



Vitor Colombo Nunes

**NOME APP: UMA FERRAMENTA
COLABORATIVA PARA ESTUDO DA
FAUNA MARINHA NO LITORAL NORTE
DO RIO GRANDE DO SUL**

Osório

2025

Vitor Colombo Nunes

**NOME APP: UMA FERRAMENTA COLABORATIVA
PARA ESTUDO DA FAUNA MARINHA NO LITORAL
NORTE DO RIO GRANDE DO SUL**

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Marcelo Paravisi

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – IFRS

Campus Osório

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Osório

2025

Vitor Colombo Nunes

**NOME APP: UMA FERRAMENTA COLABORATIVA
PARA ESTUDO DA FAUNA MARINHA NO LITORAL
NORTE DO RIO GRANDE DO SUL**

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Marcelo Paravisi

Orientador

Professor

Convidado 1

Professor

Convidado 2

Osório

2025

AGRADECIMENTOS

Os agradecimentos principais são direcionados à Gerald Weber, Miguel Frasson, Leslie H. Watter, Bruno Parente Lima, Flávio de Vasconcellos Corrêa, Otavio Real Salvador, Renato Machnievscz¹ e todos aqueles que contribuíram para que a produção de trabalhos acadêmicos conforme as normas ABNT com L^AT_EX fosse possível.

Agradecimentos especiais são direcionados ao Centro de Pesquisa em Arquitetura da Informação² da Universidade de Brasília (CPAI), ao grupo de usuários *latex-br*³ e aos novos voluntários do grupo *abnT_EX2*⁴ que contribuíram e que ainda contribuirão para a evolução do abnT_EX2.

¹ Os nomes dos integrantes do primeiro projeto abnT_EX foram extraídos de <<http://codigolivre.org.br/projects/abntex/>>

² <<http://www.cpai.unb.br/>>

³ <<http://groups.google.com/group/latex-br>>

⁴ <<http://groups.google.com/group/abntex2>> e <<http://www.abntex.net.br/>>

RESUMO

Segundo a [ABNT \(2003, 3.1-3.2\)](#), o resumo deve ressaltar o objetivo, o método, os resultados e as conclusões do documento. A ordem e a extensão destes itens dependem do tipo de resumo (informativo ou indicativo) e do tratamento que cada item recebe no documento original. O resumo deve ser precedido da referência do documento, com exceção do resumo inserido no próprio documento. (...) As palavras-chave devem figurar logo abaixo do resumo, antecedidas da expressão Palavras-chave:, separadas entre si por ponto e finalizadas também por ponto.

Palavras-chave: latex. abntex. editoração de texto.

ABSTRACT

This is the english abstract.

Keywords: latex, abntex, text editoration.

LISTA DE ILUSTRAÇÕES

Figura 1 – Fluxo de trabalho do <i>Jira</i> gerado para o desenvolvimento do projeto.	19
Figura 2 – Interface do sistema SISS-Geo.	24
Figura 3 – Recorte da região costeira do litoral norte do Rio Grande do Sul.	25
Figura 4 – Interface do Sistema Urubu.	26
Figura 5 – Captura de tela de uma página <i>web</i> de registro de visualização de ocorrências cadastradas no SIMBA.	28
Figura 6 – Captura de tela de um mapa georreferenciado gerado a partir dos dados de jornada de monitoramento cadastrado no SIMBA.	29
Figura 7 – Captura de tela da página <i>web</i> de monitoramento cadastrada no SIMBA.	30
Figura 8 – Ciclo de desenvolvimento incremental.	35
Figura 9 – Fluxograma detalhado do processo atual de coleta e registro de dados no CECLIMAR.	45
Figura 10 – Identidade visual do sistema web do projeto Fauna Marinha RS.	52
Figura 11 – Identidade visual prototipada.	53
Figura 12 – Paleta de cores utilizada.	53
Figura 13 – Protótipo da tela de login do aplicativo.	54
Figura 14 – Protótipo da tela de cadastro do aplicativo.	54
Figura 15 – Protótipo da tela inicial: usuário comum (esquerda) e pesquisador (direita).	55
Figura 16 – Protótipo das telas de "Meus Registros" com estado ativo de adição de registro (direita).	56
Figura 17 – Protótipo do formulário de registro simples. Centro: <i>bottomsheet</i> informativa; direita: campo adicional ativado.	57
Figura 18 – Protótipo do formulário de registro técnico com campos taxonômicos e detalhamento avançado.	58
Figura 19 – Protótipo da tela de registros pendentes, acessível apenas a pesquisadores.	59
Figura 20 – Protótipo da tela de análise de registro. Ao centro, visualização ampliada da imagem.	60

Figura 21 – Protótipo da visualização de registro validado.	61
Figura 22 – Protótipo da visualização de registro enviado, ainda não avaliado.	61
Figura 23 – Protótipo da tela de perfil do usuário com conquistas e histórico.	62
Figura 24 – Protótipo da tela de fauna local (esquerda) e detalhes da espécie (direita).	63
Figura 25 – Protótipo da tela de recuperação de senha: formulário (esquerda), sucesso (centro) e erro de validação (direita).	64
Figura 26 – Captura de tela do protótipo navegável desenvolvido no Figma.	65
Figura 27 – Gráfico demonstrativo de <i>cards</i> criados por mês no Jira. As barras em verde representam demandas que, até a data de criação do grá- fico, já haviam sido finalizadas. As barras em vermelho representam demandas que ainda estavam em aberto.	66
Figura 28 – Gráfico demonstrativo de <i>cards</i> vs resolvidos no Jira	67
Figura 29 – Gráfico demonstrativo de tempo de resolução médio dos <i>cards</i> por semana no Jira	68
Figura 30 – <i>Splashscreen</i> do aplicativo.	79
Figura 31 – Diagrama de fluxo do processo de autenticação e cadastro no aplicativo.	80
Figura 32 – Tela de login com formulário de entrada.	81
Figura 33 – Exemplo de erro de validação do formulário de login.	81
Figura 34 – Tela de recuperação de senha.	83
Figura 35 – Tela de recuperação de senha após e-mail enviado.	84
Figura 36 – E-mail personalizado de recuperação de senha.	84
Figura 37 – Tela de cadastro de usuário com formulário simples.	85
Figura 38 – Exemplo de mensagens de erro na validação do formulário de cadastro.	85
Figura 39 – Tela de Menu Inicial com funcionalidades liberadas para usuários comuns.	86
Figura 40 – Tela de Menu Inicial com funcionalidades exclusivas de pesquisadores.	86
Figura 41 – Tela de menu inicial com funcionalidades restritas para pesquisadores.	88
Figura 42 – Barra de navegação fixa com o início selecionado.	89
Figura 43 – Tela de meus registros apresentando os registros cadastrados pelo usuário em lista vertical.	90
Figura 44 – Tela de Meus Registros com filtro de "Validado" aplicado.	91
Figura 45 – Tela de Meus Registros com filtro de "Enviado" aplicado.	91
Figura 46 – Fluxograma de geração de novo registro.	92

Figura 47 – Fluxograma de avaliação de registro.	94
Figura 48 – Tela de Cadastrar Pesquisador.	96
Figura 49 – Fluxograma de adição de pesquisador.	97
Figura 50 – Tela de Cadastrar Pesquisador com senha aleatória gerada e mensagem de sucesso.	99
Figura 51 – <i>Bottomsheet</i> de confirmação para conceder <i>role</i> de pesquisador para o e-mail já cadastrado.	99
Figura 52 – Tela da página de fauna local do projeto Fauna Marinha RS.	100
Figura 53 – Tela de Sobre o app, apresentando <i>accordions</i> com informações do projeto e links para redes sociais.	101

LISTA DE TABELAS

Tabela 1 – Cronograma de desenvolvimento — Cinza: Planejamento inicial, Azul: Replanejamento, Vermelho: Prazos adiados	22
Tabela 2 – Dependências do Projeto <i>Flutter</i>	40
Tabela 3 – Contagem e porcentagem de ocorrências por canal de comunicação	44
Tabela 4 – Requisitos funcionais do sistema	47
Tabela 5 – Requisitos não funcionais do sistema	51
Tabela 6 – Registro de desenvolvimento geral (Ordenado por Data de Entrega)	69
Tabela 6 – (Continuação) Registro de desenvolvimento geral (Ordenado por Data de Entrega)	70
Tabela 6 – (Continuação) Registro de desenvolvimento geral (Ordenado por Data de Entrega)	71
Tabela 6 – (Continuação) Registro de desenvolvimento geral (Ordenado por Data de Entrega)	72
Tabela 7 – Registro de Correções (Fixes) (Ordenado por Data de Entrega)	72
Tabela 7 – (Continuação) Registro de Correções (Fixes) (Ordenado por Data de Entrega)	73
Tabela 7 – (Continuação) Registro de Correções (Fixes) (Ordenado por Data de Entrega)	74
Tabela 7 – (Continuação) Registro de Correções (Fixes) (Ordenado por Data de Entrega)	75
Tabela 8 – Registro de Débitos Técnicos e Melhorias (Ordenado por Data de Entrega)	75
Tabela 8 – (Continuação) Registro de Débitos Técnicos e Melhorias (Ordenado por Data de Entrega)	76
Tabela 8 – (Continuação) Registro de Débitos Técnicos e Melhorias (Ordenado por Data de Entrega)	77
Tabela 9 – Campos da coleção <code>users</code>	102
Tabela 10 – Campos da subcoleção <code>users/{id}/registers</code>	104
Tabela 11 – Campos da coleção <code>animals</code>	105

Tabela 12 – Campos da coleção <code>counters</code>	106
---	-----

LISTA DE ABREVIATURAS E SIGLAS

ABNT Associação Brasileira de Normas Técnicas

abnTeX ABsurdas Normas para TeX

SUMÁRIO

1	INTRODUÇÃO	15
2	METODOLOGIA	17
2.1	Metodologia de desenvolvimento de <i>software</i>	18
2.2	Evolução Temporal do Projeto	20
3	ESTUDO DE APLICAÇÕES CORRELATAS	23
3.1	SISS-Geo	23
3.2	Sistema Urubu	25
3.3	SIMBA	27
3.4	Análise Comparativa e Direcionamentos do Projeto	28
4	REFERENCIAL TEÓRICO	31
4.1	Ciência Cidadã	31
4.2	Agenda 2030	31
4.3	Objetivos de Desenvolvimento Sustentável (ODS)	32
4.4	Metodologia Iterativa Incremental	34
4.5	Kanban	35
4.6	Jira	35
4.7	Tecnologias e Frameworks	36
4.7.1	Flutter	36
4.7.2	Dart	37
4.7.3	Firebase	37
4.7.4	Pacotes e Plugins do Flutter	38
4.7.4.1	Hive	41
4.7.4.2	Flutter Map	41
4.7.4.3	Excel	41
4.7.4.4	Graphic	42
5	RESULTADOS OBTIDOS	43

5.1	Diagnóstico do Processo Atual	43
5.1.1	Visão Geral do Processo Atual	43
5.1.2	Descrição do Fluxo de Trabalho Atual	43
5.1.3	Problemas Identificados	45
5.2	Levantamento de Requisitos	46
5.2.1	Requisitos Funcionais	46
5.2.2	Requisitos Não Funcionais	51
5.3	Prototipação da Interface com o Usuário	52
5.3.1	Definição da Identidade Visual	52
5.3.2	Criação do Design das Telas	54
5.3.3	Desenvolvimento do Protótipo Navegável	64
5.4	Projeto Gerencial	65
5.4.1	Desenvolvimento Geral	68
5.4.2	Correções	72
5.4.3	Débitos Técnicos e Melhorias	75
5.5	Implementação do Sistema	77
5.5.1	Splashscreen	79
5.5.2	Login e Cadastro	80
5.5.3	Menu Inicial e Barra de Navegação	86
5.5.4	Meus Registros	89
5.5.5	Novo Registro	92
5.5.6	Avaliar Registro	94
5.5.7	Painel de Registros	95
5.5.8	Adicionar Pesquisador	95
5.5.9	Fauna Local	99
5.5.10	Sobre o app	101
5.5.11	Armazenamento de Dados	102
5.6	Execução de Testes e Verificação de Qualidade	106
6	CONCLUSÃO	107
	REFERÊNCIAS	108

APÊNDICE A – QUISQUE LIBERO JUSTO	112
APÊNDICE B – NULLAM ELEMENTUM URNA VEL IMPER-DIET SODALES ELIT IPSUM PHARETRA LIGULA AC PRETIUM ANTE JUSTO A NULLA CURABITUR TRISTIQUE ARCU EU METUS	113
APÊNDICE A – MORBI ULTRICES RUTRUM LOREM. . .	114
ANEXO B – CRAS NON URNA SED FEUGIAT CUM SOCIIS NATOQUE PENATIBUS ET MAGNIS DIS PARTURIENT MONTES NASCETUR RIDICULUS MUS	115
ANEXO C – FUSCE FACILISIS LACINIA DUI	116

1 INTRODUÇÃO

Este projeto de conclusão de curso é fruto de uma colaboração com o CECLIMAR (Centro de Estudos Costeiros Limnológicos e Marinhos), onde se identificou a necessidade de aprimorar o processo de coleta e monitoramento de dados da fauna costeira. Diante dos desafios atuais, propõe-se o desenvolvimento de um aplicativo de Ciência Cidadã ([MARTINS; CABRAL, 2021](#)). Este tipo de aplicativo permite que o público geral contribua com dados científicos, aumentando o alcance e a eficiência da pesquisa. Atualmente, no CECLIMAR, a coleta dos dados de monitoramento é realizada manualmente: as pessoas enviam informações via [WhatsApp \(2024\)](#), e um pesquisador do órgão encaminha os dados principais para uma bolsista, que os classifica e registra em uma planilha eletrônica, armazenando as fotos em uma pasta do [Google \(2024b\)](#). Este método manual tem levado a inconsistências nos dados e exigido revisões periódicas pelo gestor do projeto.

Diante desse cenário, a solução proposta é a criação de uma aplicação que facilite o registro e a avaliação das ocorrências do objeto de estudo, trazendo também mais visibilidade e proximidade com população que contribui com o projeto. A classificação taxonômica dos animais será feita por meio do sistema, indicando características de cada espécie bem como o estado de decomposição de cada registro. A automação desses processos visa reduzir as inconsistências e otimizar a gestão dos dados coletados. Este projeto tem como objetivo geral desenvolver um aplicativo de Ciência Cidadã para otimizar o processo de coleta, classificação e gestão de dados da fauna costeira voltado para atender as demandas de profissionais do CECLIMAR.

A aplicação possui como objetivos específicos automatizar o processo de coleta e armazenamento das ocorrências para garantir precisão dos dados e facilitar os registros de observações da fauna costeira com a ajuda da população a partir de uma interface amigável e intuitiva. Padronizar os registros para garantir um banco de dados robusto, reduzindo inconsistências e minimizando a necessidade de revisões periódicas, e facilitar a realização de pesquisas e a análise de dados recebidos, liberando recursos para outras atividades de pesquisa. Além de promover a participação ativa da comunidade na conservação da biodiversidade costeira e no monitoramento ambiental.

Com este contexto, podemos afirmar que este trabalho possui seu desenvolvimento alinhado com a Agenda 2030 da ONU ([NACÕES UNIDAS, 2015a](#)), usando a integração e aplicação de tecnologias no desenvolvimento sustentável, visando abranger os itens 14 (Vida na água), 15 (Vida terrestre) e 9 (Indústria, inovação e infraestrutura). Além disso, o projeto busca promover a difusão e aplicação dos princípios da ciência cidadã, ao facilitar a colaboração entre a comunidade e cientistas. A ciência cidadã amplia a participação pública na pesquisa científica, proporcionando uma abordagem colaborativa e inclusiva na gestão ambiental. Portanto, este trabalho busca não apenas oferecer soluções práticas para o monitoramento da fauna na região costeira do Rio Grande do Sul, mas também promover uma mudança de paradigma na forma como a ciência é realizada, enfatizando a importância da participação e colaboração da comunidade na construção de um futuro sustentável.

2 METODOLOGIA

Para a realização do embasamento deste trabalho, foram utilizadas tanto a metodologia de pesquisa bibliográfica quanto a de pesquisa documental. A primeira foi essencial para o levantamento de metodologias já consolidadas e amplamente estudadas, como as que serão abordadas no referencial teórico e, a seguir, nesta seção. Já a segunda foi empregada para identificar diferentes aplicações correlatas e para a elaboração do referencial teórico, além de ter sido utilizada na análise de dados internos do CECLIMAR, conforme comentado na conformidade comentado na [Introdução](#) deste trabalho. Segundo [Gil \(2002\)](#), a pesquisa bibliográfica é desenvolvida com base em materiais como livros e artigos científicos, ou seja, materiais já consolidados. Trata-se de uma pesquisa de grande importância, pois permite que os pesquisadores acessem diversos dados e informações dispersos que, individualmente, seriam trabalhos e custosos de se coletar. Nesse tipo de pesquisa, entretanto, é necessário ter cuidado com citações de terceiros, que podem interpretar de forma equivocada algum dado ou informação originalmente levantados.

Ainda segundo o autor, a pesquisa documental se diferencia pela natureza das fontes de informação. Enquanto as pesquisas bibliográficas consistem essencialmente em um apanhado de contribuições de diversos autores sobre determinado assunto, a pesquisa documental ocorre por meio de materiais que ainda não receberam tratamento analítico ou que podem ser reelaborados, a depender dos objetos de pesquisa. As fontes da pesquisa documental são mais diversas e podem incluir conversas pessoais, entrevistas, documentos ou sites. A metodologia de pesquisa bibliográfica foi realizada através da plataforma [Google Scholar \(2025\)](#), publicações presentes no portal do [SIBBr \(2024\)](#), Sistema de Informação sobre a Biodiversidade Brasileira, além de livros disponibilizados na biblioteca do Instituto Federal Campus Osório. Já a metodologia de pesquisa documental foi levantada a partir de arquivos internos, relatórios das Nações Unidas e conteúdos disponibilizados por desenvolvedores ou organizações que participaram do desenvolvimento das aplicações correlatas.

Após a fase de revisão bibliográfica se iniciou o desenvolvimento do sistema. Esta fase foi realizada a partir do levantamento de requisitos junto de profissionais do CECLIMAR e, os requisitos levantados foram cadastrados e refinados para desenvolvi-

mento cíclico do sistema. Cada ciclo visou a entrega de um produto com incrementos de requisitos pré estabelecidos. Ao final de cada um dos ciclos de desenvolvimento o sistema foi disponibilizado para testes com servidores do CECLIMAR, os *feedbacks* recebidos foram analisados, refinados e postos para desenvolvimento no ciclo seguinte. Um ponto de atenção no desenvolvimento desse sistema é que, por se tratar de um sistema de ciência cidadã, deve estar adequado com a LGPD para que possa ser publicado na [Google Play Store \(2024\)](#) para o uso da sociedade.

2.1 METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE

Para o desenvolvimento deste projeto foi escolhida uma abordagem de metodologia ágil relacionada ao ciclo iterativo incremental ([PRESSMAN, 2011](#)). Utilizando o *Kanban* como método de gestão de fluxo de trabalho ([OHNO, 1988](#)), a fim de melhorar a eficiência e qualidade do produto final a partir da visualização das tarefas. A ferramenta escolhida para realizar este gerenciamento foi o Jira ([Atlassian, 2025](#)).

Segundo [Pressman \(2011\)](#), os ciclos de desenvolvimento incremental podem ser divididos em 5 principais etapas: comunicação, planejamento, modelagem (análise e projeto), construção (codificação e testes) e emprego (entrega, *feedback*). As etapas por ele descritas foram aplicadas neste projeto. A comunicação foi marcada por reuniões agendadas com os profissionais do CECLIMAR para definição de escopo e levantamento de requisitos do sistema. O planejamento foi a análise, o refinamento e as definições de quais funcionalidades foram desenvolvidas em cada ciclo. Durante a modelagem foi realizada a prototipação e análise dos pontos levantados na etapa anterior. A construção foi o momento onde a codificação e os testes da aplicação foram o foco de atuação. E, por fim, durante o emprego o sistema foi disponibilizado para os profissionais do CECLIMAR e uma quantidade menor de usuários que contribuiram com testes e retornos de *feedback* para análise. Essas informações coletadas foram catalogadas e inseridas no *Backlog* para serem puxados em um ciclo posterior.

O uso desta metodologia tem o objetivo de realizar uma primeira entrega que possa ser considerada um Mínimo Produto Viável (*MVP*) atendendo aos requisitos básicos propostos inicialmente, mesmo que ainda se note a ausência de funcionalidades complementares. Esse *MVP* se tornará então a base de avaliação que permitirá identificar

necessidades adicionais e ajustes necessários. Com base nessa análise, é planejado o próximo incremento, ajustando a primeira entrega e adicionando novas funcionalidades conforme as necessidades.

Para o desenvolvimento deste projeto, foi utilizado um fluxo de trabalho no *Jira* para desenvolvimento de *software* visando auxiliar no processo e manter a visibilidade das tarefas de ponta a ponta. Para o *workflow* principal, foi montado um esquema com *Backlog*, Refinamento, Em desenvolvimento, Aguardando teste, Em teste, Correção de bugs e *Done* (Figura 1).

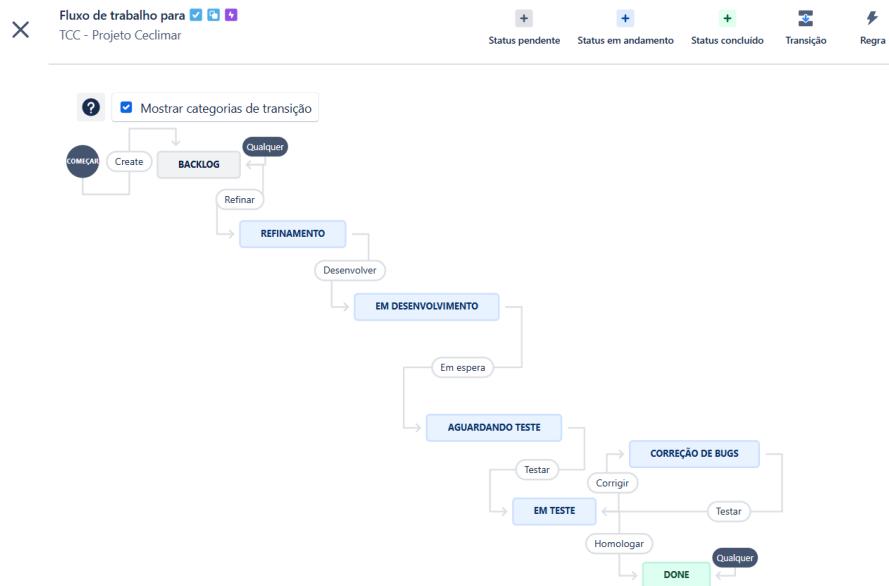


Figura 1 – Fluxo de trabalho do *Jira* gerado para o desenvolvimento do projeto.

Fonte: Autor

O *Backlog* é a coluna onde todas as tarefas, *user stories* e *bugs* serão inicialmente posicionados. Nesta etapa, as tarefas serão priorizadas antes de andarem para o próximo estágio.

No Refinamento, as tarefas puxadas do *Backlog* são detalhadas para um desenvolvimento mais assertivo. São refinados critérios de aceitação, estimativas de tempo e algum débito técnico.

As tarefas que estiverem em desenvolvimento são as que tiveram, efetivamente, o seu desenvolvimento iniciado. Assim que o desenvolvimento estiver finalizado, as tarefas

serão transferidas para aguardando teste, onde ficarão até serem puxadas para testes mais detalhados.

No estágio de teste, os critérios de aceite e a presença de *bugs* serão testados com o intuito de manter a qualidade do produto final. As tarefas que tiverem *bugs* ou divergências de regras de negócio identificadas serão movidas para a coluna Correção de *bugs* para que sejam corrigidas.

E, por último, após a homologação das tarefas nas etapas anteriores as tarefas são movidas para *Done* que indicará sua finalização.

2.2 EVOLUÇÃO TEMPORAL DO PROJETO

A Tabela 1 representa a trajetória de desenvolvimento do projeto ao longo de 16 meses, bem como as divergências e replanejamentos que foram realizados no decorrer do tempo. O trabalho teve seu início no dia 28 de março de 2024, com a realização de uma reunião de definição de escopo junto do orientador Marcelo Paravisi. No dia 1º de abril, deu-se continuidade com a definição do tema, em uma reunião externa com um membro do CECLIMAR. Essa etapa foi fundamental para delimitar o foco do trabalho e aprofundar a compreensão das dificuldades, necessidades, oportunidades e pontos críticos do projeto

Os meses seguintes, de maio a julho de 2024, foram dedicados à revisão bibliográfica e à definição metodológica. A definição metodológica se deu durante o mês de maio tendo início no dia 06, enquanto a parte de revisão bibliográfica se iniciou no dia 05 do mesmo mês e foi finalizada no final de julho.

O levantamento de requisitos e regras de negócio se deu a partir do dia 10 de maio e se estendeu até o final de agosto para que os alinhamentos de definições com a equipe do CECLIMAR fossem mais abertos e constantes. É importante ressaltar que durante o desenvolvimento houve mudanças de funcionalidades e ajustes de regras de negócio durante o período de testes que fizeram com que fosse necessário revisitar esse tópico.

Esse período inicial de estruturação se mostrou essencial para assegurar uma base sólida para iniciar o desenvolvimento. A parte de desenvolvimento técnica se iniciou com a organização da implementação do sistema no dia 20 de maio de 2024 e, com base nisso,

se iniciou a codificação do aplicativo no dia 1º de julho do mesmo ano.

