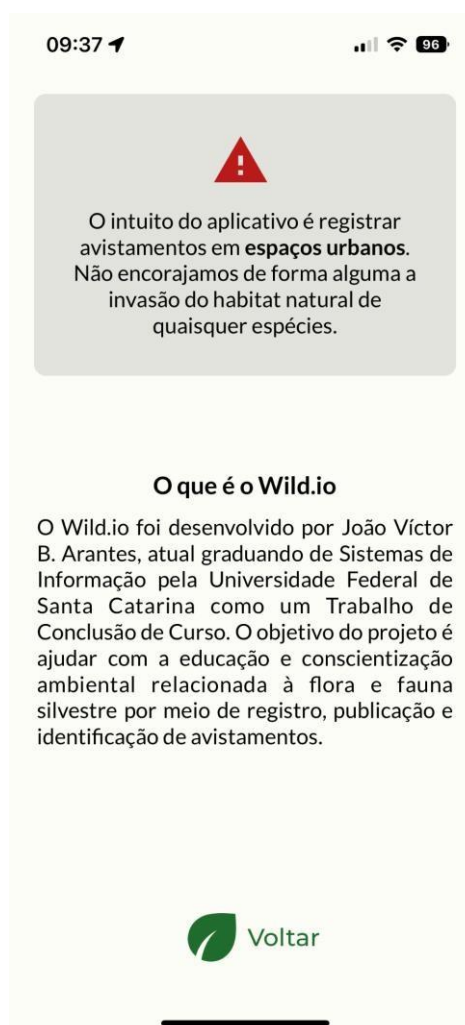
Figura 21- Aviso de função exclusiva

Fonte: Elaborada pelo autor

4.6.1.7 Tela de “Sobre o Aplicativo”

Essa é a tela que é exibida ao clicar no ícone de informação no canto inferior direito da tela principal. Essa tela foi construída para fornecer um pouco mais do contexto do aplicativo para os usuários, assim como reforçar que o projeto não tem o intuito de encorajar a invasão de habitats naturais de quaisquer espécies para realizar os registros, apenas documentar suas aparições no meio urbano.

Figura 22 - Tela “Sobre o Aplicativo”



Fonte: Elaborada pelo autor

4.6.1.8 Envio dos dados dos formulários

Um ponto importante a ser explicado é como funciona o envio de dados do front-end para a API desenvolvida para a plataforma. O trecho de código abaixo está

comentado para explicar seu funcionamento. O código em questão é referente ao envio do registro de avistamento de flora, porém os demais registros seguem lógicas bem similares.

Trecho de Código 1 - Função de registro de flora

```
async function handleSubmitFloraSighting(data: FormData) {

  // No começo são feitas algumas validações que foram
  // abstraídas para o exemplo

  // Aqui define-se a entidade do Usuário que fez o registro
  const currentLoggedUser: User = { name, email, google_id:
id };

  // Aqui, quando o registro tem uma foto, chama-se a função
  // para subir a imagem para o Firebase
  const imageUrl = image ? await uploadImageAsync(image,
`flora-${Date.now()}.jpg`) : null;

  // Abaixo é criado o objeto flora_sighting, que é o que
  // será enviado para a API para registrar no banco de dados. Ele
  // contém as informações definidas nos requisitos funcionais.
  const flora_sighting: Flora = {
    identified_species: identifiedSpecies,
    species: identifiedSpecies ? data.species : "",
    description: data.description,
    location: {
      latitude: coordinates.latitude,
      longitude: coordinates.longitude,
    },
    user: currentLoggedUser,
    photo: imageUrl as string,
    date: new Date(),
  };

  // A função abaixo vem do hook useFlora, que é o
  // componente utilizado para se comunicar com a API, que é
  // acessada pelo Axios
```

```
await saveFlora(flora_sighting);

// Embaixo ainda tem algumas funções de redirecionamento
// que foram abstraídas
}
```

4.6.2 Back-end

Nesta seção será descrito o fluxo principal da API REST (ou back-end) do sistema, passando por algumas das principais funções e configurações do ambiente.

4.6.2.1 Criação dos modelos de cada entidade

Dentro da pasta */src*, foi criada uma pasta */models*, que contém a definição de cada uma das entidades que serão recebidas pela API e enviada ao MongoDB pelo *Mongoose* através de *Mongoose Requests*.

As entidades dos modelos são definidas a partir dos *schemas* criados. *Schemas* nada mais são que estruturas que definem a forma e a organização dos dados em um banco de dados, especificando os campos, tipos de dados etc.

Os *schemas* criados foram: *floraSchema*, que define a forma dos avistamentos de flora; *parkSchema*, que define a forma dos registros de parques e *sightingsSchema*, que define a forma dos registros de avistamentos de animais silvestres.

O *floraSchema*, seguindo as definições dos requisitos funcionais, contém o um campo booleano que define se a espécie foi identificada (obrigatório), nome da espécie (apenas obrigatório se identificada), uma descrição (obrigatório), localização, que é composta pelos valores de latitude e longitude do avistamento (obrigatório), a data (obrigatório), a URL da foto (não-obrigatório) e os dados do usuário que fez o registro (o usuário é obrigatório e definido por nome, e-mail, e *google_id*).