Inicialmente, o desenvolvimento tinha um planejamento de conclusão para o final de outubro, porém, embora as atividades tenham progredido conforme o cronograma, em uma reunião com o orientador do projeto foi decidido realizar uma alteração no cronograma adicionando uma etapa de testes disponibilizando a aplicação com testadores tanto do projeto parceiro, como terceiros que contribuíram para que a aplicação adquirisse uma maturidade maior. Com isso, o prazo final de codificação foi replanejado para o início de maio de 2025 (Tabela 1).

A parte de redação da parte escrita do trabalho de conclusão foi iniciada no mês de junho para obter uma documentação mais precisa do processo e estava planejada para ser concluída até o final de novembro de 2024, porém, também foi afetada pelo replanejamento. Em vermelho na Tabela 1 podemos ver que esta etapa foi realizada dentro do prazo previsto até o mês de outubro, porém em novembro foi adiada para maio e junho para priorizar a conclusão da codificação e o aprimoramento da qualidade da aplicação. O desenvolvimento continuou sendo a atividade central até maio de 2025, acompanhado de execuções constantes de testes. No mês de maio de 2025 a redação do trabalho retornou e se estendeu, junto da revisão textual, até o final de junho para ser entregue dentro da data máxima de 26 de junho. A apresentação para a banca até o momento havia sido definida, porém tem prazo máximo de 11 de julho de 2025.

Esse cronograma evidencia o fluxo de trabalho realizado desde o início do planejamento do projeto, bem como as alterações ocorridas durante seu desenvolvimento. Para a elaboração desse material, foi fundamental que os períodos de escrita e codificação estivessem bem alinhados, permitindo traçar e documentar, com maior precisão, a linha do tempo apresentada, desde o início até a entrega final planejada.

Meses	Reunião def. de escopo	Definição de tema	Revisão bibliográfica	Def. metodológica	Levantamento de requisitos	Organização de implementação	Desenvolvimento	Redação de TCC	Revisão textual	Apresentação	Testes
Mar/24											
Abr/24											
Mai/24											
Jun/24											
Jul/24											
Ago/24											
Set/24											
Out/24											
Nov/24											
Dez/24											
Jan/25											
Fev/25											
Mar/25											
Abr/25											
Mai/25											
Jun/25											

Tabela 1 – Cronograma de desenvolvimento — Cinza: Planejamento inicial, Azul: Replanejamento, Vermelho: Prazos adiados.

Fonte: Autor

3 ESTUDO DE APLICAÇÕES CORRELATAS

Com o intuito de contextualizar e ter uma melhor visão de onde posicionar o projeto dentro do campo de pesquisa escolhido, foi realizada uma seleção de trabalhos correlatos, cujos principais critérios de aceitação incluíram a abordagem de temas como: monitoramento de fauna, desenvolvimento sustentável, gestão ambiental e ciência cidadã. Garantindo assim, relevância e alinhamento do projeto com as necessidades e avanços da área.

O ponto de partida para as pesquisas foi o SIBBr (2024). A plataforma online, pertencente ao sistema gov.br (Governo Federal do Brasil, 2024), que integra dados e informações sobre a biodiversidade e os ecossistemas de diversas fontes, os tornando acessíveis e livres para usos diversos. Nele, também é possível ter acesso ao sistema Ciência Cidadã, que consiste em uma colaboração entre a comunidade e cientistas na coleta de dados para pesquisa científica.

A partir do levantamento bibliográfico, pesquisas na plataforma Play Store (Google Play Store, 2024), presente nos dispositivos Android (Google, 2024a), foram realizadas baseadas em alguns projetos e palavras-chave pré-selecionadas. As principais palavras-chave usadas para realizar as buscas foram: animais costeiros, coleta de dados, monitoramento ambiental, ciência cidadã.

Nesta seção, serão apresentadas três aplicações separadas no levantamento documental que demonstraram possuir características e funcionalidades que podem auxiliar na definição e desenvolvimento deste sistema. Estas aplicações possuem algumas características em comum, porém cada uma delas também traz características únicas que serão importantes balizadoras nas tomadas de decisão deste projeto.

3.1 SISS-GEO

O Sistema de Informação em Saúde Silvestre (SISS-Geo) da FIOCRUZ é desenvolvido pela Plataforma Institucional Biodiversidade e Saúde Silvestre, com apoio do Laboratório Nacional de Computação Científica. É gratuito, disponível em *smartphones* e na *web*, para o monitoramento da saúde dos animais silvestres em ambientes naturais,

rurais e urbanos. Apoia a investigação da ocorrência de agentes causadores de doenças, como agentes infecciosos, que podem acometer pessoas e animais. Como instrumento de ciência cidadã torna possível, a partir de registros realizados por cidadãos comuns, profissionais de saúde, meio ambiente, pesquisadores e especialistas em vida silvestre, agir para a prevenção e controle de zoonoses e a conservação da biodiversidade brasileira (CHAME et al., 2015).

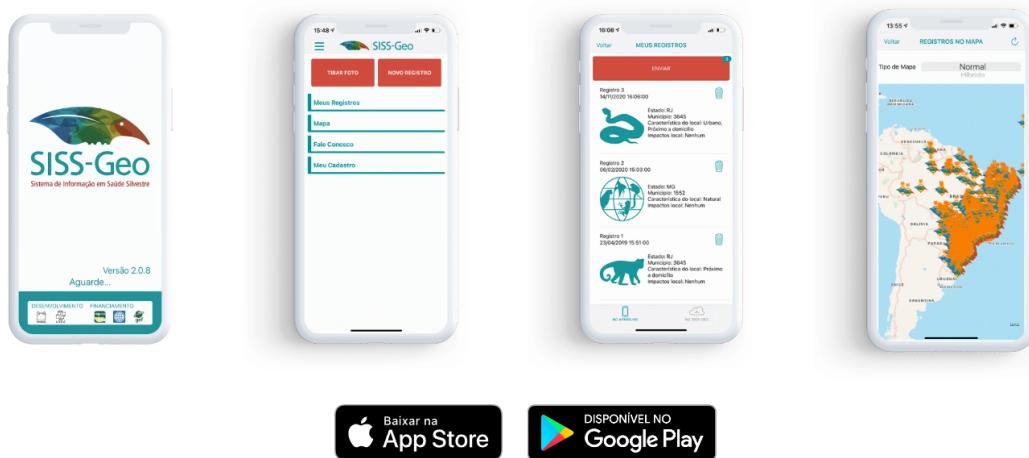


Figura 2 – Interface do sistema SISS-Geo.

Fonte: [SISS-Geo \(2024\)](#).

A aplicação *mobile* foi lançada em 2014 e está disponível tanto para iOS quanto para Android (Figura 2) e possui mais de 10.000 downloads, com avaliação média de 4,7/5 baseada em 116 avaliações de usuários. Foram registrados, até 26 de março de 2024, 33 mil registros e quase 13 mil usuários colaboradores.

O sistema é bem consolidado e amplamente utilizado no Brasil, possuindo uma ótima aceitação entre especialistas e cidadãos em diversas regiões do país. Apesar de possuir opções para identificação de dados da fauna costeira, estes se mostram limitados quando comparados com os demais biomas. Ao alterar a escala de análise, é possível perceber, com um recorte mais detalhado do cordão litorâneo, que este não é o principal foco da aplicação e, atualmente, ela possui uma participação muito maior nas regiões continentais (Figura 3).

Principais pontos fortes: funcionalidade *offline*, iniciativa ambiental de grande contribuição, aplicação leve, ideia colaborativa de preservação, georreferenciamento de

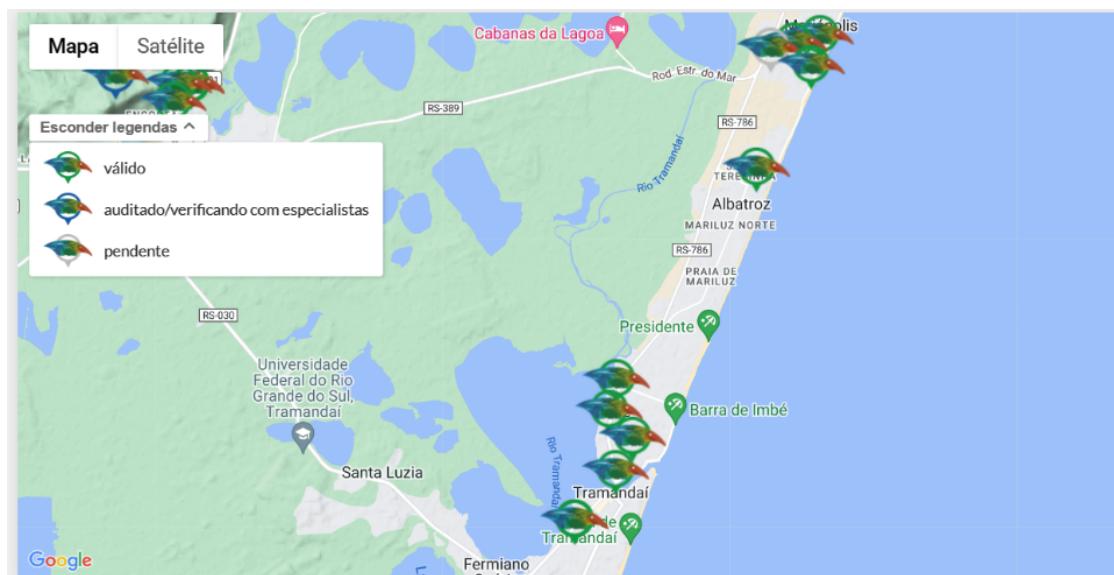


Figura 3 – Recorte da região costeira do litoral norte do Rio Grande do Sul.

Fonte: [SISS-Geo \(2024\)](#).

dados com boa visualização.

Entre as principais reclamações dos usuários estão: sistema pouco intuitivo, poucas opções de animais pré-cadastrados, interface confusa com formulário por vezes muito técnico.

3.2 SISTEMA URUBU

O Sistema Urubu é uma iniciativa do Centro Brasileiro de Ecologia de Estradas da UFLA, sob a coordenação do professor Alex Bager. Criado em 2014, o aplicativo de ciência cidadã é a maior rede para conservação da biodiversidade brasileira, destinada à coleta e gestão de informações de fauna selvagem ao longo de rodovias e ferrovias no Brasil.

A aplicação permite que voluntários enviem registros de animais atropelados e infraestruturas viárias por meio de aplicativo móvel, enquanto especialistas validam e caracterizam esses registros para torná-los confiáveis. Os dados coletados são centralizados em um banco de dados e disponibilizados em um Sistema de Informações Geográficas, facilitando a visualização e análise pelos usuários. O Sistema Urubu também oferece

ferramentas como o Urubu *web*, para gestão e validação dos dados, e o Urubu Map, para visualização geográfica dos registros (CASTRO; BAGER, 2019). Ao longo de seus anos de existência, o sistema reuniu mais de 25 mil usuários e 150 mil registros de animais atropelados em todo o território brasileiro, demonstrando seu impacto e relevância na conservação da biodiversidade.

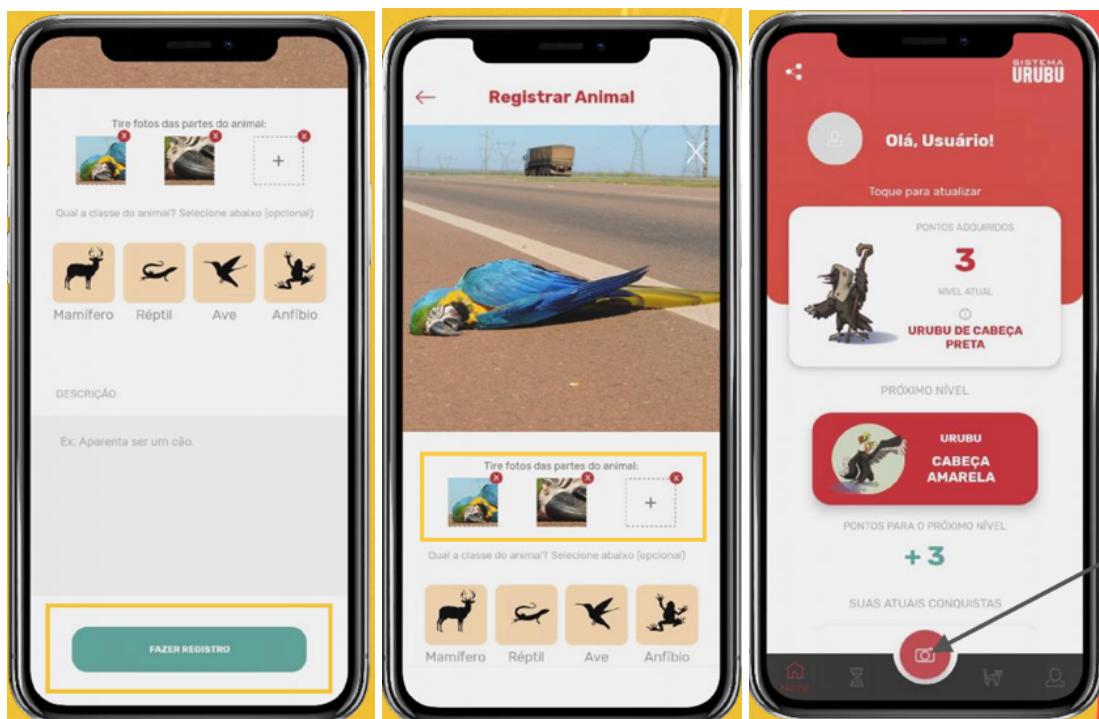


Figura 4 – Interface do Sistema Urubu.

Fonte: Capturas de tela do sistema urubu disponíveis no manual de uso do aplicativo.

O Sistema Urubu apresenta uma interface amigável e um fluxo de obtenção e envio de dados de registros intuitivo (Figura 4.a e 4.b), além de um fluxo gamificado com progressão recompensada para os usuários que contribuem com o projeto (Figura 4.c).

O fluxo de funcionamento da aplicação se inicia com a morte de algum animal em uma rodovia ou ferrovia, quando esse animal é encontrado por usuário a entrada de registro pode ser realizada via *mobile*, *web* ou por importação via planilha de Excel. Os dados de cada registro são então passados por uma análise de profissionais que decidirão se serão inseridos nos dados finais.

Apesar de ter alcançado importantes números de aceitação e uso entre 2014 e

2023, atualmente o sistema se encontra desativado e indisponível para download em todas as plataformas e seu site se encontra fora do ar.

Em contato com o coordenador do projeto Alex Bager por e-mail, foi informado de que o Sistema Urubu foi desativado devido a falta de recursos. Segundo seu relato, além dos custos para manter a aplicação no ar com investimentos contínuos, os aplicativos de ciência cidadã requerem muita comunicação e relacionamento com os participantes, o que torna a estrutura mais complexa e custosa.

3.3 SIMBA

O Sistema de Monitoramento da Biota Aquática (SIMBA), é um sistema *web* de gerenciamento de dados criado pelo Laboratório de Oceanografia Biológica da UNIVALI com a finalidade de armazenar dados coletados por instituições executoras dos projetos de monitoramentos de praias. O desenvolvimento do SIMBA se iniciou para auxiliar nos fluxos de dados entre os atores sociais envolvidos nos Projetos de Monitoramento de Praias (PMPs) e fazer com que os dados obtidos sejam disponibilizados e divulgados para a população.

Os PMPs são desenvolvidos para o atendimento de condicionantes de licenciamento ambiental federal, conduzido pelo IBAMA, de atividades de exploração e produção de petróleo e gás natural de bacias *offshore* sob atuação da Petrobras. Estes projetos tem o objetivo de avaliar as possíveis interferências na área de abrangência dos projetos, analisando tanto tetrápodes marinhos (aves, tartarugas e mamíferos) por meio do monitoramento das praias, atendimento veterinário aos animais debilitados e da coleta de dados de animais mortos, quanto resíduos sólidos encontrados ([Petrobras, 2024](#)).

Atualmente os projetos estão presentes nas bacias de Santos, Campos, Espírito Santo, Sergipe-Alagoas e Potiguar.

O SIMBA conta com funcionalidades de cadastro de ocorrências de fauna, resíduos sólidos (Figura 5), exames, jornadas de campo com os caminhamentos realizados. Os pontos de ocorrência de cada cadastro são georreferenciados, assim como as jornadas de monitoramento, gerando mapas para melhor visualização e análise dos dados (Figura 6).

O sistema possui uma área de acesso liberada ao público onde é possível visualizar dados já levantados e validados e uma área de acesso restrito liberada para pesquisadores.

The screenshot shows a web-based monitoring system interface. At the top, there's a header with the SIMBA logo and navigation links like 'Visão Geral', 'PMP-BS Área SP', 'Ocorrências de resíduos sólidos', and 'Visualizar'. A message indicates an occurrence was validated by 'Carla Beatriz' on '20/11/2023 10:24:30'. The main content area displays detailed information about a specific occurrence:

- Identificador da ocorrência:** T1020231029s328002
- Responsável:** Tiago Ucella
- Equipe:** Luis Felipe Rêgo
- Trecho:** São Francisco + Olaria
- Tipo do monitoramento:** Acionamento (60032)
- Data/Hora:** 29/10/2023 10:42:00
- Ponto:** Lat -23.7633627, Long -45.4067148
- Fotos:** Six small thumbnail images showing various environmental samples.
- Observações:** A note stating 'Tonel de óleo encontrado na faixa de areia pelo acionante. Devido as condições de oxidação do tonel, não foi possível identificar a empresa de origem. Tonel foi levado para a equipe para local de fácil acesso, para ser retirado pela empresa responsável pela limpeza de praias.'

On the left side, there's a sidebar with a tree-like menu structure:

- Ocorrências de fauna não avifauna
- Ocorrências de resíduos sólidos
- Ocorrências de resíduos oleosos
- Ocorrências de reprodução de quelônios
- Exames laboratoriais/imagens
- Exames anatomo-patológicos
- Biometrias
- WebGIS
- Relatórios parametrizáveis
- PMP-BS Área RJ
- PMP-BS Área SC/PR
- PMP-BS Fase 1
- PMP-BS Fase 2
- PMP-BC/ES

Figura 5 – Captura de tela de uma página *web* de registro de visualização de ocorrências cadastradas no SIMBA.

Fonte: ([Petrobras, 2024](#)).

Possui uma interface mais limpa e direta, atendendo a um estilo de aplicação profissional apenas com o conteúdo necessário (Figura 7). Os dados são bem organizados e acessíveis para quem estiver interessado em acompanhar os estudos e evidências coletadas. Porém, para leigos essa interface pode, inicialmente, trazer um pouco de estranheza devido ao visual menos apelativo e aos termos mais científicos apresentados.

A geração de mapas georreferenciados e a possibilidade de uso *offline* são aspectos importantes a serem destacados, pois permitem uma melhor análise e acompanhamento gráfico dos dados coletados e possibilitam que estes dados sejam obtidos até mesmo em locais mais isolados sem conexão com a internet. Outra função interessante é a liberação de dados para o público geral que se mostrar interessado em acompanhar os resultados e andamentos dos estudos nas praias onde os PMPs estão sendo realizados.

3.4 ANÁLISE COMPARATIVA E DIRECIONAMENTOS DO PROJETO

Com a análise dos principais trabalhos correlatos existentes, foi possível levantar os principais pontos de sucesso e alguns aspectos que precisam de melhorias, os quais influenciam diretamente na capacidade de sistemas similares alcançarem seus objetivos e obterem a aceitação do público-alvo.

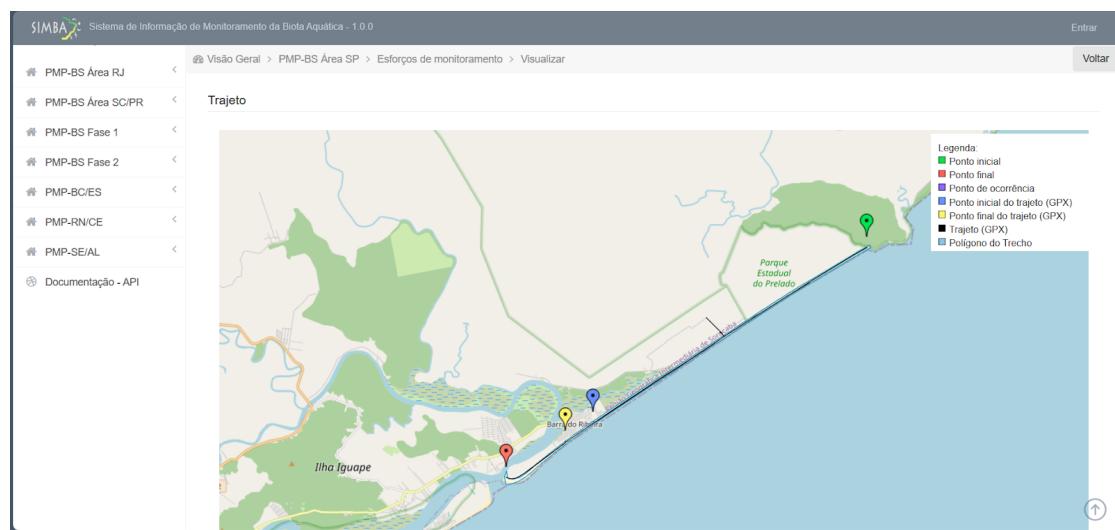


Figura 6 – Captura de tela de um mapa georreferenciado gerado a partir dos dados de jornada de monitoramento cadastrado no SIMBA.

Fonte: ([Petrobras, 2024](#)).

A receptividade do Sistema Urubu e do SISS-Geo indicou um interesse significativo da população em participar de projetos de ciência cidadã, evidenciado pelos números expressivos de *downloads* e registros de dados apresentados anteriormente.

Cada um dos três sistemas possui suas peculiaridades e objetivos distintos: enquanto o SISS-Geo está mais voltado ao monitoramento da saúde de animais silvestres e à investigação de agentes causadores de doenças, o Sistema Urubu foca principalmente no monitoramento de animais atropelados em rodovias e ferrovias. Por sua vez, o SIMBA busca fornecer visibilidade ao monitoramento costeiro realizado em regiões de exploração e produção de petróleo.

Sendo assim, a partir da análise prévia, foi possível constatar que existe uma oportunidade de posicionamento para o sistema proposto neste projeto, que se enquadra como uma plataforma com interface amigável e intuitiva, visando otimizar o processo de coleta, classificação e gestão de dados da fauna costeira. O sistema poderá se apoiar em pontos fortes identificados nos sistemas correlatos, como a funcionalidade offline do SISS-Geo, a interface amigável do Sistema Urubu e a organização dos dados do SIMBA.

O sistema proposto buscará oferecer uma experiência de usuário agradável e descomplicada para o registro de observações da fauna costeira, com opções abrangentes

The screenshot shows a web-based monitoring system interface. At the top, there's a header with the SIMBA logo and the text "Sistema de Informação de Monitoramento da Biota Aquática - 1.0.0". Below the header, a navigation menu on the left includes links like "Visão Geral", "PMP-BS Área SP" (which is highlighted in green), and various monitoring categories such as "Ocorrências de fauna avô individual", "Ocorrências de fauna avô coletiva", etc. The main content area is titled "Esforço de monitoramento" and has a "Visualizar" button. It shows a breadcrumb trail: "Visão Geral > PMP-BS Área SP > Esforços de monitoramento > Visualizar". A message indicates a registration was made by "Pedro Sardinha do Prado" on "13/12/2023 12:11:00" and last modified by "Shany Nagaoka" on "19/12/2023 15:53:38". Below this, there are two tables: one for "Informações do monitoramento" and another for "Condições do ambiente durante o monitoramento". The "Informações do monitoramento" table has columns for Código, Tipo, Tipo de veículo, Trecho, Equipe, Responsável, Data/Hora inicio, Data/Hora término, Ponto inicial, Ponto final, Completo, and Arquivo gpx. The "Condições do ambiente durante o monitoramento" table has columns for Condição do céu, Condição do mar, Maré, Vento, Direção do vento, and Observações climáticas. Both tables contain specific data entries.

Figura 7 – Captura de tela da página web de monitoramento cadastrada no SIMBA.

Fonte: (Petrobras, 2024).

de espécies para classificação e ferramentas de visualização de dados claras e acessíveis. Ao adotar uma abordagem centrada no usuário e priorizar a simplicidade e eficiência, pretende-se ampliar o engajamento da comunidade na coleta de dados científicos e contribuir significativamente para o monitoramento e conservação da biodiversidade costeira.

4 REFERENCIAL TEÓRICO

4.1 CIÊNCIA CIDADÃ

A Ciência Cidadã representa uma ponte entre a comunidade científica e o público geral, permitindo que pessoas sem formação científica formal possam contribuir em pesquisas científicas. Essa metodologia colaborativa tem se destacado em diversas áreas de pesquisa como na conservação da biodiversidade e na preservação ambiental. Através da Ciência Cidadã é possível que voluntários coletem e analisem dados, fornecendo informações que agregam no conhecimento acadêmico e auxiliam na resolução de questões sociais.

Segundo [Palma \(2016\)](#), trata-se de uma metodologia de pesquisa promissora na produção de conhecimento para ser aplicada em diversos campos científicos. Essa abordagem se destaca, em especial, com seu potencial de geração de dados e análises, temporal e espacial, quando comparada com os métodos tradicionais de pesquisa. Para [Wildschut \(2017\)](#), a metodologia da ciência cidadã tem potencial para ampliar o escopo de pesquisas e aumentar e aprimorar a capacidade na coleta de dados e que os cidadãos que participam podem contribuir com informações importantes enquanto aprendem sobre as mais diversas áreas científicas.

A construção de conhecimento colaborativo realizado entre cidadãos e cientistas se mostra uma maneira poderosa de construção de conhecimentos, que agrupa tanto no meio científico quanto social. Estes projetos instigam que as pessoas participem de maneira voluntária e ativa na resolução de situações do dia-a-dia da nossa sociedade, disseminando conhecimento de diversas áreas e fazendo com que diversos conteúdos saiam de suas bolhas científicas e obtenham um alcance maior.

4.2 AGENDA 2030

A Agenda 2030 para o Desenvolvimento Sustentável é um plano de ação global adotado pelas Nações Unidas em 2015, com o objetivo de promover a prosperidade enquanto protege o planeta. Ela estabelece 17 Objetivos de Desenvolvimento Sustentável

(ODS), que são subdivididos em 169 metas específicas. Esses objetivos englobam uma ampla gama de questões sociais, econômicas e ambientais, como erradicação da pobreza, igualdade de gênero, educação de qualidade, água limpa e saneamento, energia acessível e não poluente, trabalho decente e crescimento econômico ([NAÇÕES UNIDAS, 2015b](#)).

Segundo a organização, os ODS são projetados com os três pilares do desenvolvimento sustentável: econômico, social e ambiental. A Agenda enfatiza a importância de garantir que os direitos humanos de todos sejam realizados e que haja igualdade de gênero e empoderamento de mulheres e meninas. Além disso, reconhece que a erradicação da pobreza em todas as suas formas é o maior desafio global e, um requisito indispensável para o desenvolvimento sustentável.

A implementação da Agenda 2030 requer a mobilização de recursos e uma Parceria Global para o Desenvolvimento Sustentável, envolvendo todos os países, partes interessadas e pessoas. A Agenda é mundial, e se aplica a todos os países levando em conta diferentes realidades nacionais, capacidades e níveis de desenvolvimento. Ela promove a paz, justiça e instituições eficazes, e destaca a necessidade de ações urgentes sobre a mudança climática para proteger o planeta para as gerações presentes e futuras ([NAÇÕES UNIDAS, 2015a](#)).

4.3 OBJETIVOS DE DESENVOLVIMENTO SUSTENTÁVEL (ODS)

Os ODS são um conjunto de 17 metas estabelecidas pelas Nações Unidas para abordar os principais desafios de desenvolvimento no Brasil e em todo o mundo. Esses objetivos visam criar um futuro mais justo, equitativo e sustentável para todos ([NAÇÕES UNIDAS, 2015b](#)). São eles:

- **ODS 1:** Erradicação da Pobreza: Acabar com a pobreza em todas as suas formas, em todos os lugares.
- **ODS 2:** Fome Zero e Agricultura Sustentável: Garantir a segurança alimentar, melhorar a nutrição e promover a agricultura sustentável.
- **ODS 3:** Saúde e Bem-Estar: Assegurar uma vida saudável e promover o bem-estar para todas as idades.

- **ODS 4:** Educação de Qualidade: Garantir uma educação inclusiva, equitativa e de qualidade.
- **ODS 5:** Igualdade de Gênero: Alcançar a igualdade de gênero e empoderar todas as mulheres e meninas.
- **ODS 6:** Água Limpa e Saneamento: Garantir a disponibilidade e gestão sustentável da água e saneamento para todos.
- **ODS 7:** Energia Limpa e Acessível: Assegurar o acesso a fontes de energia acessíveis, confiáveis, sustentáveis e modernas.
- **ODS 8:** Trabalho Decente e Crescimento Econômico: Promover o crescimento econômico inclusivo e sustentável, emprego pleno e produtivo e trabalho decente para todos.
- **ODS 9:** Indústria, Inovação e Infraestrutura: Construir infraestruturas resilientes, promover a industrialização inclusiva e sustentável e fomentar a inovação.
- **ODS 10:** Redução das Desigualdades: Reduzir as desigualdades dentro e entre países.
- **ODS 11:** Cidades e Comunidades Sustentáveis: Tornar as cidades e os assentamentos humanos inclusivos, seguros, resilientes e sustentáveis.
- **ODS 12:** Consumo e Produção Responsáveis: Assegurar padrões de consumo e produção sustentáveis.
- **ODS 13:** Ação Contra a Mudança Global do Clima: Tomar medidas urgentes para combater a mudança climática e seus impactos.
- **ODS 14:** Vida na Água: Conservar e usar de forma sustentável os oceanos, mares e recursos marinhos.
- **ODS 15:** Vida Terrestre: Proteger, restaurar e promover o uso sustentável dos ecossistemas terrestres, gerir florestas de forma sustentável, combater a desertificação e deter a perda de biodiversidade.

- **ODS 16:** Paz, Justiça e Instituições Eficazes: Promover sociedades pacíficas e inclusivas para o desenvolvimento sustentável, proporcionar acesso à justiça para todos e construir instituições eficazes, responsáveis e inclusivas em todos os níveis.
- **ODS 17:** Parcerias e Meios de Implementação: Fortalecer os meios de implementação e revitalizar a parceria global para o desenvolvimento sustentável.

Os itens acima são um apelo global visando acabar com a pobreza, proteger o meio ambiente e o clima e garantir que as pessoas, em todos os lugares, possam desfrutar de paz e de prosperidade. São objetivos nos quais as Nações Unidas estão contribuindo para atingir a Agenda 2030 no Brasil ([NAÇÕES UNIDAS, 2015b](#)).

4.4 METODOLOGIA ITERATIVA INCREMENTAL

A metodologia iterativa incremental é uma abordagem de desenvolvimento de *software* que divide o processo em ciclos repetitivos. Cada iteração resulta em uma versão incrementada do *software*, que é construída sobre a versão anterior com adições e melhorias. Segundo [Pressman \(2011\)](#), essa metodologia permite que as equipes avaliem e integrem feedbacks mais rapidamente, adaptando-se às mudanças e refinando o produto ao longo do tempo. Isso contrasta com o modelo tradicional em cascata, onde cada fase deve ser concluída antes da próxima começar, sem retorno para fases anteriores.

De acordo com [Pressman \(2011\)](#), o modelo incremental é uma abordagem de desenvolvimento de *software* que foca na entrega gradual de funcionalidades em sucessivas versões incrementais. Nesse modelo, o desenvolvimento do *software* é dividido em várias partes ou incrementos, cada um entregando uma versão operante do produto, que é aprimorada e expandida em lançamentos subsequentes.

[Pressman \(2011\)](#) afirma que o modelo incremental integra elementos de fluxos de processos lineares e paralelos. Na Figura 8 é possível observar essa relação a partir das sequências lineares de forma escalonada que demonstram que ao longo do tempo cada incremento adiciona funcionalidades ou aprimoramentos ao sistema, permitindo uma evolução constante e contínua do produto final.

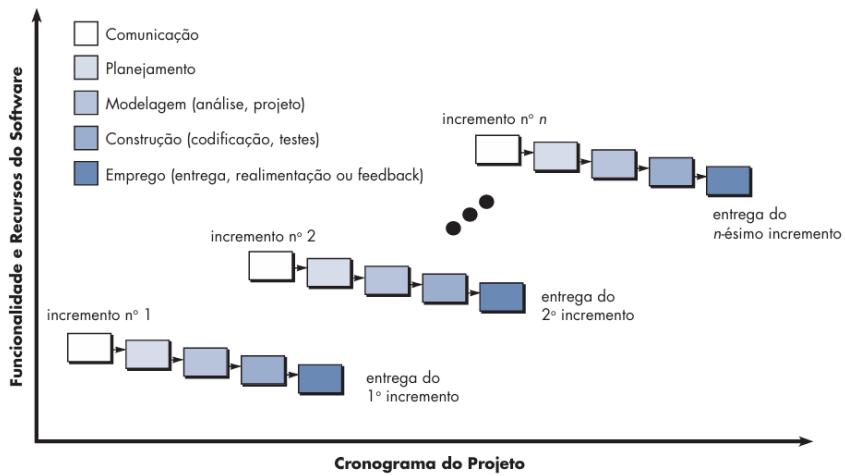


Figura 8 – Ciclo de desenvolvimento incremental.

Fonte: Pressman (2011).

4.5 KANBAN

O Kanban é um método de gestão de fluxo de trabalho que fornece uma visualização das tarefas que devem ser realizadas, quando entregá-las e quanto ainda é necessário para finalizá-las com o objetivo de aumentar a eficiência e o entendimento do desenvolvimento (OHNO, 1988). Originou-se no sistema da Toyota no Japão dos anos 1940 e foi adaptado para o desenvolvimento de *software* e sistemas de TI por David J. Anderson em 2004. Para isso, o Kanban utiliza de um quadro dividido em colunas para representar os diferentes estágios do trabalho, onde os cartões que representam as tarefas se movem de uma coluna para outra, refletindo o progresso real (ZAYAT; SENVAR, 2020).

4.6 JIRA

O *Jira* é um *software* comercial de gerenciamento de projetos desenvolvido pela empresa Atlassian (2025). Ele permite a criação, acompanhamento e gerência de tarefas e projetos a partir de uma interface personalizável que suporta diversas metodologias ágeis como Scrum e Kanban, facilitando a colaboração e comunicação durante o desenvolvimento.

A ferramenta pode ser usada para planejar *sprints*, atribuir tarefas, acompanhar

bugs, gerar relatórios e analisar o desempenho da equipe. Ele possui integração com uma variedade de ferramentas de desenvolvimento e oferece funcionalidades para personalizar fluxos de trabalho, campos e painéis, tornando-o adaptável às necessidades específicas de cada projeto ou organização.

4.7 TECNOLOGIAS E FRAMEWORKS

Neste capítulo, serão apresentadas as tecnologias escolhidas para o desenvolvimento do aplicativo, destacando os aspectos que influenciaram nas escolhas e como elas contribuíram para o desenvolvimento do projeto.

4.7.1 Flutter

O *Flutter* é um conjunto de ferramentas de UI (Interface do Usuário) criado pelo *Google* em 2017. Se diferencia por possuir um suporte multiplataforma de código aberto projetado para permitir a reutilização de código em diversos sistemas operacionais, como *iOS*, *Android*, *web* e *desktop*, enquanto possibilita que as aplicações interajam diretamente com os serviços de cada plataforma. Como publicado por seus desenvolvedores, o *Flutter* oferece uma solução centralizada para o desenvolvimento multiplataforma que facilita a criação de componentes visuais ao mesmo tempo em que realiza a integração com recursos nativos de cada tipo de dispositivo ([Flutter, 2025](#)).

Com esse *framework*, os aplicativos são compilados a partir de um único código-base para as plataformas *mobile*, *web* e *desktop*. O desenvolvimento no *Flutter* é baseado em *Widgets*, componentes escritos em *Dart* que são responsáveis pela visualização das telas e pela interação com o usuário. Esses componentes podem ou não estar associados a um estado, quando estão, são chamados de *Stateful Widgets* e, quando não, de *Stateless Widgets* ([Flutter, 2025](#)).

Além disso, o *Flutter* possui a funcionalidade de *hot reload*. Uma ferramenta que permite que alterações sejam feitas no código enquanto a aplicação está em execução, gerando uma atualização em tempo real da interface do usuário e da lógica da aplicação, mantendo inalterado seu estado de execução.

A escolha pelo *Flutter* (versão 3.29.2) se deu principalmente pela sua capacidade multiplataforma a partir de uma única base de código, otimizando tempo e recursos.

Outro fator decisivo foi a facilidade de integração com os diversos serviços do *Firebase*, que facilitam a implementação de funcionalidades essenciais, como a autenticação, o banco de dados em tempo real e o armazenamento.

4.7.2 Dart

Dart é uma linguagem de programação lançada pelo *Google* em 2013, criada com foco em produtividade e performance para aplicações multiplataforma. A linguagem é orientada a objetos e oferece tipagem estática forte, porém também é possível optar pelo uso de tipagem dinâmica, o que pode ser útil em algumas situações. ([Dart, 2025](#)).

Um aspecto importante do *Dart* é sua flexibilidade de compilação. É uma linguagem que utiliza tanto a compilação *just-in-time* (JIT) quanto *ahead-of-time* (AOT). Em tempo de desenvolvimento, o *Dart* roda no *Dart VM* usando JIT, o que permite recompilar apenas partes do código em tempo real sem reiniciar toda a aplicação. Essa funcionalidade viabiliza o recurso de *hot reload* do *Flutter*, permitindo que alterações sejam refletidas imediatamente na aplicação sem alterar o estado da interface. Após realizar o lançamento de uma versão, o *Dart* usa AOT para compilar o código diretamente para código nativo, o que garante um desempenho maior com relação ao JIT ([Dart, 2025](#)).

A escolha do Dart (versão 3.7.2) como linguagem de programação para o projeto se deu pela sua proximidade com o *Flutter*, pela tipagem forte, pelas características de programação orientada a objetos e pela facilidade de integração com o *Firebase*.

4.7.3 Firebase

O *Firebase* é uma plataforma de desenvolvimento de aplicativos móveis e *web* fornecida pelo *Google*, que oferece uma ampla gama de serviços *backend* para facilitar a criação de aplicações robustas e escaláveis. Tem como característica permitir o desenvolvimento centralizado sem a necessidade de infraestrutura própria, fornecendo soluções completas para autenticação de usuários, banco de dados em tempo real, armazenamento de arquivos, envio de notificações, análises de uso e monitoramento de desempenho. Além disso, a plataforma integra ferramentas para teste, distribuição e acompanhamento do ciclo de vida do aplicativo com suporte multiplataforma ([Google Firebase, 2025](#)).

Dentre os principais serviços disponibilizados no plano *Spark* que foram utilizados

no projeto, estão o *Firebase Authentication*, que possibilita a autenticação de usuários via e-mail/senha e *Google*, com suporte para até 50 mil usuários ativos mensais sem custo. O *Firebase Analytics* que fornece relatórios detalhados sobre o comportamento e uso do aplicativo, permitindo a análise de métricas essenciais. O *App Distribution* que possibilita a distribuição de versões de teste do aplicativo para usuários selecionados durante as fases de desenvolvimento e teste. Além do *Cloud Firestore Database* que oferece até 1 GB de armazenamento, com limites diários de 50 mil leituras, 20 mil gravações e 20 mil exclusões de documentos. E do *Cloud Storage* que disponibiliza 5 GB de armazenamento, 1 GB de download diário, além de 20 mil operações de upload e 50 mil operações de download por dia ([Google Firebase, 2025](#)).

Neste projeto, o *Firebase* foi escolhido para gerenciar a autenticação e o armazenamento de dados, utilizando o *Firebase Authentication* para autenticar usuários com diferentes métodos de login e o *Firebase Storage* para armazenar imagens, além do banco de dados NoSQL *Cloud Firestore Database* para salvar os dados gerais do sistema e do *Analytics* para geração de *dashboards* visando analisar o funcionamento geral da aplicação. A definição pelo *Firebase* se deu pela robustez e facilidade de integração com *Flutter* oferecendo uma plataforma de backend eficiente e escalável. A aplicação foi construída com a utilização dos benefícios do plano *Spark* descritos acima, que, apesar de ser um plano gratuito, traz diversas ferramentas com limites de uso suficientemente altos para projetos iniciais.

4.7.4 Pacotes e Plugins do Flutter

Como o *Flutter* possui uma arquitetura baseada em pacotes e *plugins*, para o desenvolvimento de algumas funcionalidades específicas da aplicação proposta neste trabalho foram selecionados algumas dependências externas para facilitar e aumentar as possibilidades de desenvolvimento.

As dependências do projeto podem ser classificadas em três categorias. Bibliotecas puramente em Dart, que não interagem com *APIs* nativas da plataforma, como *excel* (para manipulação de planilhas), *intl* (internacionalização e formatação de dados) e *diacritic* (remoção de acentos e caracteres especiais).

Plugins, que integram o código *Dart* com *APIs* nativas dos sistemas operacionais

Android e *iOS*. Exemplos incluem `geolocator` (geolocalização), `image_picker` (acesso à câmera e galeria) e `permission_handler` (gerenciamento de permissões).

E as ferramentas auxiliares, que contribuem com a geração de código e automação de processos durante o desenvolvimento, como `hive_generator` (geração de adaptadores para o *Hive*) e `build_runner` (execução de processos automáticos de geração de código).

Essas dependências permitem incorporar funcionalidades reutilizáveis no projeto como renderização de mapas, autenticação com Firebase e diversos formatos de gráficos sem a necessidade de implementações manuais que seriam custosas e extensas. Além disso, possibilitam o acesso a recursos nativos do dispositivo, como a geolocalização, gerenciamento de permissões, câmera e o armazenamento local no dispositivo.

A Tabela 2 apresenta todas as dependências utilizadas no projeto, especificando a versão de cada uma, bem como uma breve descrição de sua finalidade. Essa relação visa demonstrar a amplitude de recursos externos incorporados ao aplicativo e documentar as versões exatas para fins de controle de compatibilidade.

Tabela 2 – Dependências do Projeto *Flutter*

Dependência	Versão	Descrição breve
flutter	sdk: flutter	Framework principal para desenvolvimento de apps móveis.
flutter_localizations	sdk: flutter	Supporte a localização (i18n) para Flutter.
cupertino_icons	^1.0.2	Ícones estilo iOS para Flutter.
firebase_core	^3.8.1	Inicialização e integração do Firebase.
firebase_auth	^5.3.4	Autenticação com Firebase.
cloud_firestore	^5.4.4	Integração com o banco Firestore.
firebase_analytics	^11.3.3	Monitoramento e análise de uso (analytics).
firebase_app_check	~0.3.2+5	Segurança para apps usando Firebase.
google_sign_in	^6.2.2	Autenticação via Google.
persistent_bottom_nav_bar	^6.2.1	Barra de navegação inferior persistente.
image_picker	^1.1.2	Seleção de imagens da galeria ou câmera.
firebase_storage	^12.3.4	Armazenamento de arquivos no Firebase.
diacritic	~0.1.3	Remoção de acentos (diacríticos) de strings.
provider	^6.1.2	Gerenciamento de estado simplificado.
photo_view	~0.15.0	Visualização de imagens com zoom e rotação.
geolocator	^13.0.2	Geolocalização e rastreamento de posição.
geocoding	^3.0.0	Geocodificação e reversa.
phosphor_flutter	^2.1.0	Ícones customizados da biblioteca Phosphor.
url_launcher	^6.3.1	Abertura de URLs externas (navegadores, apps, etc.).
accordion	^2.6.0	Componente de interface tipo acordeão.
cached_network_image	^3.4.1	Cache para imagens carregadas da web.
hive	^2.2.3	Banco de dados local rápido e leve.
hive_generator	^2.0.1	Gerador de código para Hive.
path_provider	^2.1.1	Acesso a diretórios do sistema de arquivos.
connectivity_plus	^5.0.2	Detecção de status de conectividade (online/offline).
build_runner	^2.4.8	Utilitário para geração de código automático.
skeletonizer	^1.4.2	Animação skeleton loader (placeholders enquanto carrega).
badges	^3.1.2	Criação de badges/insígnias visuais.
pie_chart	^5.4.0	Gráficos de pizza.
graphic	^2.5.0	Biblioteca de gráficos avançados.
share_plus	^10.1.4	Compartilhamento nativo de conteúdo.
exif	^3.0.1	Leitura de dados EXIF de imagens.
latlong2	^0.9.0	Tipos de dados para latitude e longitude.
open_filex	^4.3.4	Abertura de arquivos em apps externos.
excel	^4.0.6	Manipulação de arquivos Excel (.xlsx).
flutter_map	^8.1.1	Renderização de mapas (como o OpenStreetMap).
intl	~0.19.0	Internacionalização e formatação de datas/números.
permission_handler	^12.0.0+1	Gerenciamento de permissões do sistema operacional.
device_info_plus	^11.3.3	Informações detalhadas sobre o dispositivo.

4.7.4.1 *Hive*

O Hive é um banco de dados leve e extremamente rápido, escrito inteiramente em *Dart*, projetado para ser usado em aplicações Flutter e *Dart* nativas. Ele segue o modelo de banco de dados chave-valor, oferecendo uma alternativa eficiente ao SQLite e a outros bancos de dados mais complexos, especialmente em contextos mobile ([LEIER; CONTRIBUTORIS, 2025](#)).

Nesse projeto, o Hive foi escolhido como opção de banco de dados local para armazenamento dos registros em momentos em que o dispositivo está offline, permitindo que até 40 registros sejam armazenados localmente e sincronizados assim que a conexão com a internet retornar. A principal vantagem do Hive é a simplicidade e velocidade em dispositivos móveis.

4.7.4.2 *Flutter Map*

O *Flutter Map* é um plugin gratuito, multiplataforma e de código aberto que permite a renderização de mapas dentro de aplicativos *Flutter*. Oferece uma interface fácil de usar para adicionar camadas, marcadores personalizados e interações em mapas. Possui suporte de diversos provedores de mapas, como *OpenStreetMap*, *Mapbox* e *Google Maps*, permitindo que os desenvolvedores escolham a fonte de dados que melhor atende às suas necessidades. Além disso, ele oferece suporte a recursos avançados, como geolocalização, desenho de rotas e manipulação de gestos do usuário, adição de *layers* ([FLUTTER_MAP, 2025](#)).

Neste projeto, o *Flutter Map* foi utilizado para renderizar o mapa com marcadores personalizados das localizações dos registros realizados pelos usuários, permitindo uma visão geográfica dos dados coletados.

4.7.4.3 *Excel*

O *excel* é uma biblioteca de código aberto para o *Dart* que permite a leitura e escrita de arquivos do Microsoft Excel no formato *XLSX* com customizações avançadas de template ([SINGH, 2021](#)). Neste projeto foi utilizada para construir a planilha de registros personalizada para download e compartilhamento.

4.7.4.4 *Graphic*

Graphic é uma biblioteca de visualização de dados para *Flutter*, baseada na Gramática de Gráficos de Leland Wilkinson. Com ela é possível criar gráficos personalizados de diversos tipos. Além disso, oferece suporte a interações avançadas como: seleção, legenda, tooltips, zoom, e animações para transições e carregamento dos gráficos ([CHEN, 2024](#)).

A aplicação desta biblioteca no projeto foi pensada para gerar uma visualização gráfica dos dados de registos coletados para maior síntese de algumas informações chaves. Para o resultado final, foram gerados gráficos nos formatos de pizza e barras.

5 RESULTADOS OBTIDOS

Este capítulo demonstra os artefatos produzidos para o *software*, incluindo as informações associadas desde a análise até os testes.

5.1 DIAGNÓSTICO DO PROCESSO ATUAL

5.1.1 Visão Geral do Processo Atual

Com base em informações obtidas através de entrevistas e conversar com os profissionais do CECLIMAR, foi possível identificar e mapear o processo atual de coleta e registro de dados da fauna marinha no litoral do Rio Grande do Sul.

Hoje, o CECLIMAR realiza o monitoramento da fauna costeira por meio de registros enviados pela comunidade, por pesquisadores em campo e por profissionais do Corpo de Bombeiros Militar. O processo ocorre de forma manual, como ilustrado na Figura 9. Esta abordagem envolve diversas etapas intermediárias e tarefas manuais realizadas por diferentes profissionais, trazendo uma maior probabilidade de erros e inconsistências nos dados coletados, além de aumentar o tempo de processamento e análise das informações.

5.1.2 Descrição do Fluxo de Trabalho Atual

O fluxo de trabalho pode ser descrito em etapas principais:

1. **Envio de dados:** Os participantes enviam fotos e informações descriptivas a partir de diversas fontes.
2. **Triagem inicial:** Um pesquisador analisa as mensagens recebidas por meio de análise manual em cada uma das fontes e seleciona os dados relevantes.
3. **Registro:** A bolsista transcreve os dados válidos para uma planilha eletrônica.
4. **Armazenamento de imagens:** As fotos dos registros são manualmente salvas em uma pasta no Google Drive com o id do registro na tabela.

5. **Validação:** O gestor revisa periodicamente os dados para verificar consistências e erros.

A Figura 9 apresenta o fluxo de trabalho atual de maneira esquemática, destacando as principais etapas e interações entre os envolvidos no processo de coleta e registro de dados. Como é possível ver na Tabela 3, atualmente, a maioria (48,45%) dos dados é coletada via WhatsApp (2024), mas esses envios também ocorrem por outras plataformas como ligações telefônicas, Facebook (2025), e-mail, e Instagram (2025).