O *parkSchema* é um pouco mais simples, contém um nome (obrigatório), uma descrição (obrigatória), a localização no mesmo modelo do *schema* anterior (obrigatório) e o usuário que fez o registro (obrigatório), também seguindo o padrão anterior de definição de Usuário

O *sightingSchema* segue a mesma lógica do de avistamento de flora, tendo em vista que são entidades com o mesmo comportamento, ou seja: um campo booleano que define se a espécie foi identificada (obrigatório); uma descrição (obrigatória) e nome da espécie, caso tenha sido identificada; a localização (obrigatória), data do registro (obrigatório), URL da foto e os dados do usuário que fez o registro (obrigatório).

4.6.2.2 Rotas da API

As rotas da API são definidas em arquivos que ficam dentro da pasta */routes*, lá é definido qual função do *controller* será utilizada para cada uma das rotas das requisições, o exemplo é a configuração das rotas de avistamentos de animais.

O nome da variável de atribuição do *controller* foi reduzida do código original para passar as rotas com mais clareza.

Trecho de Código 2 - Configuração das rotas da API

```
import express from "express";
import ctrl from "../controllers/sightingController";

const router = express.Router();

router.post("/", ctrl.createSighting);
router.get("/", ctrl.getAllSightingsLast30Days);
router.get("/all", ctrl.getAllSightings);
router.get("/:id", ctrl.getSightingById);
router.put("/:id", ctrl.updateSighting);
router.delete("/:id", ctrl.deleteSighting);
router.patch("/:id", ctrl.patchSighting);

export default router;
```

Ou seja, quando um desenvolvedor acessa o endereço da API com */sightings* no recurso:

- Se não informar nada e for do tipo POST, vai criar um novo avistamento

- Se não informar nada e for do tipo GET, vai retornar todos os avistamentos nos últimos 30 dias
- Se passar */all* com um método do tipo GET, vai retornar todos os avistamentos
- Se passar o ID de um avistamento com o método do tipo GET, a API vai retornar os dados referentes a esse avistamento específico
- Se passar o ID de um avistamento com o método do tipo PUT, a API vai atualizar os dados referentes a esse avistamento específico
- Se passar o ID de um avistamento com o método do tipo DELETE, a API vai excluir os dados referentes a esse avistamento específico
- Se passar o ID de um avistamento com o método do tipo DELETE, a API vai atualizar parcialmente os dados referentes a esse avistamento específico

4.6.2.3 Configurações

Nesta seção será brevemente explicado como foi feita a configuração do Express, do Cors e do Mongoose.

A configuração do Mongoose foi feita de acordo com as informações na documentação da biblioteca, em um arquivo separado de configuração, que utiliza a URI do MongoDB que fica no *.env* da aplicação.

Trecho de Código 3 - Configuração do Mongoose

```
import mongoose from "mongoose";

const connectDB = async () => {
  try {
    await mongoose.connect(process.env.MONGODB_URI || "");
    console.log("Successful MongoDB connection");
  } catch (error) {
    (...);
  }
};

export default connectDB;
```

Já o Express e o CORS são inicializados no arquivo principal do projeto, também de acordo com a documentação de ambas as bibliotecas.

O Express é instanciado na variável *app*, em seguida é adicionado o CORS como *middleware* da aplicação, permitindo que seja acessada por diferentes domínios e depois adicionando o *middleware* de *parser* JSON para que a aplicação possa processar corretamente as requisições com payload no formato JSON.

No quarto bloco do código são configuradas as rotas para cada um dos recursos (de avistamentos de animais, de flora e registro de parques), esses *Routers* são importados do arquivo de definição de cada uma das rotas, como apresentado na seção anterior.

Trecho de Código 4 - Configuração do Express

```
const app: Express = express();

app.use(cors());
app.use(express.json());
(...)
app.use("/api/sightings", sightingsRouter);
app.use("/api/parks", parksRouter);
app.use("/api/flora", floraRouter);
(...)
export default app;
```

4.6.2.4 *Controllers* da API

Os *controllers* são responsáveis por gerenciar a lógica relacionada aos recursos e responder às solicitações recebidas pelas rotas (que são apresentadas na seção 4.6.2.2). Para seguir na mesma linha do exemplo, a função abaixo mostra o fluxo acionado ao chamar a função *getAllSightings* (acionado por requisições do tipo GET para o recurso */sightings/all*) no *controller* de avistamento de animais.

Trecho de Código 5 - Função de buscar todos avistamentos de fauna

```
import { Request, Response } from "express";
import Sighting from "../models/sightings";

(...)

async getAllSightings(req: Request, res: Response) {
  try {
    const sightings = await Sighting.find();
    res.json(sightings);
  } catch (error) {
    (...)
  }
},

(...)
```

No trecho de código mencionado, a função recebe dois parâmetros: a requisição e a resposta, que são objetos de solicitação e resposta fornecidos pelo Express. Dentro da função, a resposta a ser retornada é atribuída à variável `sightings`. Essa variável utiliza a instância `Sighting`, definida pelo Modelo do Mongoose, que utiliza a *Mongoose Request find()* para buscar todas as entidades no banco de dados que correspondam à busca realizada.

Essa mesma lógica é aplicada para cada uma das funções de cada um dos *controllers*, que gerenciam diferentes recursos e ações dentro da aplicação. Cada controller é responsável por processar as requisições recebidas, executar a lógica de negócios apropriada, interagir com o banco de dados conforme necessário e, por fim, retornar uma resposta adequada ao cliente.

4.7 RELATOS DO DESENVOLVIMENTO

Nesta seção será apresentado um relato dos processos, desafios e realizações enfrentados durante a concepção e a implementação do sistema.

4.7.1 Registro de flora

Durante o desenvolvimento da plataforma, algumas características inicialmente planejadas para a aplicação sofreram alterações. Os recursos voltados à prevenção de atropelamentos mostraram-se pouco viáveis para implementação, uma vez que o aplicativo em si não é capaz de estabelecer medidas de mitigação, como dispositivos de redução de velocidade ou placas de aviso para passagens de fauna.

Uma vez que um dos objetivos iniciais do aplicativo se revelou menos viável sem o apoio de entidades governamentais ou ONGs, que poderiam implementar medidas com base nos dados coletados, a geração de atratividade para parques públicos urbanos tomou o protagonismo. Nesse contexto, foi adicionado o recurso de registro de flora, além da fauna previamente estabelecido. Conforme destacado na já citada pesquisa do Instituto Semeia (INSTITUTO SEMEIA, 2022), a população procura visitar parques públicos para “contemplar a natureza e suas belezas naturais”, o que inclui a flora como um elemento atrativo, além da fauna local.

4.7.2 Implementação do login

Diferentemente do Sistema Urubu, a presente plataforma não possui usuários validadores, ou seja, especialistas na área para autenticar os registros. Portanto, foi adicionado o recurso de login na plataforma, restringindo o registro de avistamentos somente a usuários logados. Dessa forma, com uma futura implementação de um usuário administrador (juntamente a uma funcionalidade de denúncia, para facilitar o processo), apesar de não controlar o conteúdo dos registros, cada avistamento é associado a um usuário específico, possibilitando a atribuição de responsabilidade pelos registros e, se necessário, a implementação de uma funcionalidade para banir usuários que possam comprometer o uso adequado da plataforma.

4.7.3 Intervalo de duração de exibição dos avistamentos

Para evitar a exibição de registros de avistamentos muito antigos, relacionados a animais que possam ter sido observados apenas uma vez devido a circunstâncias específicas, os registros exibidos em tela de fauna silvestre serão somente relativos

aos últimos 30 dias. Contudo, todos os registros de avistamentos permanecem armazenados no banco de dados e acessíveis via API, sem exclusão após o período de 30 dias, apenas deixam de ser exibidos.

Em relação às plantas avistadas, já que, naturalmente, elas não mudam de local, o período de exibição foi estabelecido em 180 dias. Isso se deve ao fato de que, apesar de não se locomoverem, a flora pode morrer ou ser removida por ações antrópicas. É importante frisar novamente que esses registros mais antigos também estarão disponíveis por meio da API, sem que haja exclusão de qualquer dado.

5 CONCLUSÃO

O objetivo deste trabalho foi o desenvolvimento de uma plataforma móvel destinada à promoção da educação e conscientização ambiental. Por meio do registro de espécies da fauna e flora silvestres, a plataforma incentiva a população a explorar os parques urbanos em suas respectivas cidades, contribuindo assim para a valorização e preservação do meio ambiente.

Durante o desenvolvimento da plataforma, foram analisados programas com objetivos semelhantes, como o Sistema Urubu (CASTRO; BAGER, 2019), que diferentemente do Wild.io, foca seus registros no atropelamento de animais silvestres, não no avistamento das espécies vivas. Para o aprofundamento no conhecimento do conceito de ciência cidadã, outros programas como o Muckenatlas (LEIBNIZ CENTRE FOR AGRICULTURAL LANDSCAPE RESEARCH, 2012) foram analisados. O aprofundamento no conhecimento de plataformas similares e no conceito basal desse trabalho, que é a ciência cidadã, foram fundamentais para o desenvolvimento de uma plataforma que cumpra os objetivos definidos.

A partir do desenvolvimento da plataforma, foi possível também a revisão e aprofundamento dos conhecimentos sobre algumas das tecnologias mais presentes no mercado, como o React Native, Node.JS e o MongoDB. Com a inclusão do sistema de autenticação e do registro de imagens, foi possível também aprofundar os conhecimentos através da integração da plataforma com os serviços do Firebase, da Google.

Conforme detalhado na seção 4.4, apesar da falta de tempo para a condução de testes públicos para alguns usuários, as escolhas tecnológicas para a construção da plataforma provaram ser apropriadas para atingir os objetivos propostos. As tecnologias selecionadas permitiram a criação de uma interface de usuário intuitiva e a construção de um banco de dados robusto para o registro de parques e avistamentos. As tecnologias utilizadas são suficientemente flexíveis para permitir possíveis novas funcionalidades ou melhorias no futuro, mantendo a robustez da estrutura. Essa compatibilidade entre as tecnologias escolhidas e os requisitos do projeto demonstra a eficácia da abordagem adotada, resultando em uma plataforma funcional e atraente para os usuários.

O código-fonte completo da plataforma desenvolvida neste trabalho pode ser acessado na página do projeto, no GitLab do autor (ARANTES, 2023).