Canal	Quantidade	Porcentagem
Ligaçāo	860	21,52%
E-mail	343	8,59%
Facebook	703	17,60%
WhatsApp	1935	48,45%
Instagram	153	3,83%
Total	3994	100%

Tabela 3 – Contagem e porcentagem de ocorrências por canal de comunicação

Fonte: Autor

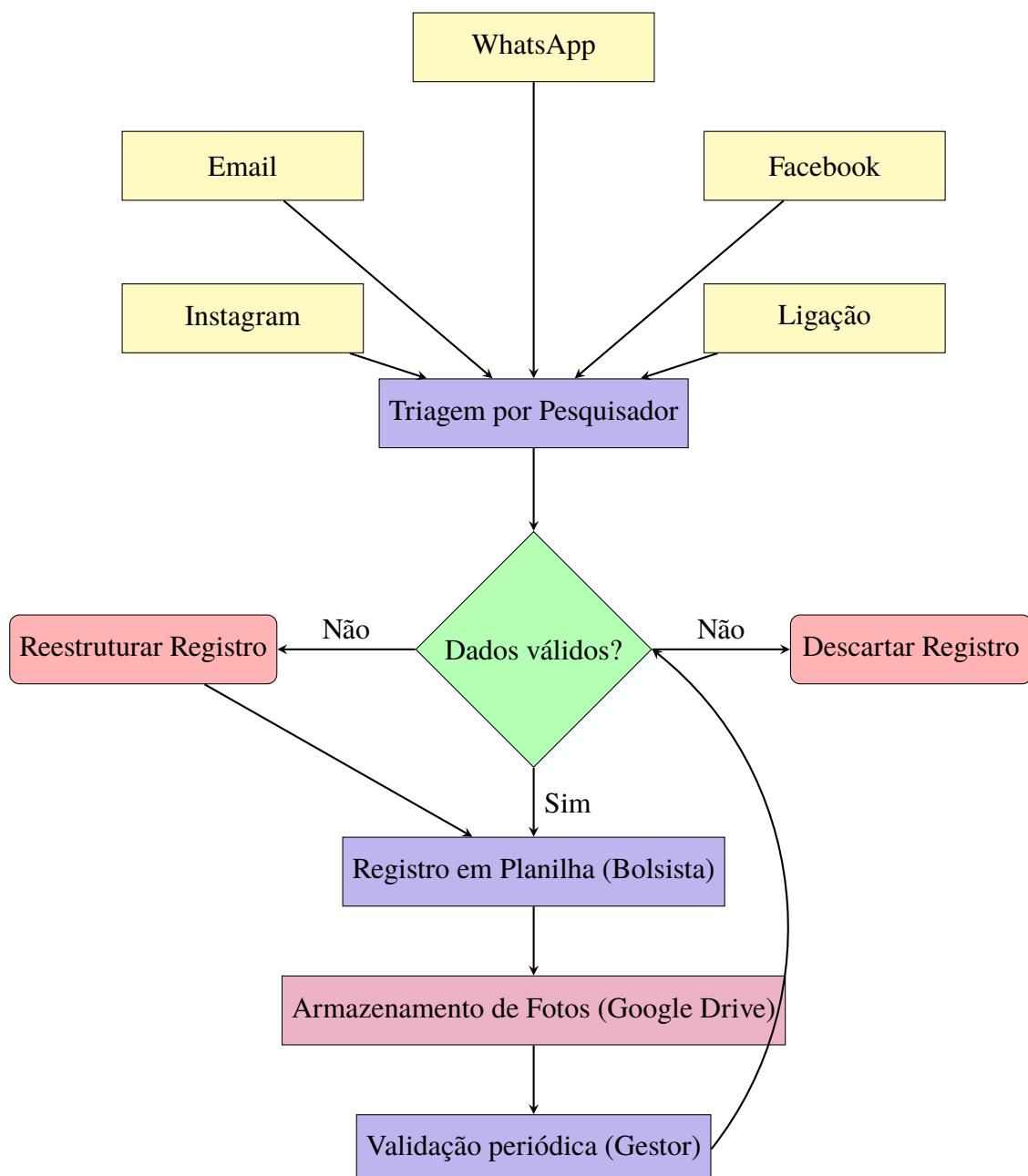


Figura 9 – Fluxograma detalhado do processo atual de coleta e registro de dados no CECLIMAR.

Fonte: Autor

5.1.3 Problemas Identificados

Os principais problemas observados neste fluxo incluem:

- Risco de erros humanos durante transcrição dos dados.
- Dificuldade de padronização nas descrições enviadas.
 - Para que a tabela automatizada funcione, os dados precisam estar exatamente iguais.
 - Erros de digitação na taxonomia dos animais.
 - Problemas com formatações de campos.
- Atraso no tempo de registro e análise das ocorrências.
- Necessidade de revisões periódicas para manter a qualidade dos dados.
- Falta de integração entre os dados (planilha e fotos separadas).
- Necessidade de acesso a múltiplas fontes para avaliar os registros.
- Pouca visibilidade da participação da comunidade no projeto.

5.2 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos foi essencial para compreender as necessidades do projeto e orientar o desenvolvimento das funcionalidades do aplicativo. Segundo Sommerville (2011), os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que ele irá oferecer e suas restrições de funcionamento. Neste projeto, a definição dos requisitos prioritários para o MVP¹ e das suas limitações foi orientada por reuniões com o coordenador do projeto Fauna Marinha RS, especialmente com base na experiência prática dos profissionais do CECLIMAR, que indicaram as funcionalidades mais relevantes para o sistema.

5.2.1 Requisitos Funcionais

Nesta seção, serão expostos os requisitos funcionais do sistema, que segundo Guedes (2018) correspondem ao que o cliente quer que o sistema realize, ou seja, as

¹ MVP é a sigla para *Minimum Viable Product* (Produto Mínimo Viável), um conceito que se refere à versão mais simples de um produto que ainda entrega valor ao usuário e permite validações com o mínimo de esforço e custo.

funcionalidades do software. Esses requisitos estão organizados em tabelas distintas, apresentadas a seguir. A Tabela 4 sintetiza os 63 requisitos funcionais da versão 2.3.6 do aplicativo, que é a versão mais atualizada até o momento da entrega deste relatório. Durante o desenvolvimento, os requisitos foram revisados e atualizados conforme necessário, com base em feedbacks e testes realizados.

Tabela 4 – Requisitos funcionais do sistema

Código	Descrição
RF01	O sistema deve permitir o registro de novos usuários com e-mail e senha.
RF02	O sistema deve permitir o registro de novos usuários com conta Google integrada ao Firebase Authentication.
RF03	O sistema deve diferenciar dois tipos de usuários: cientista cidadão (user) e pesquisador (admin).
RF04	O sistema deve permitir que um pesquisador cadastre outro pesquisador, com geração automática de senha segura (8 dígitos alfanuméricos e caracteres especiais).
RF05	O sistema deve permitir que um pesquisador conceda a outro usuário que já possui conta a role de pesquisador.
RF06	O sistema deve permitir o login via Firebase Authentication com e-mail e senha.
RF07	O sistema deve permitir o login via conta Google integrada ao Firebase Authentication.
RF08	O sistema deve unificar dados quando uma conta possui ambos os métodos de login.
RF09	O sistema deve permitir redefinição de senha através do fluxo do Firebase Authentication (envio de e-mail de redefinição).
RF10	O sistema deve permitir a recuperação da senha.
RF11	O sistema deve permitir que o usuário faça logoff.
RF12	O sistema deve permitir a exclusão da conta do usuário, independente do método de cadastro.
RF13	O sistema deve manter os registros realizados pelo usuário mesmo após a exclusão da conta, para preservação de dados científicos.

(Continua na próxima página)

Código	Descrição
RF14	O sistema deve permitir atualização do perfil de usuário cadastrado com e-mail e senha (nome, foto, e-mail).
RF15	O sistema deve permitir ao usuário visualizar e gerenciar seu perfil.
RF16	O sistema deve possuir um conjunto de conquistas para cada usuário.
RF17	O sistema deve permitir ao usuário visualizar os últimos registros do seu perfil.
RF18	O sistema deve permitir ao usuário visualizar a quantidade total de registros realizadas por ele.
RF19	O sistema deve permitir envio de registros "simples".
RF20	O sistema deve permitir envio de registros "técnicos" com dados mais específicos.
RF21	O sistema deve permitir envio de até 2 imagens por registro, sendo uma obrigatória.
RF22	O sistema deve permitir envio de registros utilizando GPS do dispositivo em tempo real para obter as coordenadas geográficas.
RF23	O sistema deve validar os formulários de envio de registros com restrições específicas (ex.: limite de caracteres).
RF24	O sistema deve permitir que o usuário envie uma imagem a partir da galeria do celular.
RF25	O sistema deve permitir que o usuário envie uma imagem a partir da camera do celular.
RF26	O sistema deve permitir consultar dados de geolocalização a partir das.
RF27	O sistema deve permitir envio de registros utilizando a opção número da guarita para basear as coordenadas geográficas.
RF28	O sistema deve permitir envio de registros utilizando coordenadas da cidade para definir a localização.
RF29	O sistema deve permitir envio de registros utilizando ponto de referência como campo aberto.
RF30	O sistema deve facilitar a ativação da geolocalização do dispositivo pelo usuário.
RF31	O sistema deve permitir visualizar a imagem de um registro em tela cheia.

(Continua na próxima página)

Código	Descrição
RF32	O sistema deve exibir o nome da cidade automaticamente com base na localização do registro.
RF33	O sistema deve permitir o armazenamento de registros quando enviados offline.
RF34	O sistema deve apresentar uma tela com informações sobre o projeto e os desenvolvedores.
RF35	O sistema deve permitir que o usuário visualize todos registros que ele enviou.
RF36	O sistema deve permitir que o usuário filtre os registros enviados e já validados.
RF37	O sistema deve permitir que o usuário receba o retorno do profissional para cada registro.
RF38	O sistema deve permitir que um pesquisador delete um registro enviado por qualquer usuário.
RF39	O sistema deve permitir a visualização detalhada de um registro selecionado.
RF40	O sistema deve apresentar uma barra de navegação fixa com as opções: início, registros, novo registro, perfil e sobre.
RF41	O sistema deve apresentar um menu inicial com todas as funcionalidades disponíveis.
RF42	O sistema deve assegurar que apenas usuários pesquisadores tenham acesso a algumas funcionalidades.
RF43	O sistema deve possuir uma seção para apresentar os principais animais da fauna local.
RF44	O sistema deve possuir uma tela de carregamento inicial.
RF45	O sistema deve permitir que o usuário visualize os registros enviados por outros usuários.
RF46	O sistema deve disponibilizar um painel de registros com estatísticas, mapas e filtros detalhados para acompanhamento.
RF47	O sistema deve permitir que os usuários comuns exportem um arquivo CSV com dados sensíveis ocultos.
RF48	O sistema deve permitir que os pesquisadores exportem um arquivo CSV com os dados gerais dos registros.

(Continua na próxima página)

Código	Descrição
RF49	O sistema deve permitir que sejam vistos todos os registros enviados ainda não avaliados.
RF50	O sistema deve permitir que sejam vistos todos os registros já avaliados.
RF51	O sistema deve apresentar um mapa interativo com pontos de registros.
RF52	O sistema deve apresentar os animais que possuem mais registros documentados.
RF53	O sistema deve incrementar o número de encontros com cada espécie após uma avaliação de pesquisador.
RF54	O sistema deve possuir um contador por classe para os mamíferos, as aves e os répteis.
RF55	O sistema deve permitir a busca de registros por espécie.
RF56	O sistema deve permitir a busca de número de registros dentro de uma faixa de tempo.
RF57	O sistema deve permitir que pesquisadores avaliem de todos os registros enviados.
RF58	O sistema deve permitir a inserção de novos animais no banco de dados.
RF59	O sistema deve permitir que o pesquisador adicione comentários aos registros enviados.
RF60	O sistema deve permitir que os dados enviados por um usuário sejam editáveis para um pesquisador realizar a avaliação.
RF61	O sistema deve permitir ao pesquisador atualizar a localização ao avaliar um registro.
RF62	O sistema deve redirecionar para o site do projeto Fauna Marinha RS para dados da fauna local.
RF63	O sistema deve redirecionar para as redes sociais do projeto Fauna Marinha RS.

Fonte: Autor

5.2.2 Requisitos Não Funcionais

Segundo Pressman (2011), os requisitos não funcionais podem ser descritos como uma característica de qualidade, desempenho, segurança ou restrição geral de um sistema. Na Tabela 5, temos o detalhamento desses requisitos, com a finalidade de sintetizar as propriedades essenciais que asseguram o bom funcionamento da aplicação. Tais requisitos são fundamentais para garantir aspectos como usabilidade, confiabilidade, portabilidade e eficiência, além de assegurar conformidade com as tecnologias e práticas adotadas no desenvolvimento do sistema.

Tabela 5 – Requisitos não funcionais do sistema

Código	Descrição
RNF01	O sistema deve ser desenvolvido utilizando Flutter e Dart, garantindo compatibilidade multiplataforma.
RNF02	O sistema deve utilizar autenticação segura via Firebase, com criptografia adequada para senhas e tokens.
RNF03	O sistema deve apresentar alta disponibilidade e ser resiliente a falhas de rede.
RNF04	O sistema deve indicar quando alguma funcionalidade não está disponível.
RNF05	O sistema deve seguir as melhores práticas de UX, com validação clara de formulários e feedbacks visuais para os usuários.
RNF06	O sistema deve garantir a privacidade dos dados do usuário, omitindo dados sensíveis em exportações para cientistas cidadãos.
RNF07	O sistema deve apresentar desempenho aceitável mesmo em dispositivos móveis de média capacidade.
RNF08	O sistema deve fornecer acessibilidade básica, permitindo navegação simples e intuitiva.
RNF09	O sistema deve apresentar feedbacks constantes sobre os status de processamento da aplicação.
RNF10	O sistema deve ser inicialmente disponibilizado para Android.

Fonte: Autor

5.3 PROTOTIPAÇÃO DA INTERFACE COM O USUÁRIO

A prototipação foi dividida em quatro etapas principais, com foco na representação visual da aplicação e na simulação dos fluxos de navegação dos usuários. A seguir, são descritas as fases de prototipação realizadas.

5.3.1 Definição da Identidade Visual

A identidade visual do aplicativo foi construída com base nos elementos gráficos já existentes no projeto Fauna Marinha RS (Figura 10). Foram utilizadas cores similares e o logotipo oficial do projeto, com o objetivo de manter a coerência visual entre as diferentes aplicações e garantir o reconhecimento por parte dos usuários já familiarizados com a identidade do projeto.

A paleta de cores foi definida de modo a remeter ao ambiente marinho, utilizando tons de azul e uma textura que remete à água. Essa paleta é apresentada na Figura 12.

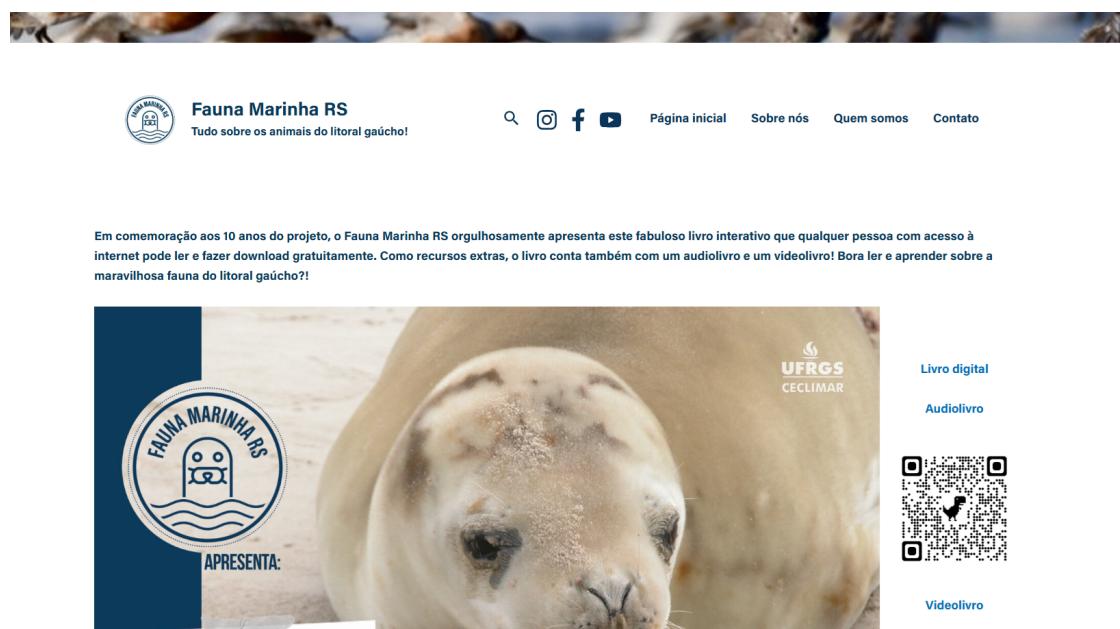


Figura 10 – Identidade visual do sistema web do projeto Fauna Marinha RS.

Fonte: (Fauna Marinha RS, 2025)



Figura 11 – Identidade visual prototipada.

Fonte: Autor

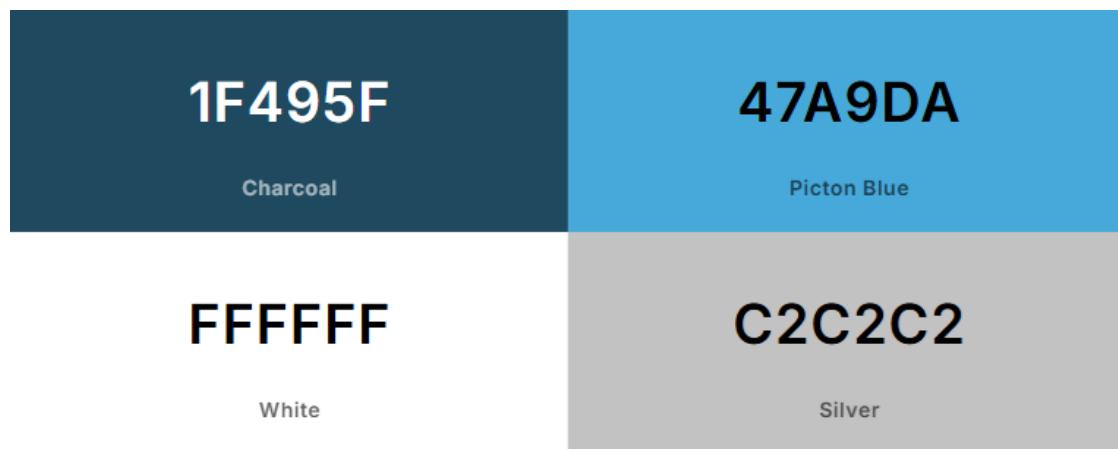


Figura 12 – Paleta de cores utilizada.

Fonte: Autor

5.3.2 Criação do Design das Telas

Com base na identidade visual definida, foram elaborados os layouts das principais telas do aplicativo. Essa etapa concentrou-se na organização dos elementos de interface, na usabilidade e na experiência do usuário.

Inicialmente, foram criadas as telas de login (Figura 13) e de registro de usuário (Figura 14).

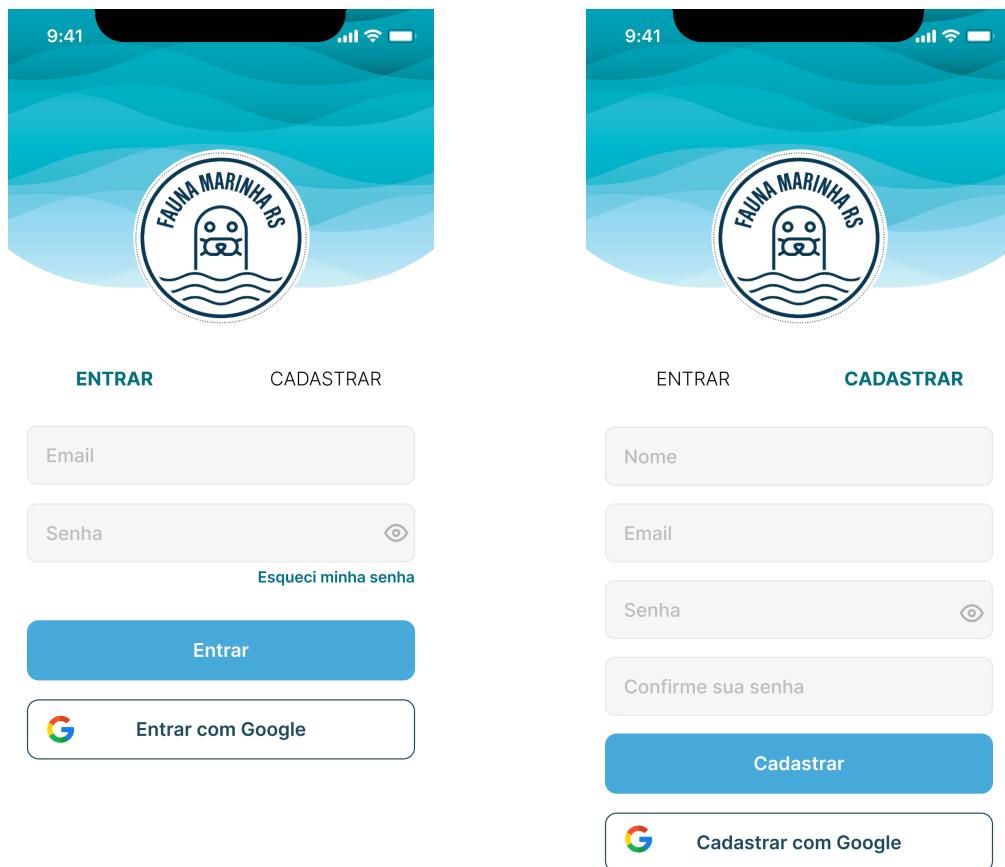


Figura 13 – Protótipo da tela de login do aplicativo.

Figura 14 – Protótipo da tela de cadastro do aplicativo.

Fonte: Autor

Em seguida, foi criada a tela inicial (Figura 15), com duas variações: uma para usuários comuns e outra para pesquisadores. O layout foi pensado para facilitar o acesso

às funcionalidades, com botões de acesso rápido e barra de navegação fixa na parte inferior da tela. A versão para pesquisadores apresenta os mesmos botões, com a adição das funcionalidades exclusivas, avaliação de registros e acesso ao painel geral.

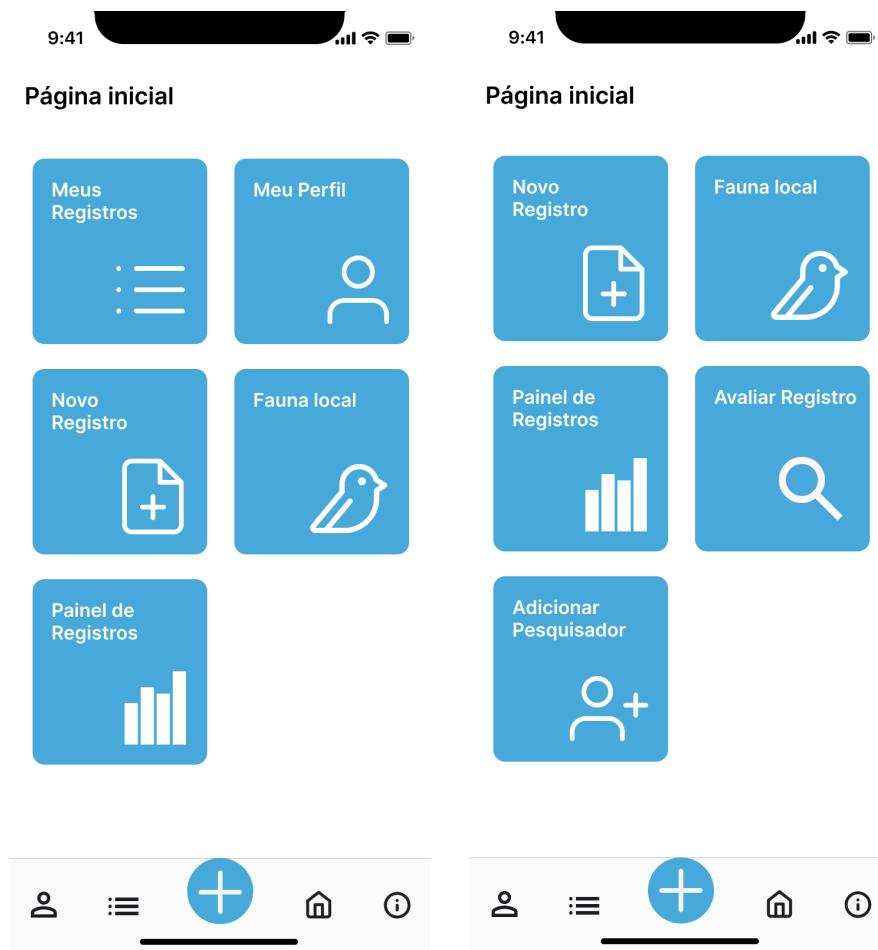


Figura 15 – Protótipo da tela inicial: usuário comum (esquerda) e pesquisador (direita).

Fonte: Autor

A tela "Meus Registros"(Figura 16) permite ao usuário visualizar todos os registros já enviados, com filtros por status (enviado e validado). Também foi projetado um estado com *bottomsheet* acionado a partir do botão de adicionar registro ("+") na barra de navegação. Esse botão está disponível em qualquer tela, permitindo adicionar registros rapidamente.