5.1 POSSÍVEIS DESDOBRAMENTOS

Nessa seção são introduzidas algumas ideias de funcionalidades ou desdobramentos para a plataforma do ponto atual em frente.

Inicialmente, um aspecto que seria abordado neste capítulo de potenciais desdobramentos, seria a inclusão do registro de flora no sistema. Contudo, essa inclusão se mostrou necessária já na versão atual, devido à constatação das dificuldades em implementar funcionalidades efetivas para prevenir atropelamentos de fauna silvestre nas rodovias sem suporte governamental. Dessa forma, o enfoque da plataforma passou a ser, predominantemente, a geração de atratividade para os parques urbanos. Assim, além do registro de fauna, a plataforma passou a permitir também o registro de flora.

5.1.1 Implementação do papel de administrador

Como mencionado brevemente na seção 4.7.2, um bom próximo passo para a plataforma, seria a criação de um papel de administrador no sistema. Esse papel seria capaz de excluir registros indevidos, assim como banir os usuários responsáveis pelos registros em questão, evitando a exibição dos mesmos e ajudando a evitar novos registros que não seguem o intuito da plataforma.

5.1.2 Parceria com órgãos governamentais

A falta de parceria com órgãos governamentais capazes de implementar medidas efetivas de mitigação foi o principal obstáculo enfrentado em relação à funcionalidade de redução do atropelamento de fauna silvestre. Estabelecer uma colaboração com uma entidade com poder de atuação permitiria recuperar essa intenção inicial da plataforma. Uma entidade que poderia estar aberta a uma integração e tem uma maior relevância política, seria a já citada anteriormente no presente trabalho, SiBBr, que já estabeleceu uma parceria com a plataforma

iNaturalist¹⁹, que tem uma proposta similar ao Wild.io, entretanto para diferentes fins. A SiBBr consome os dados coletados na plataforma, exibe os dados e credita colaboradores. Além do SiBBr, claro que podem ser citadas diversas outras entidades como o IBAMA ou a própria polícia ambiental, que poderiam implementar as medidas de mitigação para de fato evitar atropelamentos.

Além da parceria com órgãos capazes de implementar as medidas de mitigativas, órgãos de pesquisa como o próprio Departamento de Ecologia e Zoologia da Universidade Federal de Santa Catarina podem se beneficiar desses dados para um mapeamento da presença de determinadas espécies em regiões específicas do espaço urbano, por exemplo.

Utilizando os dados coletados pela população através do aplicativo, a identificação de áreas com maior concentração de animais silvestres próximos às estradas se tornaria viável. Com base nessas informações, seria possível implementar medidas apropriadas, como a instalação de radares de velocidade e placas de aviso para passagens de animais.

Porém vale ressaltar que o público-alvo da plataforma é realmente qualquer um que tenha interesse em colaborar e contribuir com mais dados, não apenas entidades governamentais ou de órgãos de pesquisa, que têm como interesse os dados coletados pelo público geral.

5.1.3 Gamificação

A introdução de elementos de gamificação na plataforma tem um potencial significativo para amplificar o engajamento dos usuários. A gamificação, que envolve a aplicação de conceitos de design de jogos em contextos não lúdicos, poderia motivar os participantes a contribuir mais ativamente, estimulando a competição saudável e o senso de realização.

Por exemplo, a plataforma poderia incorporar um sistema de pontos, onde os usuários ganham pontos por cada nova observação registrada. Estes pontos poderiam ser utilizados para alcançar distintos níveis de 'explorador', por exemplo, ou até mesmo desbloquear distintivos virtuais, aumentando assim a sensação de progresso e realização. Da mesma forma, desafios semanais ou mensais poderiam

¹⁹ Saiba mais em: <https://www.inaturalist.org>

ser introduzidos, incentivando os usuários a visitar diferentes parques ou a documentar espécies específicas. Essas funções de gamificação, além de aumentar a participação dos usuários, também poderiam enriquecer a base de dados disponível, melhorando assim a atratividade e a valorização dos parques públicos.

REFERÊNCIAS

ARANTES, J. **Wild.io**: Sistema de Registro de Parques Públicos. Disponível em: <https://codigos.ufsc.br/joao.arantes/wild-io>. Acesso em: 15 mai. 2023.

BAGER, A. et al. Os caminhos da conservação da biodiversidade brasileira frente aos impactos da infraestrutura. **Biodiversidade Brasileira (Biobrasil)**, [S. L.], v. 6, n. 1, p. 75-86, 07 mar. 2016. Disponível em: <https://revistaeletronica.icmbio.gov.br/BioBR/article/view/530/456>. Acesso em: 23 nov. 2022.

BOTELHO RGM. 2012. Impactos das rodovias sobre as áreas protegidas no Brasil: O caso do Parque Estadual dos Três Picos (RJ), p. 21-25. In: **Encontro Científico do Parque Estadual dos Três Picos**, Cachoeiras de Macacu. CNPTIA.

CASTRO, É. P.; BAGER, Alex. SISTEMA URUBU: a ciência cidadã em prol da conservação da biodiversidade. **Revista Brasileira de Tecnologias Sociais**, [S.L.], v. 6, n. 2, p. 111, 11 out. 2019. Editora UNIVALI. <http://dx.doi.org/10.14210/rbts.v6n2.p111-130>.