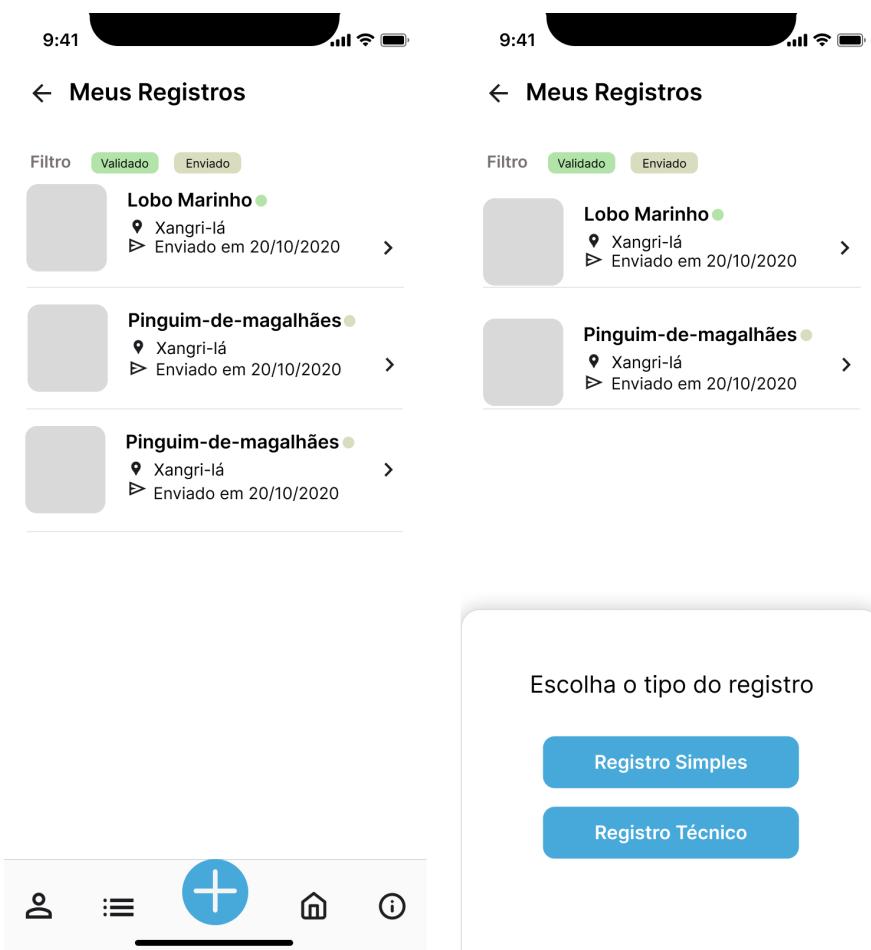


Figura 16 – Protótipo das telas de "Meus Registros" com estado ativo de adição de registro (direita).

Fonte: Autor

Dois tipos de formulários de registro foram projetados: um simples (Figura 17), voltado a usuários leigos e envios rápidos, e outro técnico (Figura 18), com campos adicionais para coleta de dados mais detalhados. Ambos possuem validações, campos obrigatórios e opcionais, e suporte ao envio de imagens com orientações apresentadas em *bottomsheet*.

Os campos específicos de classe, ordem, família e gênero foram projetados para serem um dropdown, que apresentará as opções disponíveis para o usuário selecionar.

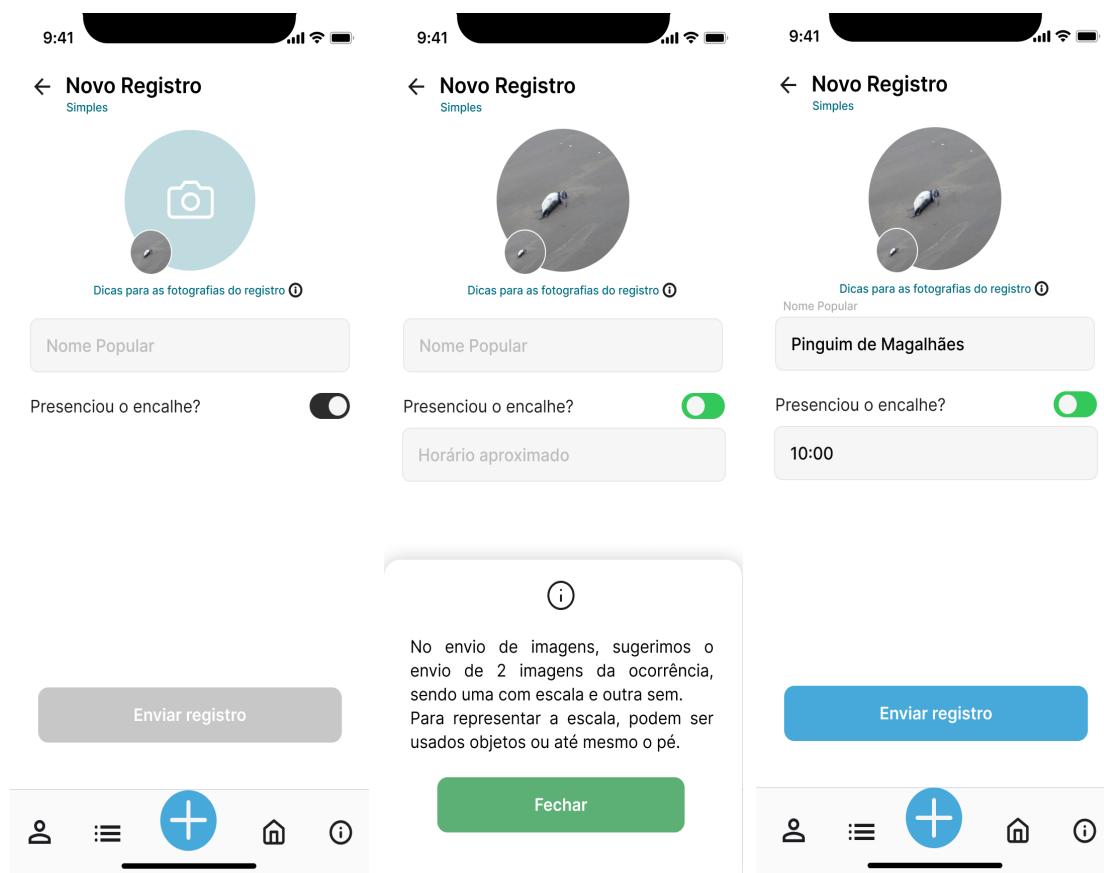


Figura 17 – Protótipo do formulário de registro simples. Centro: *bottomsheet* informativa; direita: campo adicional ativado.

Fonte: Autor

O protótipo apresenta duas telas de "Novo Registro Técnico". A interface é clean, com uma barra superior preta com o tempo (9:41) e ícones de conexão. Cada tela tem uma barra de topo com o título "← Novo Registro Técnico".

A primeira tela (esquerda) inclui:

- Campo para upload de foto com placeholder "Nome Popular".
- Campo para "Espécie".
- Campo para "Município" e "Nº Guarita".
- Botão para "Presenciou o encalhe?" com placeholder "Observações (Opcional)".
- Botão azul "Enviar registro".

A segunda tela (direita) é similar, com o seguinte conteúdo adicional:

- Botão para "Presenciou o encalhe?" com placeholder "Horário aproximado".
- Campo para "Observações (Opcional)".
- Campo suspenso para "Classe (Opcional)".
- Campo para "Ordem (Opcional)".
- Campo para "Família (Opcional)".
- Campo para "Gênero (Opcional)".

As barras de navegação contêm ícones para usuário, menu, casa e ajuda.

Figura 18 – Protótipo do formulário de registro técnico com campos taxonômicos e detalhamento avançado.

Fonte: Autor

A tela de "Registros Pendentes"(Figura 19) foi projetada para uso exclusivo de pesquisadores, permitindo acesso aos registros ainda não avaliados enviados por todos os usuários.



Figura 19 – Protótipo da tela de registros pendentes, acessível apenas a pesquisadores.

Fonte: Autor

A tela de "Analizar Registro"(Figura 20) foi projetada para permitir que o pesquisador visualize os dados enviados em cada registro com a possibilidade de editar os dados, adicionar comentários e enviar uma atualização o registro. Se fez importante adicionar um estado para esta tela, onde o pesquisador consiga selecionar a imagem para uma visualização em tela cheia, facilitando a análise do registro.

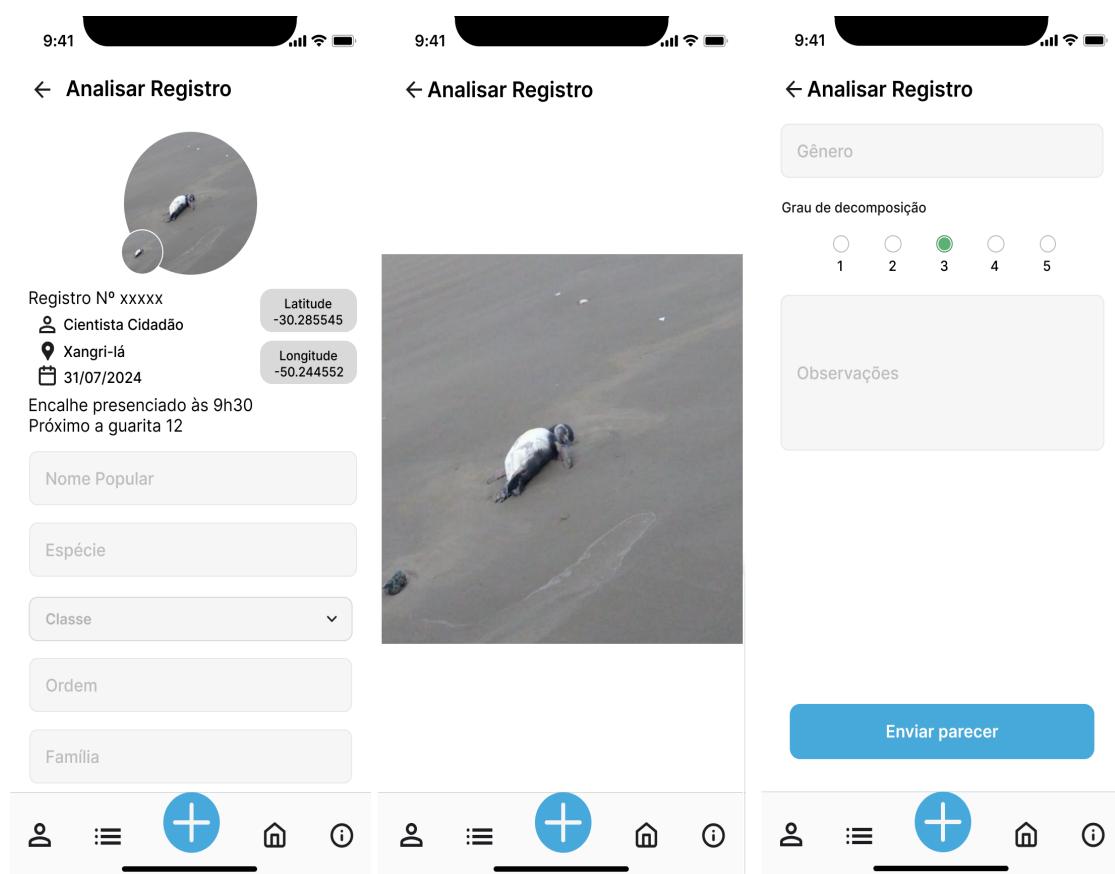
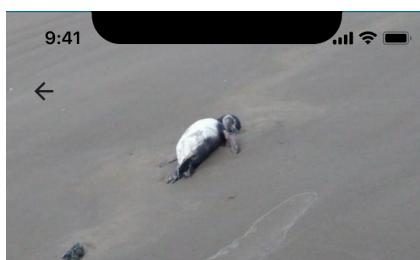


Figura 20 – Protótipo da tela de análise de registro. Ao centro, visualização ampliada da imagem.

Fonte: Autor

A tela de "Visualizar Registro" foi projetada para permitir que o usuário visualize os dados enviados e validados. Para os dados já foram validados (Figura 21), o registro apresentará dados extras sobre aquele registro, como grau de decomposição, parecer do profissional e taxonomia avaliada. Já para os dados que ainda não foram validados (Figura 22), o usuário verá apenas as informações que ele mesmo enviou.

**Pinguim-de-magalhães**

Registro Nº xxxx

Cientista Cidadão

Xangri-lá

31/07/2024

Latitude
-30.285545Longitude
-50.244552

Validado

Nome Popular: Pinguim

Espécie: Pinguim-de-magalhães

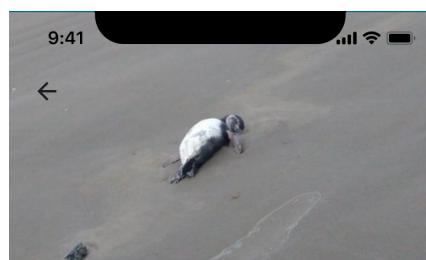
Encontrado próximo a guarita X

Grau de decomposição 3

Parecer do profissional

Animal de espécie XXXXX classe XXXX ordem XXXXX família XXXX gênero XXXXX. O estado de conservação deste registro é X.

Foram encontrados outros X registros desta espécie na região

**Pinguim-de-magalhães**

Registro Nº xxxx

Cientista Cidadão

Xangri-lá

31/07/2024

Latitude
-30.285545Longitude
-50.244552

Enviado

Nome Popular: Pinguim

Encontrado próximo a guarita X



Figura 21 – Protótipo da visualização de registro validado.



Figura 22 – Protótipo da visualização de registro enviado, ainda não avaliado.

Fonte: Autor

A tela de perfil do usuário (Figura 23) apresenta dados básicos, quantidade de registros enviados, últimos registros e conquistas. As conquistas são exibidas em formato de medalhas organizadas em uma grade com três colunas.

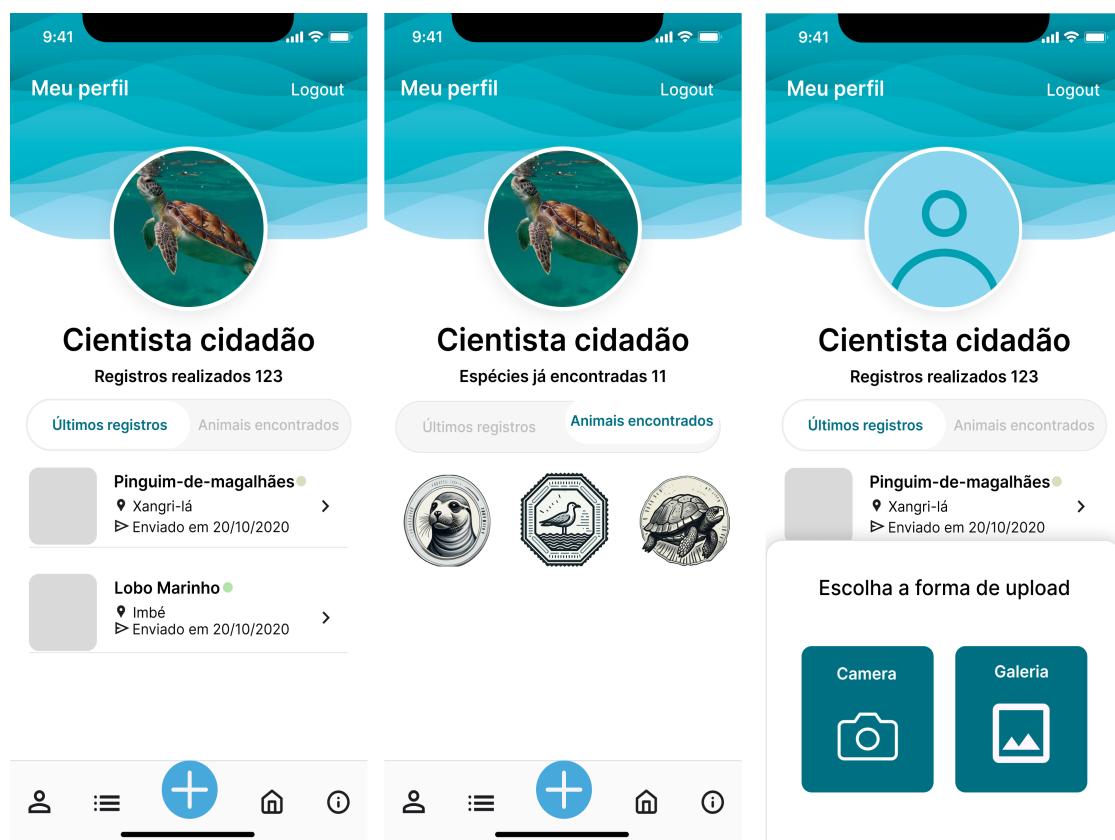


Figura 23 – Protótipo da tela de perfil do usuário com conquistas e histórico.

Fonte: Autor

As telas de fauna local (Figura 24) seguem o mesmo padrão visual das demais telas do aplicativo, se assemelhando em estrutura com a de "Meus Registros" e "Visualizar Registro". Elas apresentam, respectivamente, uma lista de espécies e uma tela de detalhes com informações específicas sobre cada animal que for selecionado.

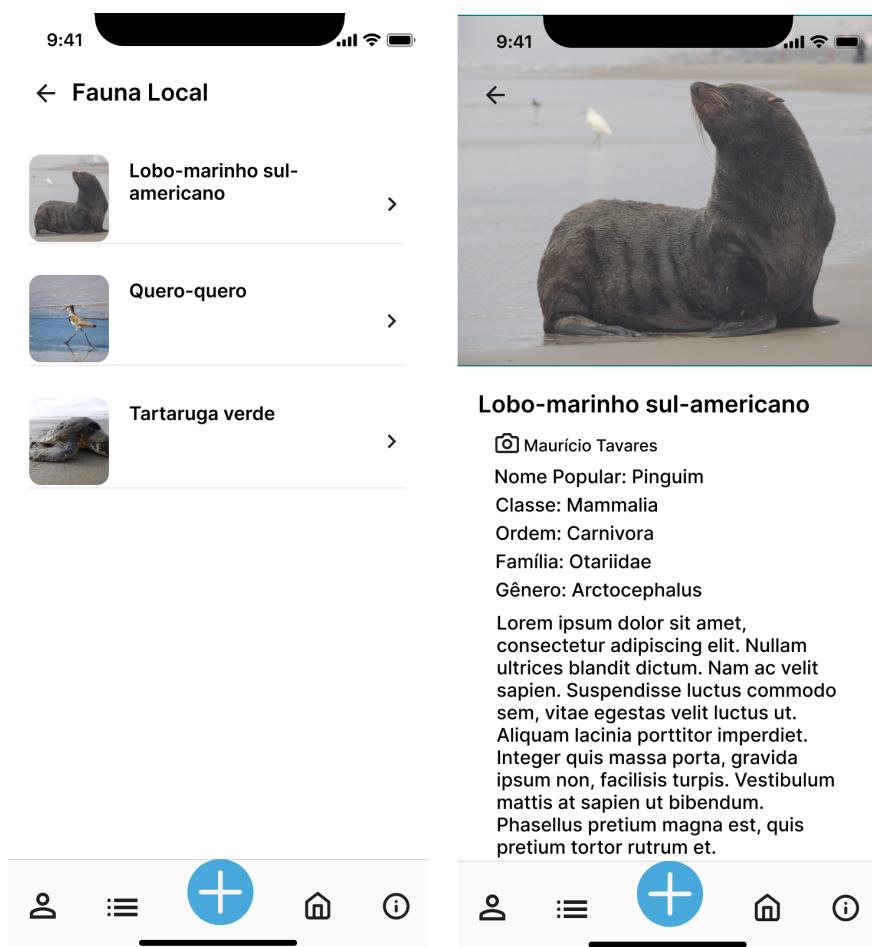


Figura 24 – Protótipo da tela de fauna local (esquerda) e detalhes da espécie (direita).

Fonte: Autor

Por fim, foi projetada a tela de recuperação de senha (Figura 25), incluindo os estados de sucesso e de erro na validação dos campos de entrada. O padrão de erro apresentado nesta tela foi o mesmo utilizado em outras telas do aplicativo.

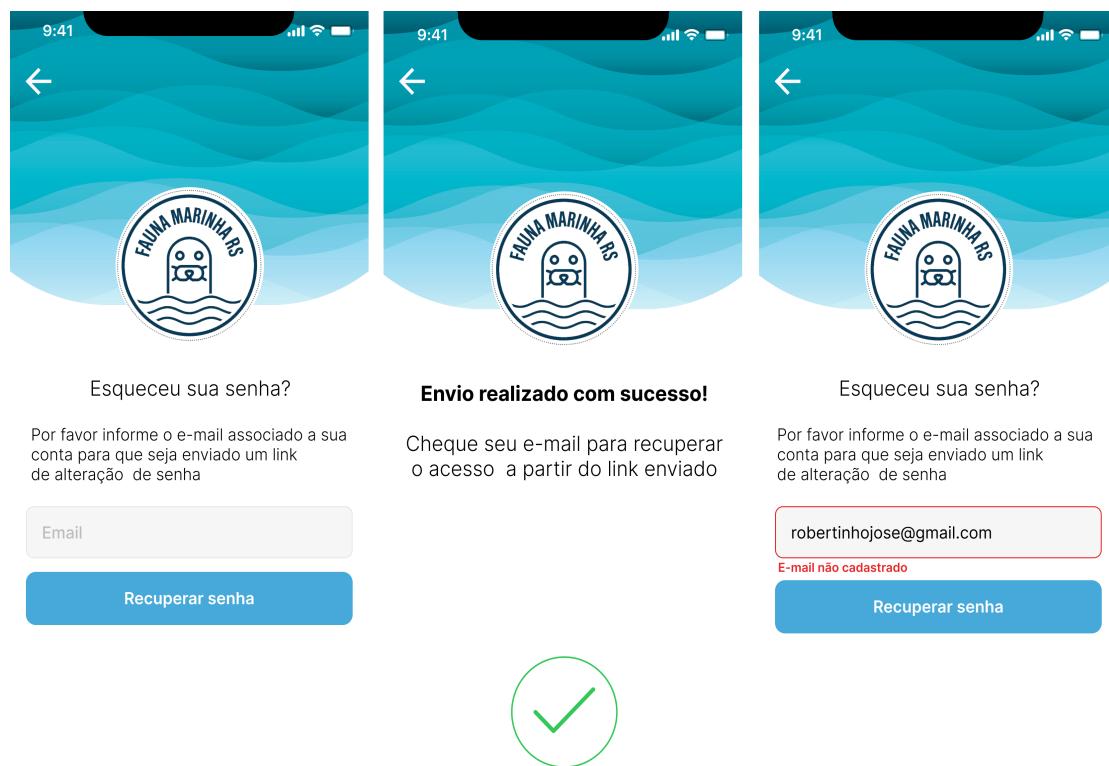


Figura 25 – Protótipo da tela de recuperação de senha: formulário (esquerda), sucesso (centro) e erro de validação (direita).

Fonte: Autor

5.3.3 Desenvolvimento do Protótipo Navegável

Utilizando a ferramenta Figma (versão gratuita *online*), foi desenvolvido um protótipo navegável de média/alta fidelidade. Essa etapa permitiu visualizar os fluxos de interação entre o usuário e o sistema, antecipar ajustes necessários e alinhar as funcionalidades às expectativas.

Foram criadas ligações interativas entre as principais telas, simulando os redirecionamentos e ações de navegação. O protótipo também serviu como ferramenta de validação junto a terceiros e como referência visual para a fase de implementação (Figura 26).

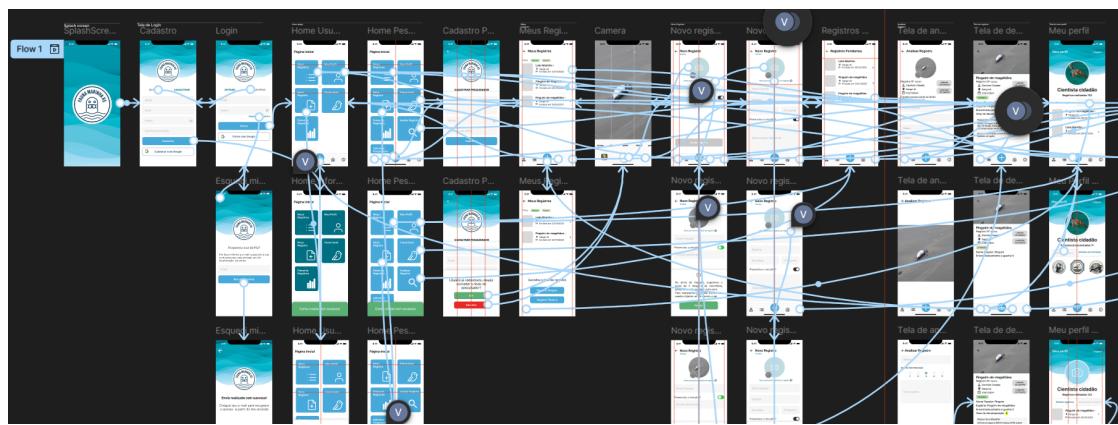


Figura 26 – Captura de tela do protótipo navegável desenvolvido no Figma.

Fonte: Autor

5.4 PROJETO GERENCIAL

A gerência desse projeto foi realizada utilizando a ferramenta Jira, que possibilitou o controle de tarefas e o acompanhamento do progresso do desenvolvimento. O fluxo de trabalho descrito na [Seção de metodologia](#) deste trabalho foi seguido para garantir a organização e a eficiência das tarefas. Os *cards* gerados no *board* foram puxados para trabalho a medida que havia tempo disponível para atuação. Foram catalogadas não só tarefas de codificação, como também tarefas de prototipação.

Foram gerados ao todo, 131 *cards* desde julho de 2024 quando se deu inicio a prototipação da aplicação. Na Figura 27 é possível observar a quantidade de itens no eixo vertical relacionada com o mês de criação no eixo horizontal. Com ele podemos concluir que os meses com mais entrada de demandas foram os meses de novembro de 2024, janeiro de 2025 e março de 2025, onde o número de *cards* criados foi maior que 18.