COORDENADORIA DE COMUNICAÇÃO DA UFLA. **Novo aplicativo ajuda a proteger motoristas e animais contra acidentes em estradas do País**. Portal UFLA. 2021. Disponível em: <https://ufla.br/noticias/pesquisa/14501-novo-aplicativo-ajuda-a-protoger-motoristas-e-animais-contra-acidentes-em-estradas-do-pais>. Acesso em: 13 jul. 2022.

GARCIA, Graça et al. **GUIÃO DE BOAS PRÁTICAS**: soluções para minimização de impactes das estradas na fauna. [S. L.]: Universidade de Évora, 2021. 46 slides, color. Disponível em: <https://lifelines.uevora.pt/wp-content/uploads/2022/01/V3-Guião-LINES-IMPACTES-PT-300dpi.pdf>. Acesso em: 1 nov. 2022.

INSTITUTO SEMEIA. **Parques do Brasil**: Percepções da População. 2022. Instituto Semeia. 2022. Disponível em: http://semeia.org.br/arquivos/2022_Parques_do_Brasil_Percepcoes_da_Populacao.pdf. Acesso em: 10 jul. 2022.

LEIBNIZ CENTRE FOR AGRICULTURAL LANDSCAPE RESEARCH. Der Mückenatlas: deutschland kartiert die stechmücken. **Deutschland kartiert die Stechmücken**. 2012. Disponível em: <https://mueckenatlas.com/>. Acesso em: 21 out. 2022.

LEIBNIZ CENTRE FOR AGRICULTURAL LANDSCAPE RESEARCH. The ‚Mückenatlas‘. **Deutschland kartiert die Stechmücken**. 2023. Disponível em: <https://mueckenatlas.com/about/>. Acesso em: 09 jun. 2023.

MAMEDE, S.; BENITES, M.; ALHO, C.J.R. Ciência Cidadã e sua Contribuição na Proteção e Conservação da Biodiversidade na Reserva da Biosfera do Pantanal. **Revista Brasileira de Educação Ambiental (Revbea)**, [S.L.], v. 12, n. 4, p. 153-

164, 29 set. 2017. Universidade Federal de São Paulo.
<http://dx.doi.org/10.34024/revbea.2017.v12.2473>.

MASSE, M. **REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces**. [s.l.] O'Reilly Media, Inc., 2011.

MEIRELLES, F. S. **Pesquisa Anual do Uso da TI**. 33. ed. Fgv, 2022. Disponível em: <https://eaesp.fgv.br/producao-intelectual/pesquisa-anual-uso-ti>. Acesso em: 6 dez. 2022.

PERNAT, N. et al. Citizen science versus professional data collection: comparison of approaches to mosquito monitoring in germany. *Journal Of Applied Ecology*, [S.L.], v. 58, n. 2, p. 214-223, 6 out. 2020. Wiley. <http://dx.doi.org/10.1111/1365-2664.13767>.

SACCO FLORES ALMEIDA TEIXEIRA, F.. **Renovar: Um MVP para monitorar a qualidade do ar**. Repositório Institucional UFSC. Florianópolis, 2020. Disponível em: <https://repositorio.ufsc.br/handle/123456789/192181>. Acesso em: 11 out. 2022.

SiBBR. **Ciência cidadã: Anfíbios de bromélia**. Disponível em: <https://www.sibbr.gov.br/cienciacidada/anfibios-bromelia.html>. Acesso em: 31 maio 2023.

SiBBR. **Ciência cidadã: Onde estão as Baleias e Golfinhos?**. Disponível em: <https://www.sibbr.gov.br/cienciacidada/baleiasgolfinhos.html>. Acesso em: 31 maio 2023.

SiBBR. **Ciência cidadã: Táxeus**. Disponível em: <https://www.sibbr.gov.br/cienciacidada/taxeus.html>. Acesso em: 31 maio 2023.

SiBBR. **Projetos de Ciência Cidadã**. Disponível em: <https://www.sibbr.gov.br/cienciacidada/projetos.html>. Acesso em: 31 maio. 2023.

SISTEMA DE INFORMAÇÃO SOBRE A BIODIVERSIDADE BRASILEIRA. **O que é Ciência Cidadã**. SiBBR. Disponível em: sibbr.gov.br/cienciacidada/oquee.html. Acesso em: 11 jul. 2022.

SISTEMA URUBU. **Dados de Atropelamentos no Brasil**. Sistema Urubu. 2020. Disponível em: <https://sistemaurubu.com.br/dados/>. Acesso em: 10 jul. 2022.

STACK OVERFLOW. **Stack Overflow Developer Survey 2022**. Disponível em: <https://survey.stackoverflow.co/2022/>. Acesso em: 26 abr. 2023.

STATCOUNTER. **Desktop vs Mobile vs Tablet Market Share Worldwide**. [S. l.], 2022. Disponível em: <https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet>. Acesso em: 6 dez. 2022.