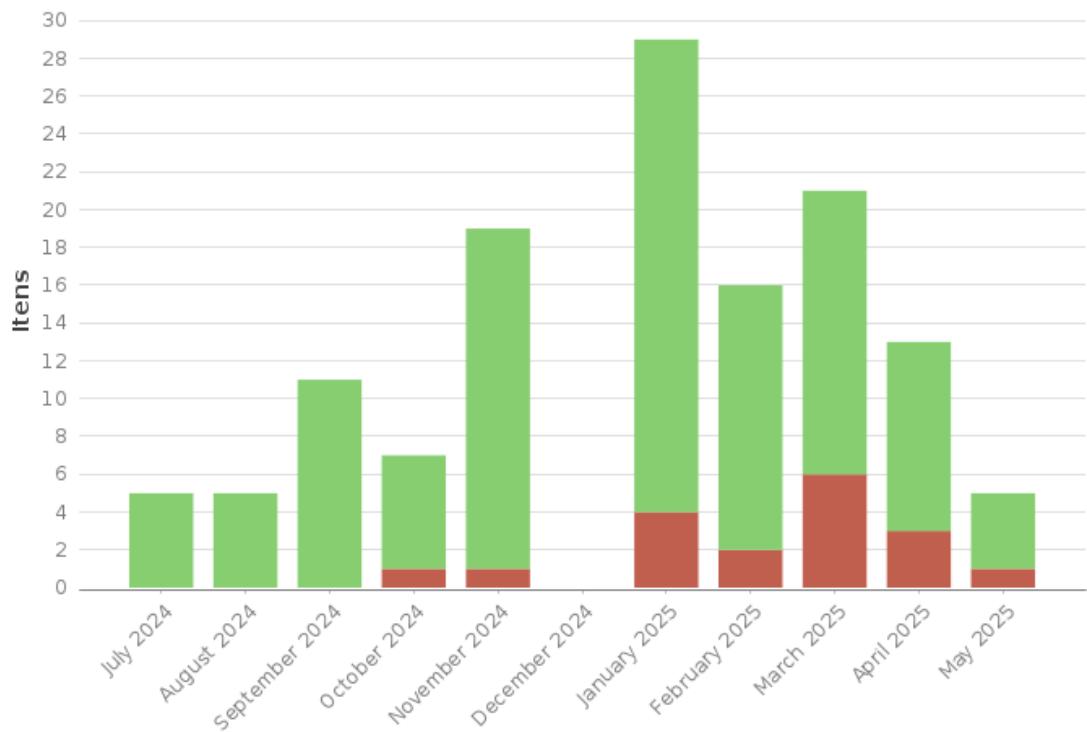
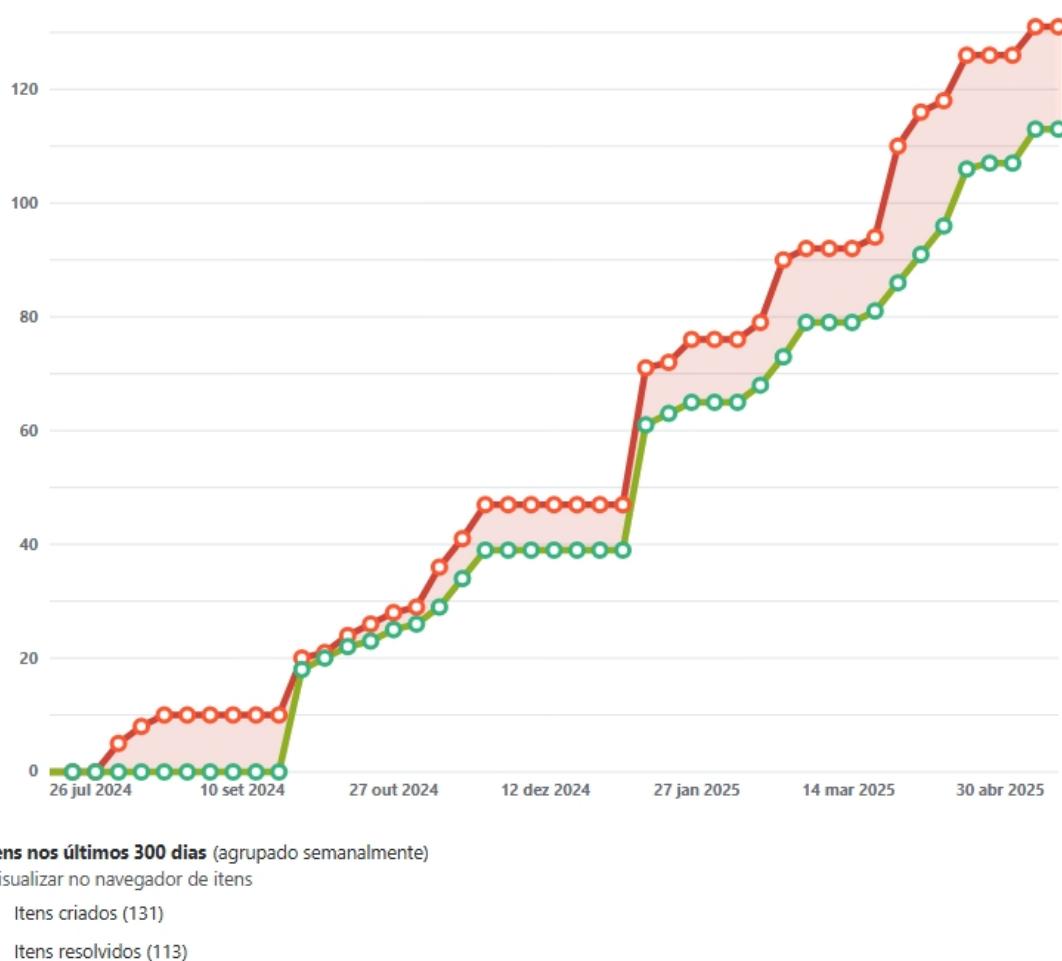


Figura 27 – Gráfico demonstrativo de *cards* criados por mês no Jira. As barras em verde representam demandas que, até a data de criação do gráfico, já haviam sido finalizadas. As barras em vermelho representam demandas que ainda estavam em aberto.

Fonte: Autor

A Figura 28 traz uma visão cumulativa do trabalho realizado no board. A partir dele podemos ver que, dos 131 cards criados, 113 foram concluídos, o que representa 86,26% do total. Podemos observar também que, em momentos de maior atuação no projeto, a criação de cards está diretamente relacionada com a quantidade de cards resolvidos. Isso está principalmente relacionado com a disponibilidade de tempo para atuar no projeto, e também com a criação de bugs e novas demandas que surgiram durante o desenvolvimento.

Figura 28 – Gráfico demonstrativo de *cards* vs resolvidos no Jira

Fonte: Autor

A Figura 29 apresenta o tempo médio, em dias, para a resolução dos *cards* ao longo das semanas. É possível notar uma grande variação nos tempos médios, com valores baixos no período de novembro e um pico em maio de 2025. Esse aumento progressivo reflete a falta de atuação evidenciada no mês de dezembro vide Figura 28, o aumento na complexidade das demandas e o acúmulo de cards de melhoria no *Backlog* que foram criados e não priorizados.

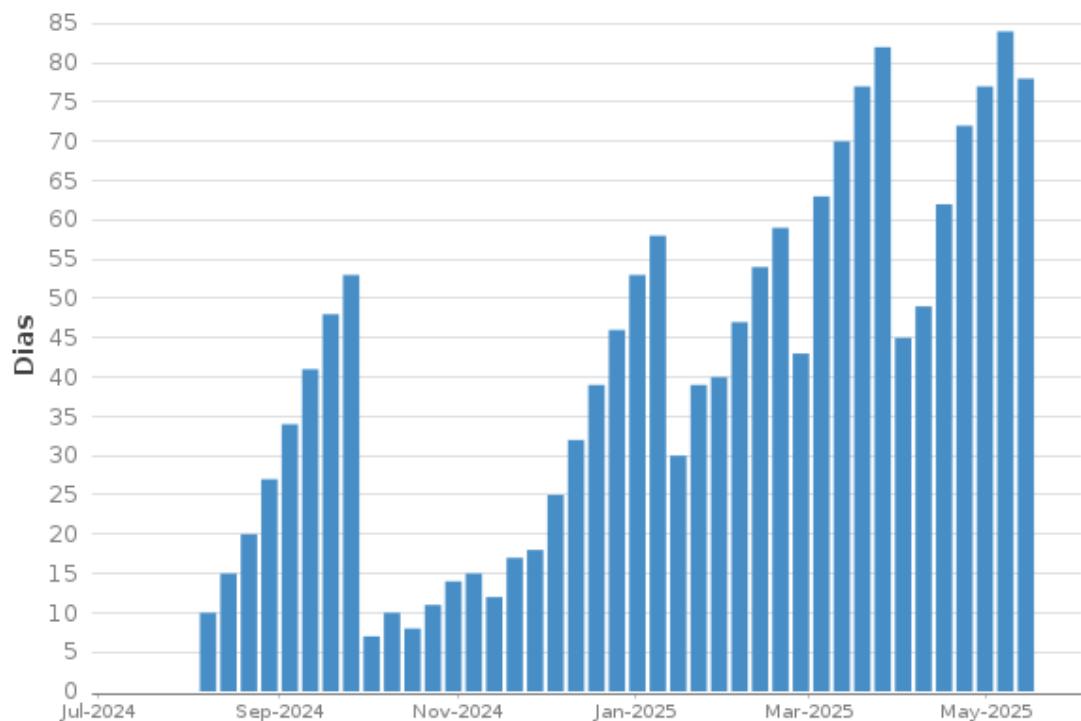


Figura 29 – Gráfico demonstrativo de tempo de resolução médio dos *cards* por semana no Jira

Fonte: Autor

Os *cards* foram organizados em três grupos principais, que representam as diferentes atuações necessárias para atuar em cada uma delas.

5.4.1 Desenvolvimento Geral

Esses cards representam demandas que passaram por etapas de planejamento e refinamento, refletindo as histórias de usuário e tarefas técnicas previstas ou requisitadas durante o desenvolvimento. A Tabela 6 apresenta esse conjunto de funcionalidades. Ao todo foram 66 demandas sendo 62 concluídas e 4 que ainda estão no *Backlog*.

Tabela 6 – Registro de desenvolvimento geral (Ordenado por Data de Entrega)

Chave	Resumo	Status	Entregue
CEC-1	Prototipação inicial das telas	Concluído	2024-09-27
CEC-2	Tela de registro	Concluído	2024-09-27
CEC-3	Tela de login	Concluído	2024-09-27
CEC-4	Tela de novo registro Simples	Concluído	2024-09-27
CEC-5	Tela de meus registros	Concluído	2024-09-27
CEC-6	Tela de Coletanea animais	Concluído	2024-09-27
CEC-7	Tela de perfil do usuário	Concluído	2024-09-27
CEC-8	Tela de adicionar pesquisador	Concluído	2024-09-27
CEC-9	Tela de esqueci minha senha	Concluído	2024-09-27
CEC-10	Tela de selos de conquista	Concluído	2024-09-27
CEC-11	Tela de novo registro Técnico	Concluído	2024-09-27
CEC-12	Tela de home	Concluído	2024-09-27
CEC-13	Tela de analisar Registro	Concluído	2024-09-27
CEC-14	Tela de SplashScreen	Concluído	2024-09-27
CEC-15	Tela sobre o app	Concluído	2024-09-27
CEC-16	Codificação da SplashScreen	Concluído	2024-09-27
CEC-17	Codificação da Tela de Registro	Concluído	2024-09-28
CEC-18	Codificação da Tela de Login	Concluído	2024-09-28
CEC-21	Codificação esqueci minha senha	Concluído	2024-09-29
CEC-20	Codificação da Tela de Cadastrar Pesquisador	Concluído	2024-10-03
CEC-19	Codificação da HomePage	Concluído	2024-10-07
CEC-22	Adição funcionalidade de adicionar novo registro + bottom sheet	Concluído	2024-10-09
CEC-23	Tela de registro simples	Concluído	2024-10-15
CEC-24	Tela de registro tecnico	Concluído	2024-10-21
CEC-25	Tela de registro controller	Concluído	2024-10-21
CEC-28	Tela de perfil do usuário	Concluído	2024-11-02

(Continua na próxima página)

Tabela 6 – (Continuação) Registro de desenvolvimento geral (Ordenado por Data de Entrega)

Chave	Resumo	Status	Entregue
CEC-30	Mensagens de feedback para usuario: sucesso e erro	Concluído	2024-11-23
CEC-29	Tela Meus registros	Concluído	2024-11-03
CEC-31	Criar listas para os itens da tela de perfil	Concluído	2024-11-03
CEC-33	Tela view de Registro	Concluído	2024-11-08
CEC-35	Texto da tela de sobre o app	Concluído	2024-11-16
CEC-38	Adicionar dependencia e configurar para captar a localizacao	Concluído	2024-11-10
CEC-34	Criar montagem do registro + cadastrar registro	Concluído	2024-11-11
CEC-40	Tela de avaliar registro	Concluído	2024-11-15
CEC-39	Adicionar botao para remover imagem adicionada	Concluído	2024-11-16
CEC-42	Funcionalidade de deletar conta	Concluído	2024-11-23
CEC-44	Tratamento de erros do login firebase	Concluído	2024-11-23
CEC-41	Adicionar badge para o icone de meus registros	Concluído	2024-11-23
CEC-48	Criação do BD firebase	Concluído	2025-01-07
CEC-53	Ordenação por data dos registros	Concluído	2025-01-08
CEC-59	Adicionar skeleton de carregamento em alguns widgets	Concluído	2025-01-09
CEC-46	Lógica de adição de pesquisador + lógica de conceder role de pesquisador pra usuário já existente	Concluído	2025-01-10
CEC-66	Adição de uma badge para avisar registros não visualizados	Concluído	2025-01-11
CEC-60	Adicionar facilidade para ativar a geolocalização do dispositivo	Concluído	2025-01-11
CEC-78	Skeletonizer tela inicial	Concluído	2025-02-14

(Continua na próxima página)

Tabela 6 – (Continuação) Registro de desenvolvimento geral (Ordenado por Data de Entrega)

Chave	Resumo	Status	Entregue
CEC-98	Geração de csv para exportação	Concluído	2025-03-21
CEC-54	Tela painel de registros	Concluído	2025-03-23
CEC-110	Adicionar update de localização na avaliação do registro	Concluído	2025-03-31
CEC-116	Atualização da versão do projeto*	Concluído	2025-03-31
CEC-111	Adicionar o campo das quaritas/ municipio para avaliar registro	Concluído	2025-03-31
CEC-118	Adição de campo livre ponto de referencia	Concluído	2025-04-06
CEC-112	Migração dos dados atuais para o firebase	Refinamento	2025-04-06
CEC-109	Deletar registros	Concluído	2025-04-07
CEC-125	badges para classes no perfil com contador	Concluído	2025-04-17
CEC-106	Possibilidade de adicionar um novo animal não presente no banco de dados quando for identificado	Concluído	2025-04-18
CEC-114	Modificar restrições do csv	Concluído	2025-04-18
CEC-108	Filtrar quantidade por classe	Concluído	2025-04-18
CEC-55	Criação de badges para os animais	Concluído	2025-04-23
CEC-49	Implementar cache dos registros enviados quando não há internet	Concluído	2025-05-12
CEC-37	Realizar integração front para receber dados da API baseado em critérios dos inputs	Concluído	2025-05-12
CEC-76	Ajustes de responsividade devices menores	Concluído	2025-05-12
CEC-107	Adicionar pedido para que a pessoa não mostre seu rosto no informativo da foto	Concluído	2025-05-12
CEC-81	Desenvolvimento de redirect para opção de fauna local	Concluído	2025-05-12
CEC-58	Tela fauna local	Backlog	
CEC-115	Gerar termos de uso	Backlog	

(Continua na próxima página)

Tabela 6 – (Continuação) Registro de desenvolvimento geral (Ordenado por Data de Entrega)

Chave	Resumo	Status	Entregue
CEC-105	Adicionar campo de tipo de usuário que realizou o envio do registro	Backlog	
CEC-113	Mapa para marcar local aproximado no envio do registro	Backlog	

5.4.2 Correções

Ao longo do processo de desenvolvimento, foram identificados e registrados bugs e falhas de funcionamento, a origem da geração desses cards pode ser tanto de testes internos, quanto de validações de testadores que obtiveram alguma das builds do projeto e reportaram alguma inconsistência. Essas inconsistências foram tratadas por meio de cards como correções (*fixes*) no ambiente Jira.

A Tabela 7 a seguir apresenta uma relação dos 43 apontamentos dessa categoria que foram documentados, desse total, 42 foram resolvidos e 1 ainda está no *Backlog*. Isso demonstra a evolução contínua do sistema em busca de estabilidade e qualidade do produto final.

Tabela 7 – Registro de Correções (Fixes) (Ordenado por Data de Entrega)

Chave	Resumo	Status	Entregue
CEC-61	Fix: Feedback para o usuário quando um registro offline é enviado	Concluído	2025-01-10
CEC-67	Fix: Ajustes de condicionais de tela de avaliar registro	Concluído	2025-01-11
CEC-65	Fix: Mensagem de não adição de imagem	Concluído	2025-01-11
CEC-69	Fix: Ajuste de envio da imagem independente do input escolhido	Concluído	2025-01-11
CEC-63	Fix: Adicionar um ícone X no input de dropdown para deletar o conteúdo	Concluído	2025-01-11

(Continua na próxima página)

Tabela 7 – (Continuação) Registro de Correções (Fixes) (Ordenado por Data de Entrega)

Chave	Resumo	Status	Entregue
CEC-36	Fix: Mensagem de dicas para fotografia	Concluído	2025-01-11
CEC-71	Fix: Deletar foto de perfil quando conta é deletada	Concluído	2025-01-11
CEC-62	Fix: Ajustar visualização de filtros para ser mais intuitivo	Concluído	2025-01-11
CEC-56	Fix: Ajuste do header registros pendentes + meus registros	Concluído	2025-01-17
CEC-77	Fix: Ajuste dos cards de meus registros (responsividade)	Concluído	2025-01-23
CEC-92	Fix: Ajuste de italic na apresentação da taxonomia	Concluído	2025-02-22
CEC-95	Fix: Alterar a quantidade de caracteres do campo obs para 600 + travar escrita	Concluído	2025-02-22
CEC-83	Fix: Ajuste no regex de cadastro do campo de nome popular	Concluído	2025-02-22
CEC-94	Fix: Ajuste de campos opcionais que estao sendo apresentados como obrigatorios no envio da avaliacao do registro	Concluído	2025-02-22
CEC-88	Fix: A partir da localizacao do registro adicionar o campo nome da cidade	Concluído	2025-02-23
CEC-90	Fix: Na tela de registro simples, permitir que o usuario adicione a cidade e a guarita	Concluído	2025-02-23
CEC-86	Fix: Alteração do gráfico em pontos por gráfico em barras tela de painel de registros	Concluído	2025-02-23
CEC-99	Fix: Campo de obs nao está sendo mantido	Concluído	2025-03-21
CEC-102	Fix: Botão voltar tela de registros avaliados	Concluído	2025-03-23
CEC-100	Fix: Loading da image do bottomsheet mapa	Concluído	2025-03-24
CEC-117	Fix: Limpar dados de guarita municipio	Concluído	2025-04-01

(Continua na próxima página)

Tabela 7 – (Continuação) Registro de Correções (Fixes) (Ordenado por Data de Entrega)

Chave	Resumo	Status	Entregue
CEC-119	Fix: Ajustar aparecimento dos botoes de avaliar registro + adicionar pesquisador	Concluído	2025-04-05
CEC-121	Fix: Validação de liberar envio de registros	Concluído	2025-04-06
CEC-101	Fix: Loading entrar com Google ausente	Concluído	2025-04-06
CEC-127	Fix: Ajuste na visualização do retorno do profissional na tela de register view	Concluído	2025-04-17
CEC-126	Fix: Ajuste de estado inicia do seletor deve avaliar registro	Concluído	2025-04-17
CEC-128	Fix: Classe no model de AnimalResponse	Concluído	2025-04-18
CEC-131	Fix: Overflow do nome na tela de perfil	Concluído	2025-04-18
CEC-132	Fix: Problema no envio de registro offline sem localização	Concluído	2025-05-06
CEC-133	Fix: Validação dos campos registros simples	Concluído	2025-05-06
CEC-134	Fix: Ajuste de mostrar registros de guaridas no mapa	Concluído	2025-05-06
CEC-136	Fix: Formatação de coordenadas quando randomizador de guarita é acionado	Concluído	2025-05-06
CEC-43	Fix: Arrumar redirect após o cadastro	Concluído	2025-05-12
CEC-50	Fix: Alteração semântica do campo de gênero	Concluído	2025-05-12
CEC-51	Fix: Ajustes propostos pela professora Karen	Concluído	2025-05-12
CEC-52	Fix: Comportamento do dropdown dos inputs de pesquisa	Concluído	2025-05-12
CEC-68	Fix: Ajustes de condicionais da tela de visualizar registro	Concluído	2025-05-12
CEC-75	Fix: Adicionar lógica de contador de animais encontrados após avaliações	Concluído	2025-05-12

(Continua na próxima página)

Tabela 7 – (Continuação) Registro de Correções (Fixes) (Ordenado por Data de Entrega)

Chave	Resumo	Status	Entregue
CEC-79	Fix: Lógica para permitir acesso a features offline HomePage	Concluído	2025-05-12
CEC-130	Fix: Ajuste de bloqueio de switch quando gps esta desligado	Concluído	2025-05-12
CEC-82	Fix: Ajuste no datePicker register pannel	Concluído	2025-05-12
CEC-85	Fix: Ajuste no tamanho do campo de ob-servacao e retorno do profissional	Concluído	2025-05-12
CEC-80	Fix: Enviar e manipular o switch permite duplicar o registro	Concluído	2025-05-12
CEC-104	Fix: Campo de busca quando apagado deve apresentar todas as opcoes novamente	Backlog	

5.4.3 Débitos Técnicos e Melhorias

Além das funcionalidades previstas e dos *bugs* encontrados, também foram abertos *cards* para apontamentos de melhorias para algumas funcionalidades e respostas do sistema e débitos técnicos produzidos durante a codificação. Esses pontos surgiram durante a implementação de funcionalidades, revisões de código ou testes exploratórios e regressivos.

A Tabela 8 apresenta as melhorias e ajustes técnicos catalogados, evidenciando práticas de refatoração e aprimoramento contínuo. Ao todo foram 19 *cards* abertos com esse intuito, dos quais 8 foram resolvidos e 11 ainda estão no *Backlog*.

Tabela 8 – Registro de Débitos Técnicos e Melhorias (Ordenado por Data de Entrega)

Chave	Resumo	Status	Entregue
CEC-74	Melhoria: Adicionar um marcador nos registros que ainda não foram visualizados na tela de meus registros	Concluído	2024-11-23

(Continua na próxima página)

Tabela 8 – (Continuação) Registro de Débitos Técnicos e Melhorias (Ordenado por Data de Entrega)

Chave	Resumo	Status	Entregue
CEC-73	Melhoria: Poder abrir a imagem na tela de visualizar registro	Concluído	2025-01-11
CEC-70	Melhoria: Melhorar carregamento das imagens nas telas que apresentam todos os registros	Concluído	2025-01-11
CEC-123	Melhoria: Alterar nome de salvamento das imagens dos registros	Concluído	2025-04-07
CEC-129	Melhoria Adicionar loading no botao de avaliar registro	Concluído	2025-04-18
CEC-87	Melhoria: Captar localizacao de uma imagem da galeria quando presente	Concluído	2025-05-12
CEC-32	Melhoria: Ajustar centralização da lista de badges na tela de perfil	Concluído	2025-05-12
CEC-96	Melhoria: Adicionar switch para envio de imagens da galeria tela de registro técnico	Concluído	2025-05-12
CEC-93	Melhoria: Adicionar pré cadastro dos municípios do litoral norte, apenas seleção	Concluído	2025-05-12
CEC-45	Melhoria: Modificar logica de tratamento de exceptions para remocao da passagem de BuildContext	Backlog	
CEC-124	Melhoria: Notificação quando registro é avaliado	Backlog	
CEC-122	Melhoria: Exception de email nao encontrado	Backlog	
CEC-137	Melhoria: Lógica de retry quando a internet está ruim	Backlog	
CEC-91	Melhoria: Glossário ilustrado dos principais animais	Backlog	

(Continua na próxima página)

Tabela 8 – (Continuação) Registro de Débitos Técnicos e Melhorias (Ordenado por Data de Entrega)

Chave	Resumo	Status	Entregue
CEC-57	Melhoria: Validação de email cadastrado esqueci minha senha	Backlog	
CEC-120	Melhoria: Criar uma thumb com a imagem dos registros no mapa do painel	Backlog	
CEC-26	Melhoria: Refac da navegação	Backlog	
CEC-97	Fix: Alteracao da ordem dos campos do formulário de avaliar registro	Backlog	
CEC-72	Fix: Ajustar lógica de visualização da foto de perfil	Backlog	
CEC-103	Fix: Pop up informando saída do app no botao de fauna local	Backlog	

5.5 IMPLEMENTAÇÃO DO SISTEMA

Nesta seção, iremos abordar a codificação do aplicativo e apresentar o produto final gerado após as entrevistas, desenvolvimento, ajustes de funcionamento e correções de *bugs*.

O aplicativo FaunaMar foi desenvolvido utilizando uma arquitetura modular baseada no padrão *MVC* (Model-View-Controller), que promove a separação clara entre as responsabilidades da aplicação. A camada Model sendo responsável pelos dados e pela comunicação com o banco, enquanto a View, trata da interface e interação com o usuário e a Controller gerencia a lógica de negócio e coordena a comunicação entre as outras duas. Além disso, foram desenvolvidos widgets personalizados reutilizáveis, aplicados em diversos pontos da interface, com o objetivo de padronizar elementos visuais e reduzir a duplicação de código facilitando alterações na interface. Essa estrutura segue princípios fundamentais da engenharia de software, como coesão, baixo acoplamento e reutilização de componentes, o que contribui para a escalabilidade do sistema e a manutenção do código.