APÊNDICES

Apêndice A - Artigo sobre o próprio TCC (SBC)

Wild.io: Sistema de Monitoramento de Avistamento de Animais Silvestres no Espaço Urbano

João Vítor B. Arantes¹

Departamento de Informática e Estatística Universidade Federal de Santa Catarina (UFSC) – Florianópolis, SC – Brasil

joao.arantes@grad.ufsc.br

***Abstract.** This project aims to implement citizen science through the development of a mobile application that will allow the community to record and document sightings of wild animals in cities. The data collected on the platform will have various applications, such as environmental education, ecological awareness, tourist attraction, and assistance in political decision-making, for example, to improve safety in areas with a high presence of animals near motorways. The project also seeks to increase the attractiveness of public parks based on the recorded fauna.*

***Resumo.** Este projeto visa implementar a ciência cidadã por meio do desenvolvimento de um aplicativo móvel que permitirá à comunidade registrar e documentar avistamentos de animais silvestres nas cidades. Os dados coletados na plataforma terão aplicações variadas, como educação ambiental, conscientização ecológica, atração turística e auxílio na tomada de decisões políticas, por exemplo, para melhorar a segurança em áreas com alta presença de animais próximos a vias de automóveis. O projeto também busca aumentar a atratividade dos parques públicos com base na fauna registrada.*

1. Introdução

Em 2020, o Brasil registrou a morte de 475 milhões de animais silvestres em decorrência de atropelamentos nas rodovias do país, o que equivale a aproximadamente 17 óbitos por segundo (SISTEMA URUBU, 2020). Em resposta a essa alarmante estatística e à tendência de baixa frequência de visitas a parques urbanos, conforme observado pelo Instituto Semeia (2022), propõe-se a prática do conceito de ciência cidadã, com a coleta de dados participativa pelo público em geral (SISTEMA DE INFORMAÇÃO SOBRE A BIODIVERSIDADE BRASILEIRA, 2020). O objetivo é desenvolver um aplicativo móvel que permita aos usuários registrar avistamentos de animais silvestres em áreas urbanas, ajudando assim a prevenir atropelamentos e destacar parques urbanos com maior diversidade de fauna.

Justifica-se a iniciativa tendo em vista a quantidade de smartphones em uso no Brasil, com mais de 242 milhões de dispositivos (MEIRELLES, 2022), bem como a motivação para visitar parques e contemplar a natureza (INSTITUTO SEMEIA, 2022). Assim, a proposta é não apenas mitigar as fatalidades de animais silvestres, mas também fomentar o uso de espaços públicos.

Os principais objetivos do projeto incluem a contribuição para a educação e conscientização ambiental através do registro, publicação e identificação de avistamentos de animais silvestres (objetivo Geral). Além disso, também busca investigar todas as possíveis aplicações dos dados coletados, levantar as melhores tecnologias e bibliotecas a serem utilizadas, desenvolver o aplicativo móvel proposto e publicar o código do aplicativo sob a licença de open source (objetivos específicos).

A plataforma focará exclusivamente no registro de fauna e flora em áreas urbanas, evitando, assim, implicações legais e éticas associadas ao registro em ambientes selvagens ou de acesso restrito. A intenção é documentar avistamentos de animais silvestres em espaços urbanos, sem estimular o deslocamento dos usuários até os habitats naturais desses animais.

2. Fundamentação teórica

O conceito de ciência cidadã, que envolve a participação voluntária da comunidade na coleta de dados para o desenvolvimento de novos conhecimentos científicos, é essencial em diversas áreas de estudo (MAMEDE et al., 2017).

Entretanto, a ocupação dos espaços públicos pela sociedade é limitada. Uma pesquisa do Instituto Semeia (2022) mostra que cerca de 20% dos entrevistados não sabem o nome de qualquer parque urbano, e 70% da população visita esses parques com baixa frequência. Faltam meios de divulgação desses espaços e informações sobre as atividades que eles oferecem.

Além disso, a mortalidade de fauna em regiões de autoestradas brasileiras é preocupante, com mais de 475 milhões de vertebrados atropelados por ano (BAGER et al., 2016). Com a expansão das malhas rodoviárias, muitas áreas de conservação são afetadas, a exemplo da Caatinga, representando 21,6% da área afetada (BOTELHO, 2012). Esta situação indica a necessidade de um controle legal mais eficaz sobre a proteção das áreas de preservação brasileiras.

3. Trabalhos Correlatos

3.1 Sistema Urubu

O Sistema Urubu é uma plataforma de ciência cidadã voltada para a coleta de dados sobre atropelamento de animais selvagens, animais vivos cruzando vias e medidas de mitigação (CASTRO; BAGER, 2019). Ela é única em sua abordagem pois, além da coleta de dados pelos cidadãos, possui validadores voluntários especializados que realizam uma rigorosa identificação taxonômica dos animais a partir das fotos enviadas. O sistema é modular, contendo módulos para coleta e envio de dados, gestão de usuários e registros, e análise e visualização de dados. Além disso, é projetado para suportar diferentes funções de usuários, incluindo validadores, gestores e administradores.

3.2 Renovar

O Projeto Renovar visa resolver o problema da má qualidade do ar em centros urbanos, com foco em Florianópolis, onde não existia um sistema de monitoramento da qualidade do ar (SACCO FLORES ALMEIDA TEIXEIRA, 2018). O projeto desenvolve dispositivos de coleta e monitoramento de qualidade do ar, um módulo web para visualização desses dados, e documentação de um modelo escalável baseado nos módulos desenvolvidos. Além disso, o projeto incentiva a participação da ciência cidadã e trabalha com a Internet das Coisas (IoT). A estrutura técnica do módulo web foi construída usando arquitetura RESTful, com várias tecnologias para front-end e back-end.

3.3 Mückenatlas

Em resposta ao aumento de mosquitos invasivos na Europa, a Alemanha iniciou um programa de monitoramento nacional, utilizando coleta profissional e ciência cidadã, chamado Mückenatlas (PERNAT et al., 2020; LEIBNIZ CENTRE, 2012). Os cidadãos coletam mosquitos intactos e os enviam para análise, enquanto os profissionais usam armadilhas especializadas. O estudo comparou a eficácia dos métodos, cobertura de habitat, detecção de espécies invasivas e registro de espécies. Embora a coleta profissional tenha abrangido menos municípios, refletiu melhor a distribuição real do solo alemão. Enquanto isso, a ciência cidadã detectou mais espécies exóticas e teve uma maior cobertura espacial. Concluiu-se que o monitoramento passivo (ciência cidadã) é eficaz e mais econômico, embora a eficácia possa variar dependendo das condições econômicas do país.

3.4 Lifelines

O Projeto Life Lines desenvolveu um guia para promover a segurança da fauna e da flora em áreas florestais afetadas por infraestruturas lineares (GARCIA, 2021). O guia propõe um processo de cinco etapas, incluindo a coleta de informações preliminares, planejamento de ações, execução, manutenção e monitoramento contínuo. Além disso, introduz "fichas de solução" que documentam tecnicamente cada implementação, detalhando aspectos como espécies-alvo, considerações ecológicas e técnicas, eficácia, custo-benefício e ilustrações do projeto.

4. Desenvolvimento

O desenvolvimento da aplicação em questão envolve diversas etapas. O projeto centra-se na criação de um aplicativo móvel que permite aos usuários visualizar um mapa da cidade, mostrando não só as vias da cidade e parques públicos, mas também o registro de avistamentos de animais silvestres feitos por eles e por outros usuários. O aplicativo incentiva a interação ativa e colaborativa, permitindo que os usuários registrem avistamentos de animais no espaço urbano.

Os requisitos da plataforma foram estabelecidos com base em alguns outros sistemas semelhantes, mas principalmente no Sistema Urubu, um projeto de ciência cidadã voltado para o registro de atropelamentos de animais. Além disso, os objetivos próprios da plataforma foram considerados. A aplicação foi projetada com requisitos funcionais e não funcionais definidos para atender ao propósito de preencher lacunas observadas em trabalhos relacionados e abordar as necessidades específicas do aplicativo.

Os casos de uso para a plataforma Wild.io foram apresentados para destacar as principais funcionalidades do sistema a partir da perspectiva do usuário. Entre elas, está a capacidade de todos os usuários visualizarem qualquer tipo de registro, seja de avistamentos de fauna e flora ou de parques.

Diversas tecnologias foram utilizadas para desenvolver a arquitetura do projeto, cada uma escolhida com base em sua adequação ao propósito pretendido. O uso destas tecnologias foi justificado para assegurar a transparência do processo de desenvolvimento.

A arquitetura da plataforma foi explicada em detalhes, dividida em três seções principais: o banco de dados, o back-end e o front-end. O banco de dados armazena todos os avistamentos e parques registrados, o back-end utiliza uma arquitetura RESTful e o front-end é responsável por renderizar as telas do aplicativo a partir dos dados recebidos. A operação completa da plataforma requer que todas essas três seções estejam ativas e disponíveis simultaneamente.

Em seguida, o funcionamento da arquitetura da aplicação é apresentado, com destaque para as principais funcionalidades, tanto a nível de código quanto no próprio aplicativo. O processo de como os dados são armazenados, como a API processa as requisições e como o front-end renderiza as informações recebidas foi esclarecido.

Por último, todas as telas do aplicativo são minuciosamente apresentadas, descrevendo a função de cada um dos elementos apresentados em cada uma das telas, desde a tela de login até a tela de cadastro e visualização de cada uma das entidades disponíveis na plataforma (fauna, flora e parques). Além disso, o fluxo geral da API da aplicação também é apresentado, mostrando como cada entidade é salva por baixo dos panos e qual o gatilho de cada uma das rotas.

4.1. Relatos do desenvolvimento

Durante o desenvolvimento do sistema, mudanças foram necessárias para se adaptar aos desafios encontrados. Inicialmente focado na prevenção de atropelamentos de fauna, o aplicativo precisou redirecionar seu objetivo para a geração de atratividade para parques urbanos, uma vez que a implementação de medidas de mitigação mostrou-se inviável sem o apoio de entidades governamentais ou ONGs. Desta forma, adicionou-se um recurso de registro de flora, criando um novo ponto de atração, conforme sugerido pelos dados do Instituto Semeia.

Além disso, a plataforma, diferentemente do Sistema Urubu, adicionou um recurso de login, tornando o registro de avistamentos restrito a usuários logados, permitindo atribuir responsabilidade

pelos registros e a possibilidade de banir usuários que possam comprometer o uso adequado da plataforma. Para evitar a exibição de registros muito antigos, um limite foi definido: 30 dias para fauna silvestre e 180 dias para flora. Importante salientar que todos os registros permanecem acessíveis via API e não são excluídos após esses períodos.