As funcionalidades do sistema foram construídas de maneira independente, permitindo que as alterações e melhorias possam ser implementadas de forma isolada, sem impactar outras partes do sistema. Existem dois tipos de usuários no sistema, os usuários comuns, chamados de cientistas cidadãos, e os pesquisadores, que são os profissionais que validam os dados enviados.

As principais seções do aplicativo incluem:

- **Login/Cadastro:** autenticação e criação de contas de usuários;
- **Menu Inicial:** tela de acesso rápido às funcionalidades principais;
- **Envio de Registros:** seção destinada ao cadastro de novos avistamentos de fauna, é possível enviar dois tipos de registros, um simples e outro técnico;
- **Painel de Registros:** seção que centraliza informações sobre os registros enviados, permitindo visualização, filtragem e exportação de dados;
- **Meus Registros:** seção onde o usuário pode acompanhar e gerenciar os registros enviados por ele;
- **Avaliar Registro:** seção exclusiva para pesquisadores, para análise e validação de dados enviados por todos os usuários;
- **Fauna Local:** catálogo externo de espécies da fauna marinha local;
- **Meu Perfil:** seção de visualização e edição de informações pessoais do usuário;
- **Adicionar Pesquisador:** seção exclusiva para pesquisadores, permitindo a inclusão de novos usuários com perfil de pesquisador;
- **Sobre o app:** informações sobre a versão e créditos do projeto.

A seguir, cada uma dessas seções será apresentada com suas respectivas telas, fluxos de interação, decisões de projeto e aspectos técnicos sobre o funcionamento do sistema.

5.5.1 Splashscreen

Foi gerada uma *splashscreen* para o aplicativo, essa tela traz o logo do projeto Fauna Marinha RS e é apresentada durante o carregamento completo do aplicativo para que o usuário tenha um feedback visual de que o aplicativo está sendo iniciado (Figura 30).

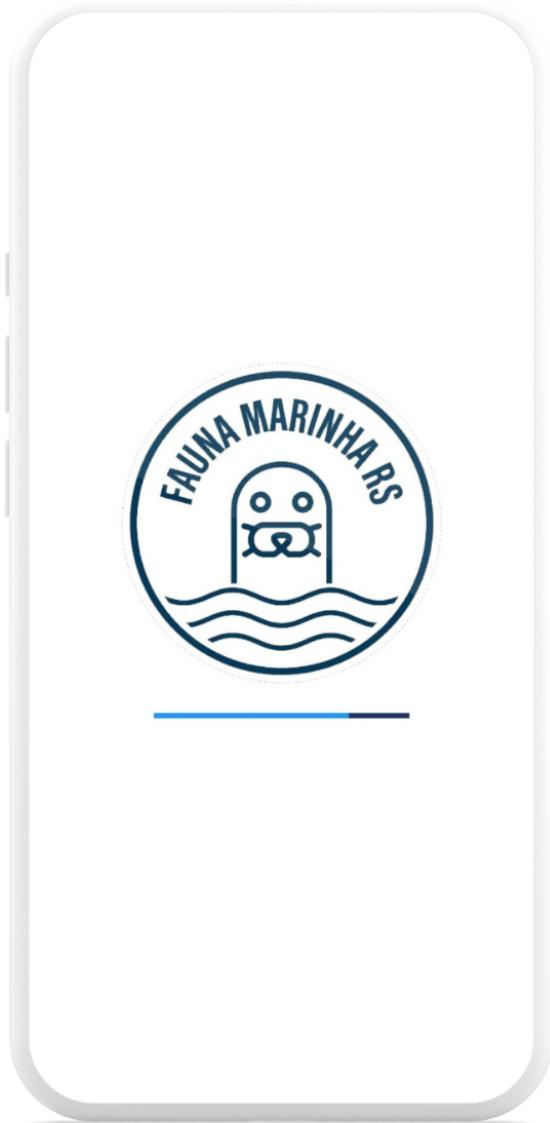


Figura 30 – *Splashscreen* do aplicativo.

Fonte: Autor

5.5.2 Login e Cadastro

Após a *splashscreen*, caso o usuário não tenha feito login previamente, ele é direcionado para a tela de login. Esta tela apresenta as seguintes opções: login com e-mail e senha, login via conta Google, realizar cadastro e recuperar senha (Figura 32).

A Figura 31 apresenta o diagrama de fluxo que resume o processo de autenticação do aplicativo, abrangendo os principais caminhos possíveis, como login com diferentes provedores, recuperação de senha e cadastro de novos usuários. Esse diagrama auxilia na compreensão geral da navegação inicial e da lógica de acesso implementada no sistema.

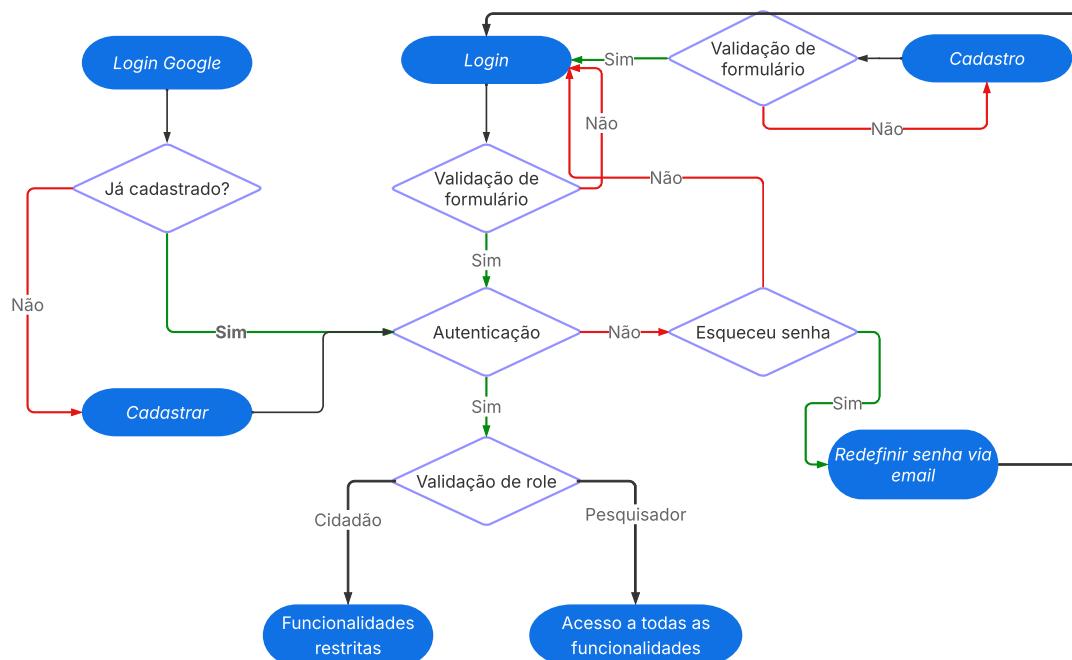


Figura 31 – Diagrama de fluxo do processo de autenticação e cadastro no aplicativo.

Fonte: Autor

Em ambos os provedores de login, o processo de autenticação é realizado por meio do Firebase Authentication, para garantir a segurança e a privacidade dos dados do usuário. A autenticação via Google é implementada utilizando o pacote `google_sign_in`, que permite integração com a conta Google do usuário, facilitando o acesso ao aplicativo.

O formulário de login possui validações para garantir que o usuário informe um e-mail em formato válido (com "@"e ".com"), e uma senha que atenda o padrão esperado (Figura 33).

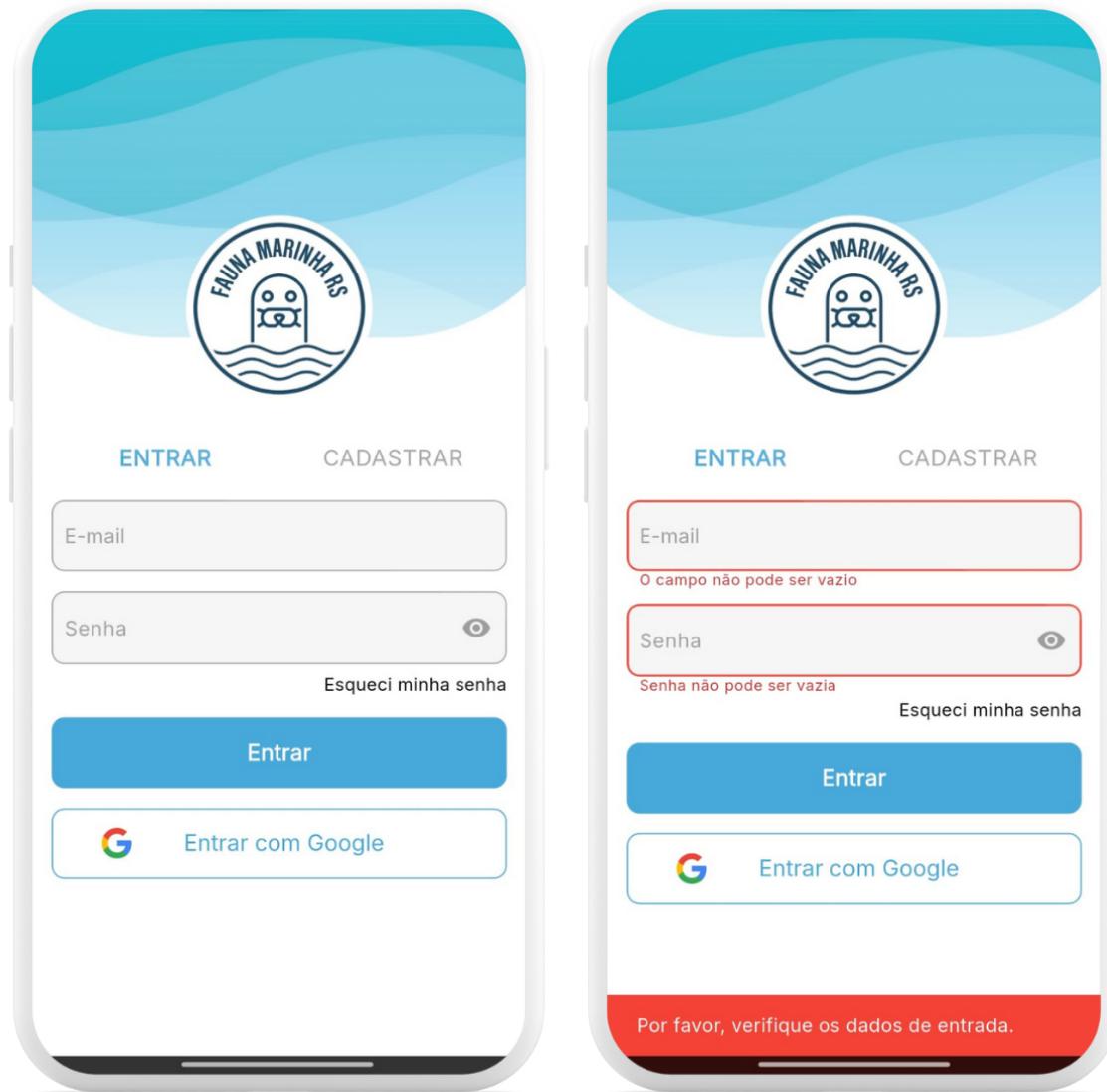


Figura 32 – Tela de login com formulário de entrada.

Figura 33 – Exemplo de erro de validação do formulário de login.

Fonte: Autor

O Firebase Authentication utiliza uma versão internamente modificada do algo-

ritmo *Scrypt* para realizar o *hash* das senhas dos usuários (Firebase Open Source, 2023). Durante o processo de registro, ao criar uma senha, o sistema aplica uma função de *hash* criptográfica à senha fornecida, gerando uma sequência de caracteres única, conhecida como *hash*. Segundo Serafim (2012), “o *hash* é irreversível. Não há informação suficiente no valor do *hash* h para recuperar a mensagem m”, o que assegura que a senha original não possa ser obtida a partir do valor armazenado. Essa abordagem está em conformidade com a Lei Geral de Proteção de Dados (LGPD), que exige que dados pessoais, como senhas, sejam tratados de forma segura e responsável (Brasil, 2018).

Apenas o *hash* gerado é armazenado no banco de dados, vinculado ao identificador do usuário. No momento do login, a senha fornecida é novamente processada pela mesma função de *hash* e comparada ao valor armazenado. Caso os dois hashes coincidam, a autenticação é concluída com sucesso, caso contrário, o acesso é negado.

Para evitar duplicidade de contas e oferecer maior flexibilidade ao usuário, optou-se pela utilização da funcionalidade de vinculação de contas do Firebase. Essa funcionalidade permite que, quando um usuário tenta se autenticar com um provedor (por exemplo, e-mail e senha) usando um endereço de e-mail que já está associado a outro provedor (como o Google Sign-In), o Firebase automaticamente une esses métodos de login em uma única chave de usuário.

Caso o usuário tenha esquecido a senha, ele pode solicitar a recuperação por meio do e-mail cadastrado. Para isso, o usuário deve clicar em "Esqueci minha senha" na tela de login, que irá realizar o redirecionamento para a tela de recuperação de senha (Figura 34). Após a inserção e validação do e-mail, o aplicativo envia um e-mail de recuperação com um link seguro para redefinir a senha (Figuras 35 e 36). Todo esse processo é realizado utilizando métodos do Firebase Authentication, que permite gerenciar o envio do e-mail e realiza o processo de substituição da senha antiga pela nova.

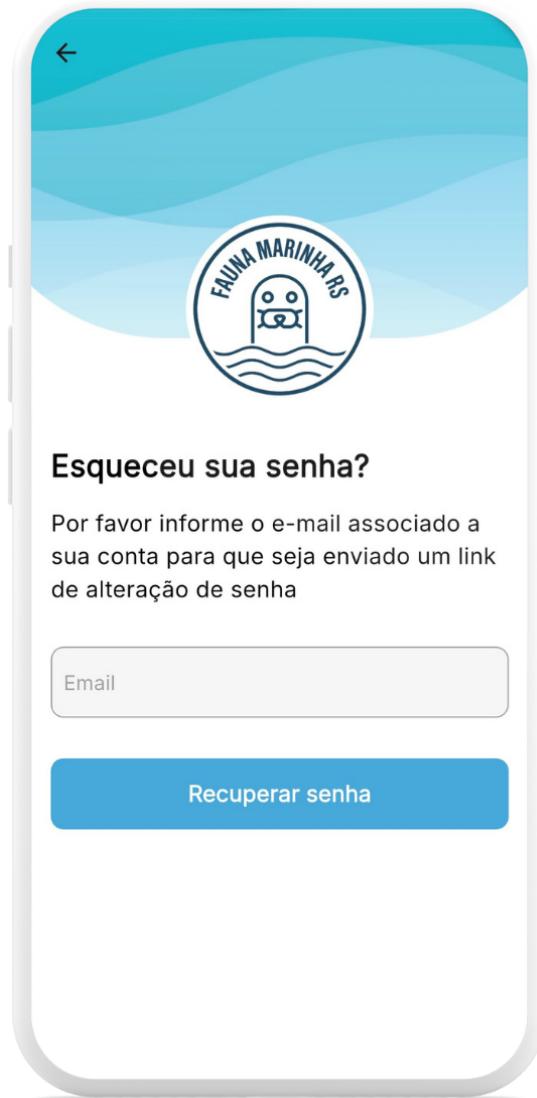


Figura 34 – Tela de recuperação de senha.

Fonte: Autor

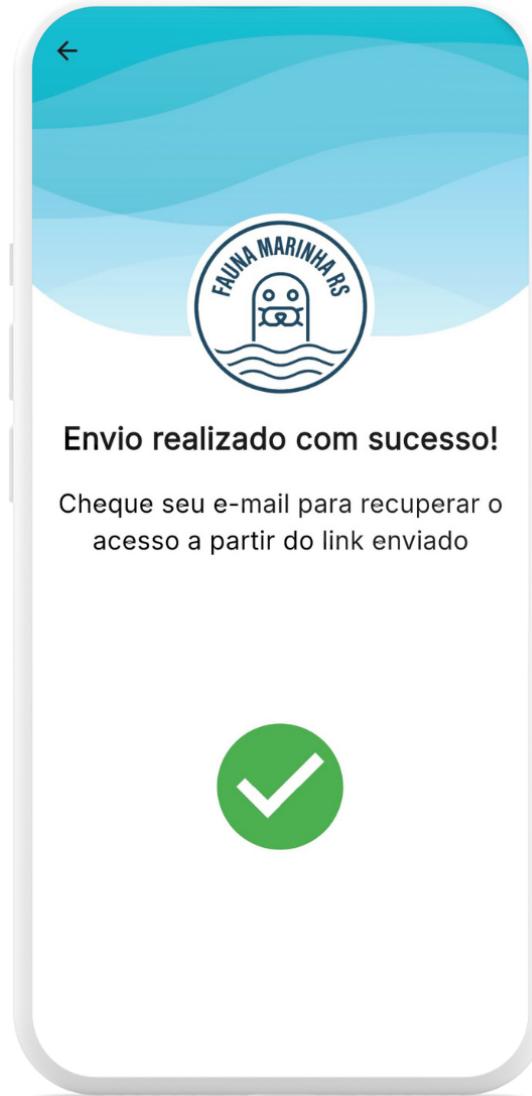


Figura 35 – Tela de recuperação de senha após e-mail enviado.

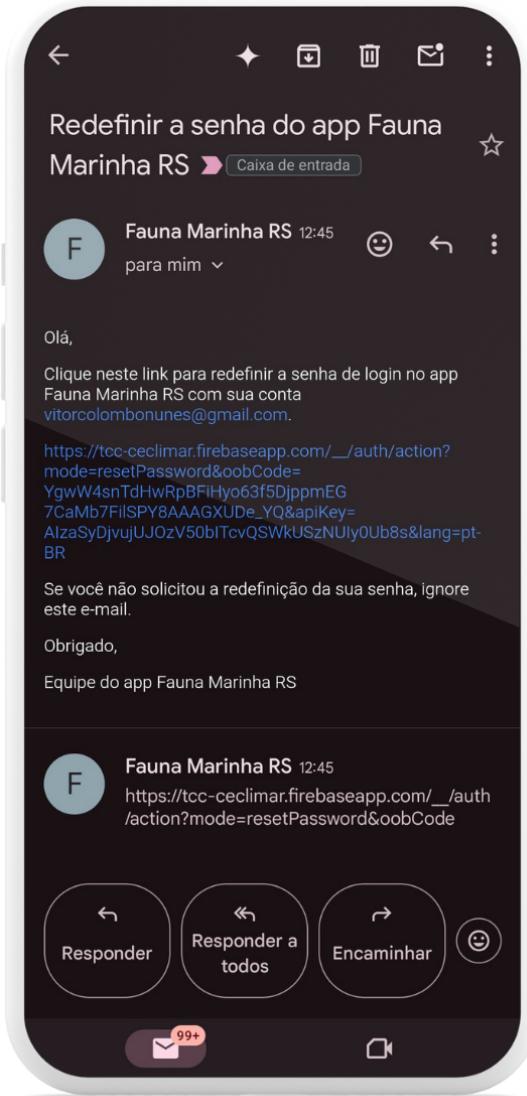


Figura 36 – E-mail personalizado de recuperação de senha.

Fonte: Autor

Para usuários que ainda não possuem uma conta, a tela de cadastro (Figura 37) oferece a opção de criar uma nova conta. O cadastro é realizado por meio de um formulário com validação (Figura 38) que exige apenas dados básicos necessários para o funcionamento do sistema, como nome, e-mail e senha, evitando a coleta excessiva de informações pessoais conforme as diretrizes da LGPD (Brasil, 2018). Todos os usuários

criados por essa tela serão registrados inicialmente como usuários comuns. A senha deve atender aos critérios de segurança, incluindo pelo menos um caractere especial, uma letra minúscula, uma letra maiúscula e deve possuir, no mínimo, 6 caracteres. Após o cadastro bem sucedido, o usuário é autenticado e redirecionado para o menu inicial da aplicação.

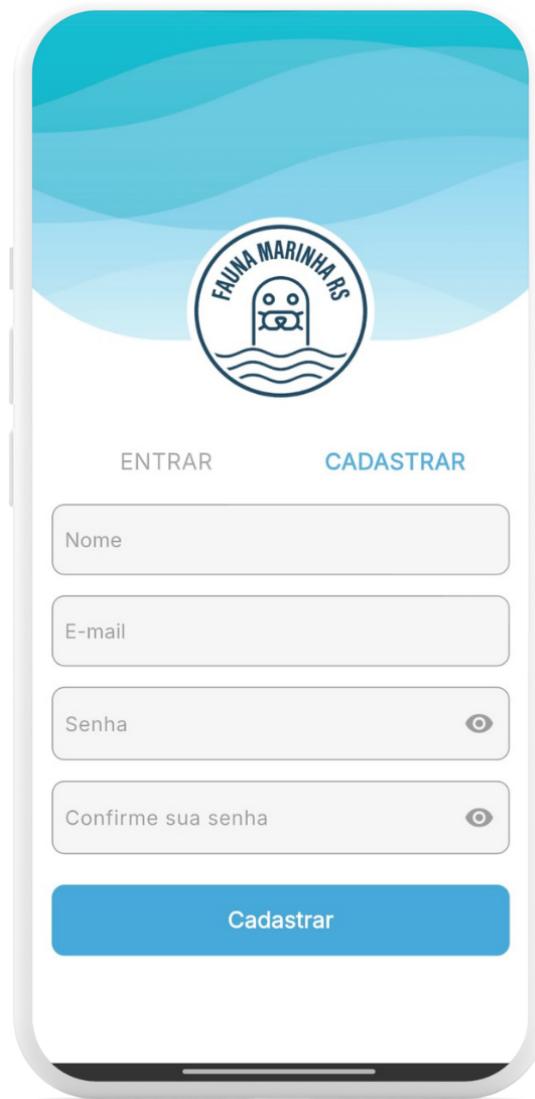


Figura 37 – Tela de cadastro de usuário com formulário simples.

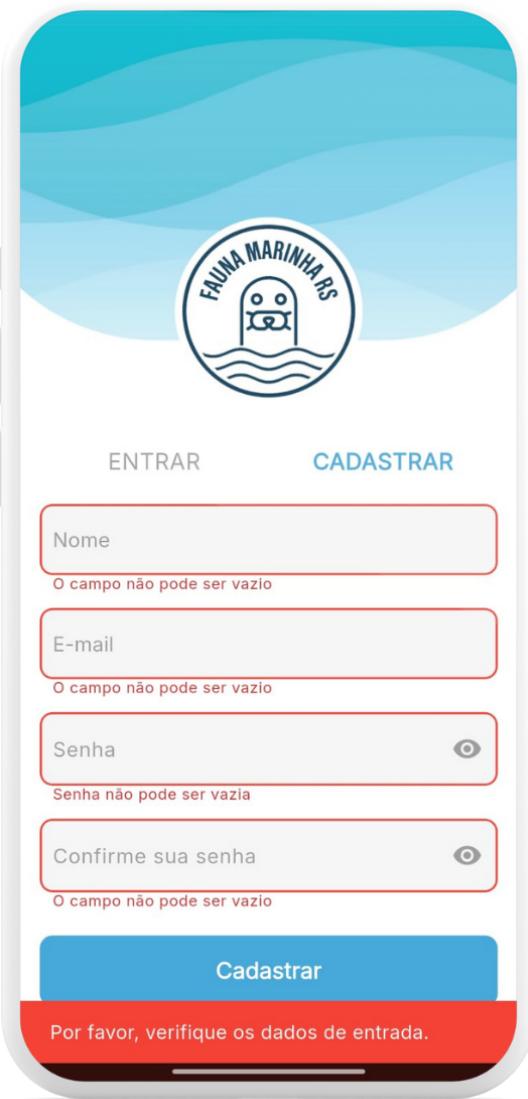


Figura 38 – Exemplo de mensagens de erro na validação do formulário de cadastro.

Fonte: Autor

5.5.3 Menu Inicial e Barra de Navegação

Com o usuário autenticado, o aplicativo exibe o menu inicial (Figura 39), que centraliza todas as funcionalidades do sistema em um *grid* 4x2. Nele estão dispostos *cards* padronizados com o nome e ícones representativos para cada uma das funcionalidades existentes para facilitar a navegação do usuário.