5. Conclusão

Este trabalho alcançou com sucesso o desenvolvimento de uma plataforma móvel voltada à promoção da educação e conscientização ambiental através do registro de espécies da fauna e flora silvestres, incentivando a exploração de parques urbanos. Diversas plataformas foram analisadas, como o Sistema Urubu (CASTRO; BAGER, 2019) e o Muckenatlas (LEIBNIZ CENTRE FOR AGRICULTURAL LANDSCAPE RESEARCH, 2012), embora o Wild.io, diferentemente do Sistema Urubu, foque no avistamento de espécies vivas.

Tecnologias como React Native, Node.JS e MongoDB foram empregadas para o desenvolvimento, e a integração com serviços do Firebase permitiu a criação de um sistema de autenticação e registro de imagens. Desdobramentos futuros incluem a implementação de um papel de administrador no sistema, parcerias com órgãos governamentais e a introdução de elementos de gamificação para engajamento dos usuários. A parceria com entidades como o SiBBr, IBAMA, ou a polícia ambiental, poderia permitir a implementação de medidas de mitigação de atropelamentos de fauna silvestre. Além disso, a gamificação, envolvendo a introdução de um sistema de pontos e desafios, poderia incentivar maior participação dos usuários, enriquecendo a base de dados e aumentando a atratividade dos parques públicos.

Referências

- BOTELHO RGM. 2012. Impactos das rodovias sobre as áreas protegidas no Brasil: O caso do Parque Estadual dos Três Picos (RJ), p. 21-25. In: Encontro Científico do Parque Estadual dos Três Picos, Cachoeiras de Macacu. CNPTIA.
- CASTRO, É. P.; BAGER, Alex. SISTEMA URUBU: a ciência cidadã em prol da conservação da biodiversidade. Revista Brasileira de Tecnologias Sociais, [S.L.], v. 6, n. 2, p. 111, 11 out. 2019. Editora UNIVALI. <http://dx.doi.org/10.14210/rbts.v6n2.p111-130>.
- GARCIA, Graça et al. GUIÃO DE BOAS PRÁTICAS: soluções para minimização de impactes das estradas na fauna. [S. L.]: Universidade de Évora, 2021. 46 slides, color. Disponível em: <https://lifelines.uevora.pt/wp-content/uploads/2022/01/V3-Guião-LINES-IMPACTES-PT-300dpi.pdf>. Acesso em: 1 nov. 2022.
- INSTITUTO SEMEIA. Parques do Brasil: Percepções da População. 2022. Instituto Semeia. 2022. Disponível em:http://semeia.org.br/arquivos/2022_Parques_do_Brasil_Percepcoes_da_Populacao.pdf. Acesso em: 10 jul. 2022.
- LEIBNIZ CENTRE FOR AGRICULTURAL LANDSCAPE RESEARCH. Der Mückenatlas: deutschland kartiert die stechmücken. Deutschland kartiert die Stechmücken. 2012. Disponível em: <https://mueckenatlas.com/>. Acesso em: 21 out. 2022.
- MAMEDE, S.; BENITES, M.; ALHO, C.J.R. Ciência Cidadã e sua Contribuição na Proteção e Conservação da Biodiversidade na Reserva da Biosfera do Pantanal. Revista Brasileira de Educação Ambiental (Revbea), [S.L.], v. 12, n. 4, p. 153-164, 29 set. 2017. Universidade Federal de São Paulo. <http://dx.doi.org/10.34024/revbea.2017.v12.2473>.
- MEIRELLES, F. S. Pesquisa Anual do Uso da TI. 33. ed. Fgv, 2022. Disponível em: <https://eaesp.fgv.br/producao-intelectual/pesquisa-anual-uso-ti>. Acesso em: 6 dez. 2022.

PERNAT, N. et al. Citizen science versus professional data collection: comparison of approaches to mosquito monitoring in germany. *Journal Of Applied Ecology*, [S.L.], v. 58, n. 2, p. 214-223, 6 out. 2020. Wiley. <http://dx.doi.org/10.1111/1365-2664.13767>.

SACCO FLORES ALMEIDA TEIXEIRA, F.. Renovar: Um MVP para monitorar a qualidade do ar. Repositório Institucional UFSC. Florianópolis, 2020. Disponível em: <https://repositorio.ufsc.br/handle/123456789/192181>. Acesso em: 11 out. 2022.

SISTEMA DE INFORMAÇÃO SOBRE A BIODIVERSIDADE BRASILEIRA. O que é Ciência Cidadã. SiBBr. Disponível em: sibbr.gov.br/cienciacidada/oquee.html. Acesso em: 11 jul. 2022.

SISTEMA URUBU. Dados de Atropelamentos no Brasil. Sistema Urubu. 2020.

Disponível em: <https://sistemaurubu.com.br/dados/>. Acesso em: 10 jul. 2022.

Apêndice B - Código-fonte da aplicação

O código-fonte completo da aplicação criada ao longo do desenvolvimento do projeto Wild.io pode ser encontrado no GitLab da Universidade Federal de Santa Catarina, juntamente com instruções de instalação e execução da plataforma, disponível no endereço: <https://codigos.ufsc.br/joao.arantes/wild-io>.