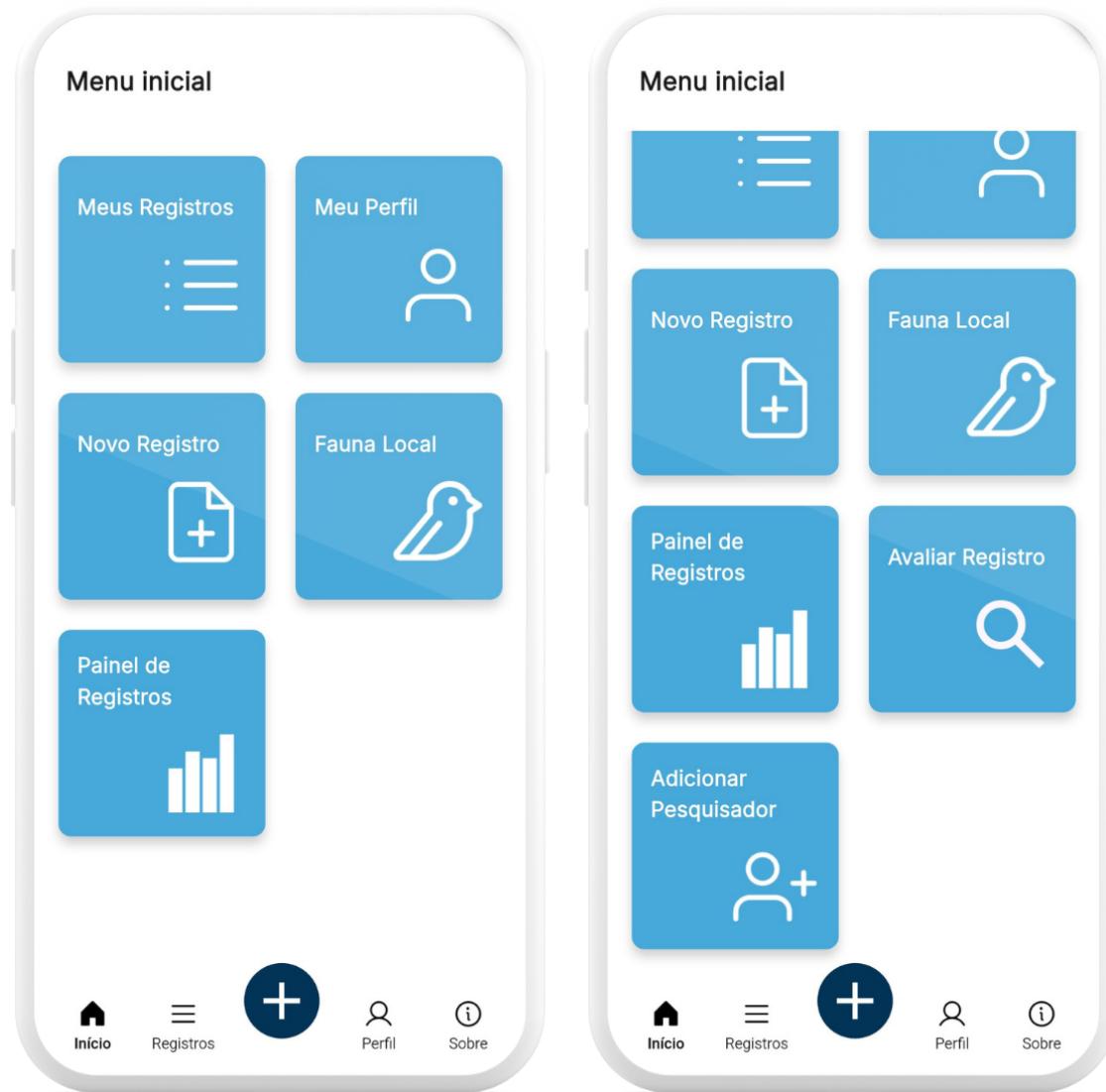


Figura 39 – Tela de Menu Inicial com funcionalidades liberadas para usuários comuns.

Figura 40 – Tela de Menu Inicial com funcionalidades exclusivas de pesquisadores.

Fonte: Autor

Este menu possui variações que dependem do tipo de usuário autenticado e da conexão do dispositivo com a internet. Se o usuário for um cientista cidadão, ele terá acesso apenas às funcionalidades básicas de "Meus Registros", "Meu Perfil", "Novo Registro", "Fauna Local" e "Painel de Registros".

Por outro lado, se o usuário for um pesquisador, ele terá acesso às funcionalidades adicionais de "Avaliar Registros" e "Adicionar Pesquisador" (Figura 40), porém essas funcionalidades só estarão disponíveis se o dispositivo estiver conectado à internet, pois dependem de acesso ao banco de dados remoto para funcionar corretamente. A Figura 41 apresenta o estado do menu inicial quando essas funcionalidades estão desabilitadas. Os *cards* das funcionalidades em questão dão lugar a um outro que informa que elas estão indisponíveis por falta de conexão com a internet.



Figura 41 – Tela de menu inicial com funcionalidades restritas para pesquisadores.

Fonte: Autor

O aplicativo apresenta também conta com uma barra de navegação inferior fixa que apenas não está presente nas telas de login e cadastro. Ela contém botões de acesso rápido com texto e ícones representativos de cada funcionalidade, permitindo que o usuário navegue facilmente entre as principais seções do aplicativo.

A barra é composta por cinco elementos principais. O primeiro botão, da esquerda

para a direita é o de "Início", representado por um ícone de casa, que redireciona o usuário ao "Menu Inicial" da aplicação. O segundo botão, "Registros", representado por um ícone de lista e leva à tela de "Meus registros". Centralizado na barra existe um botão flutuante com um ícone de "+", que permite que o usuário acesse rapidamente um formulário para envio de um registro. O quarto botão é o de "Perfil", identificado por um ícone de usuário ele redireciona para a tela de "Meu Perfil". E, por último, o botão "Sobre", indicado por um ícone de informação, redireciona para a tela de "Sobre o app".

É possível identificar em qual seção o usuário está no momento, pois o ícone se torna preenchido na coloração preta, enquanto os demais ícones permanecem no formato vazado (Figura 42).



Figura 42 – Barra de navegação fixa com o início selecionado.

Fonte: Autor

5.5.4 Meus Registros

A tela de "Meus Registros" é onde o usuário pode visualizar todos os registros enviados por ele, tanto os que foram validados quanto os que ainda estão pendentes de avaliação. Essa tela apresenta uma lista de registros que possui *scroll* vertical, essa lista é populada por registros ordenados por data de envio, do mais recente para o mais antigo, e são exibidos em blocos de informações que incluem o nome popular do animal, a cidade onde foi avistado, a data de envio e o status de validação e uma imagem representativa (Figura 43).

As informações dos registros, junto com suas imagens, são baixadas do banco de dados remoto, e guardadas em cache local para trazer uma melhor performance na navegação do usuário e permitir que ele possa visualizar os registros mesmo quando estiver sem conexão com a internet.

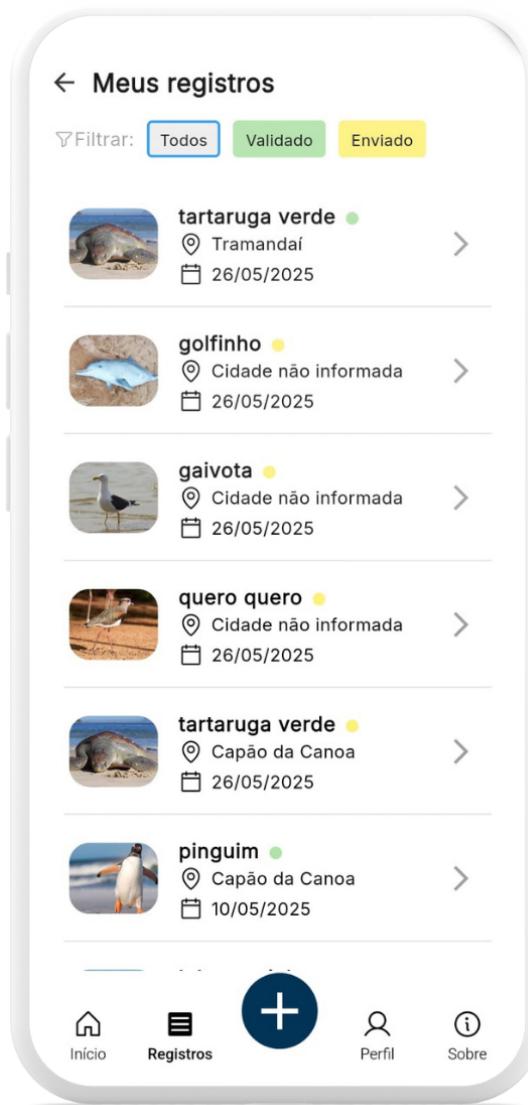


Figura 43 – Tela de meus registros apresentando os registros cadastrados pelo usuário em lista vertical.

Fonte: Autor

Nessa tela o usuário pode realizar a filtragem dos registros baseado nos seus status de validação, podendo escolher entre "Todos", "Validado" e "Enviado". Por padrão, a tela é carregada com o filtro de "Todos" selecionada e apresenta ambos os status. Porém, caso o usuário deseje, o filtro de "Validado" irá mostrar apenas registros já foram avaliados por um pesquisador e que possuem as informações conferidas, estes são representados com

um ponto verde ao lado do nome do animal (Figura 44). Enquanto o filtro de "Enviado" irá apenas mostrar registros que ainda não foram avaliados, são representados com um ponto amarelo ao lado do nome do animal (Figura 45).

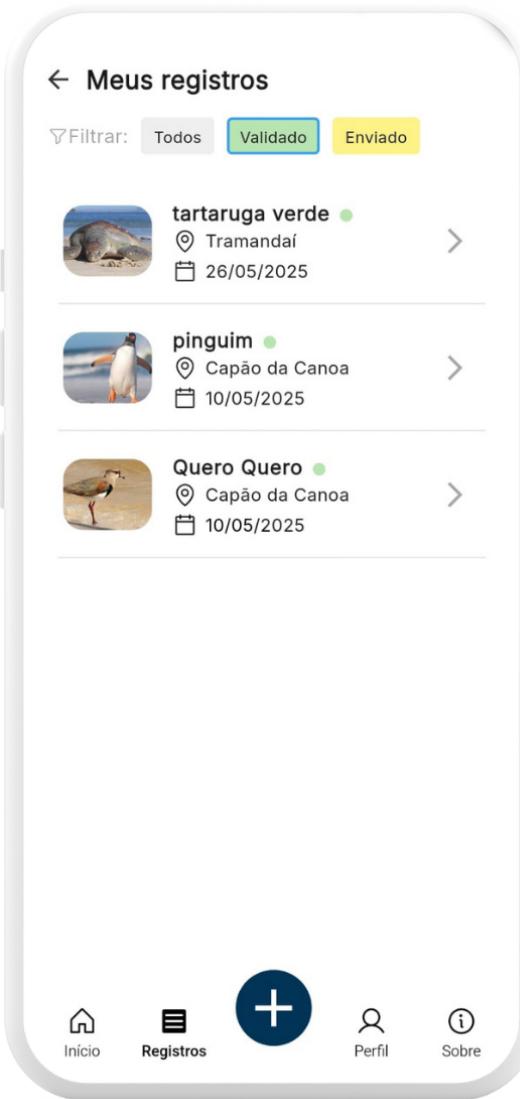


Figura 44 – Tela de Meus Registros com filtro de "Validado" aplicado.

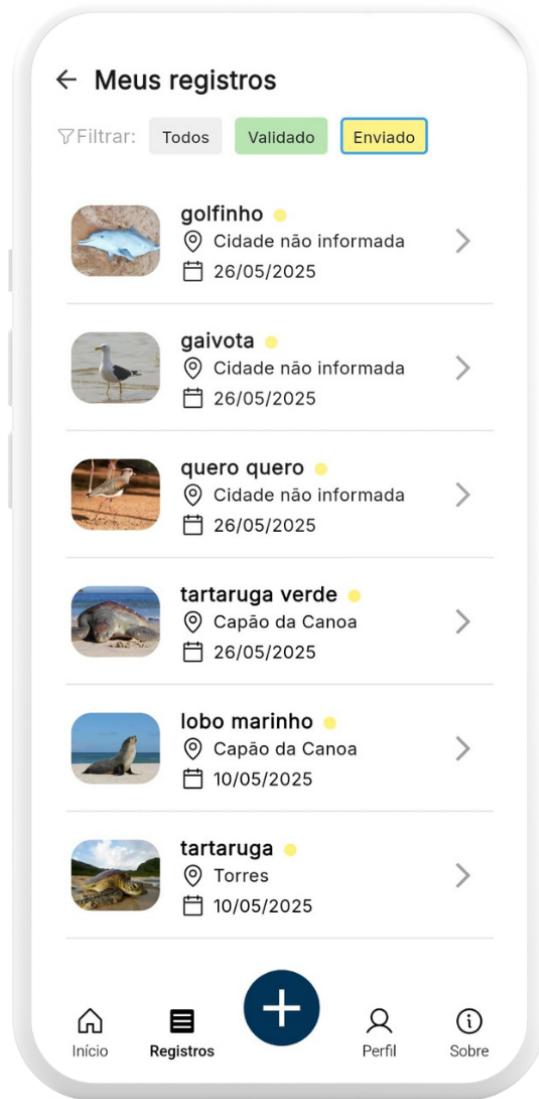


Figura 45 – Tela de Meus Registros com filtro de "Enviado" aplicado.

Fonte: Autor

5.5.5 Novo Registro

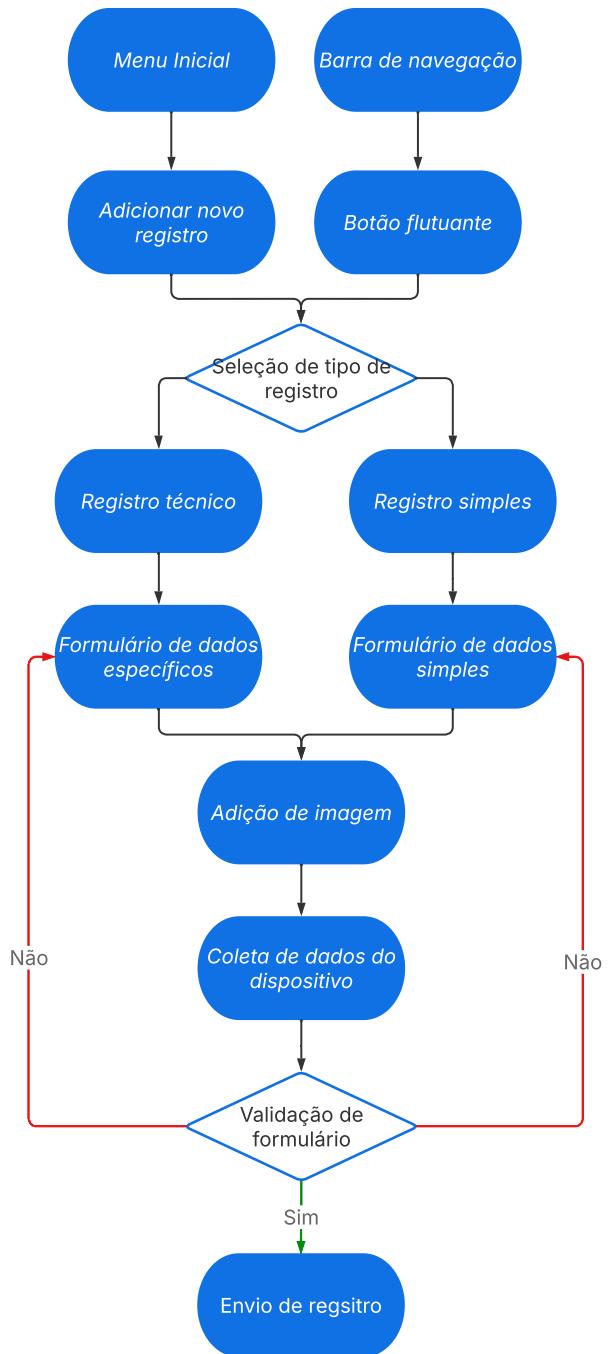


Figura 46 – Fluxograma de geração de novo registro.

Fonte: Autor

5.5.6 Avaliar Registro

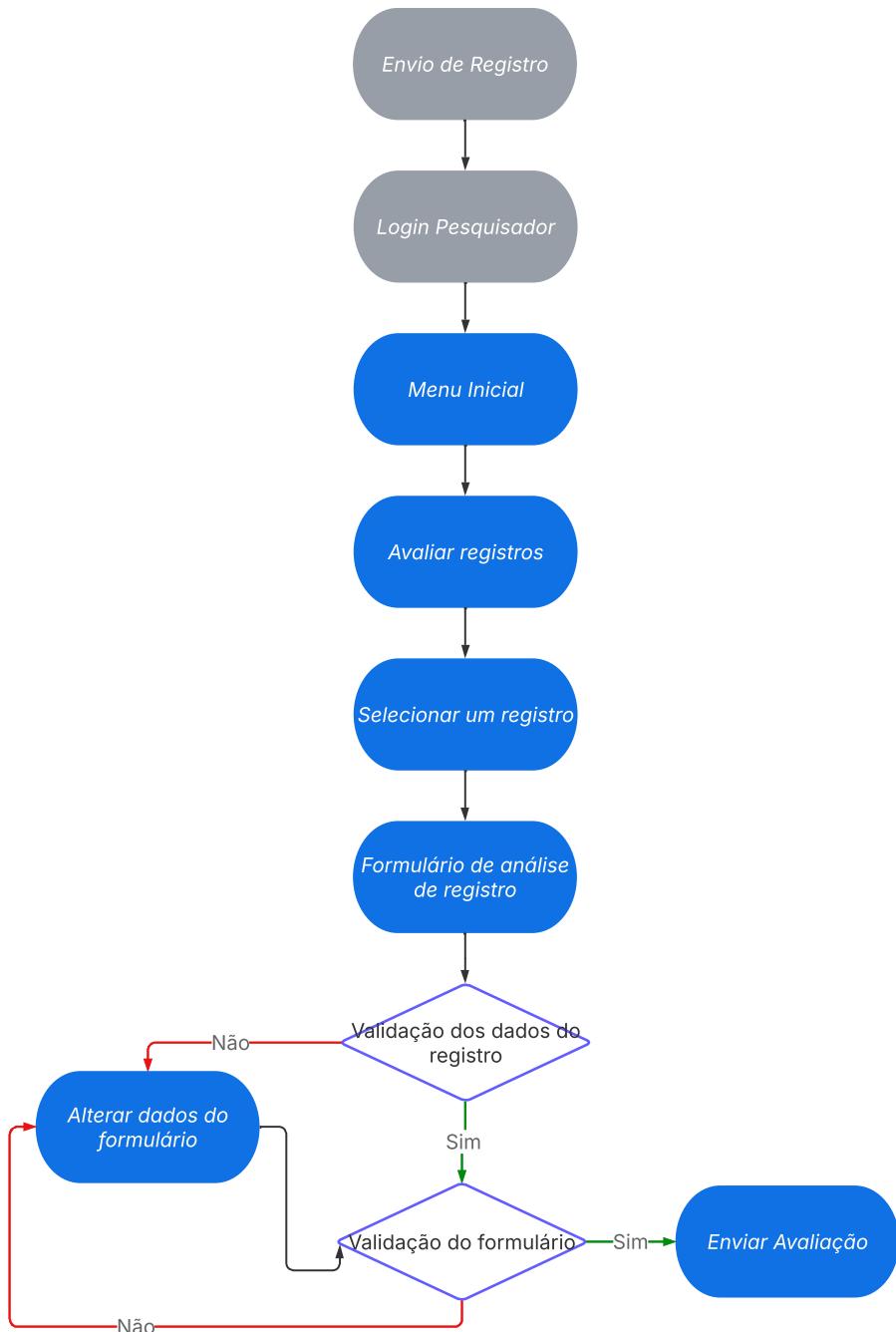


Figura 47 – Fluxograma de avaliação de registro.

Fonte: Autor

5.5.7 Painel de Registros

5.5.8 Adicionar Pesquisador

A funcionalidade de "Adicionar Pesquisador" é exclusiva para usuários com perfil de pesquisador e só está disponível quando o dispositivo está conectado à internet e permite que um pesquisador adicione novos usuários ao sistema com o perfil de pesquisador ou conceda permissões de pesquisador a usuários comuns já cadastrados. O fluxo dessa funcionalidade é apresentado na Figura 49.

A tela de "Adicionar Pesquisador" apresenta um formulário onde o pesquisador pode inserir o nome e o e-mail do usuário que deseja adicionar ou conceder a *role* de pesquisador (Figura 48).

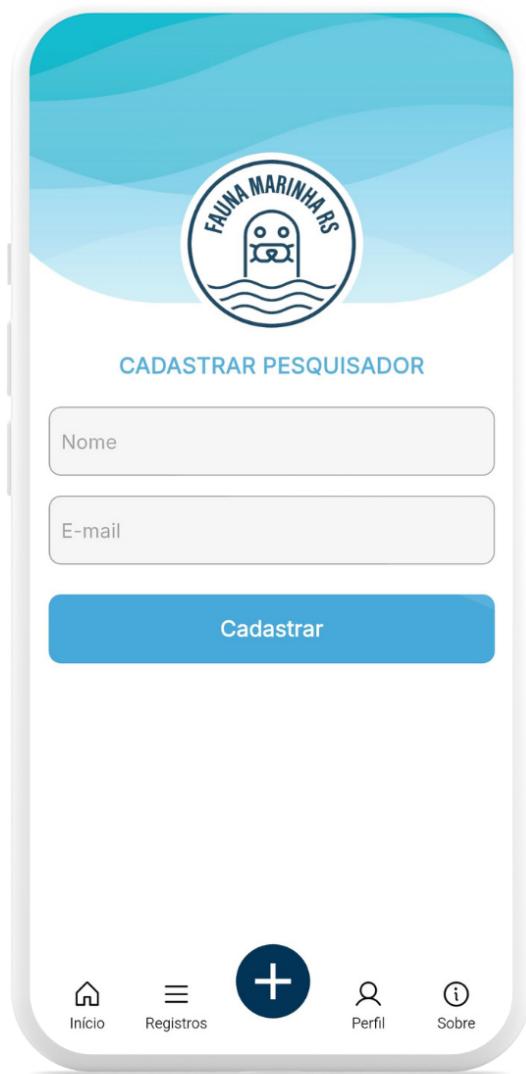


Figura 48 – Tela de Cadastrar Pesquisador.

Fonte: Autor

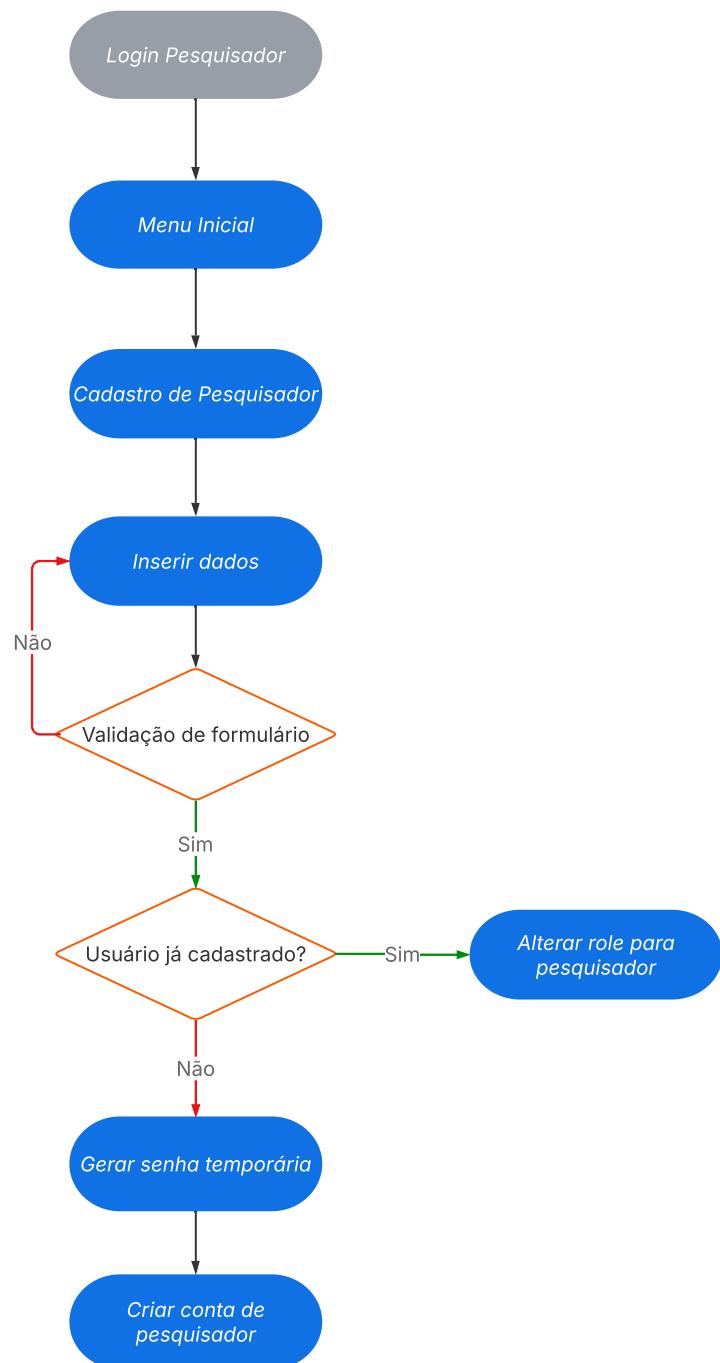


Figura 49 – Fluxograma de adição de pesquisador.

Fonte: Autor

Caso o usuário já esteja cadastrado no sistema, o sistema irá apresentar uma *bottomsheet* informando que o usuário já existe e perguntará se o usuário deseja realmente conceder a *role* de pesquisador para ele, caso o usuário aceite, o sistema irá atualizar as permissões do usuário e informar que a ação foi realizada com sucesso (Figura 51).

Caso o e-mail informado não esteja cadastrado, o sistema irá criar um novo usuário com as informações fornecidas e gerar uma senha temporária aleatória para ele, que será apresentada em tela com opção de copiar para a área de transferência (Figura 50). Essa senha temporária deve ser informada pelo novo usuário ao realizar o primeiro login, onde ele poderá alterar a senha para uma de sua escolha.



Figura 50 – Tela de Cadastrar Pesquisador com senha aleatória gerada e mensagem de sucesso.

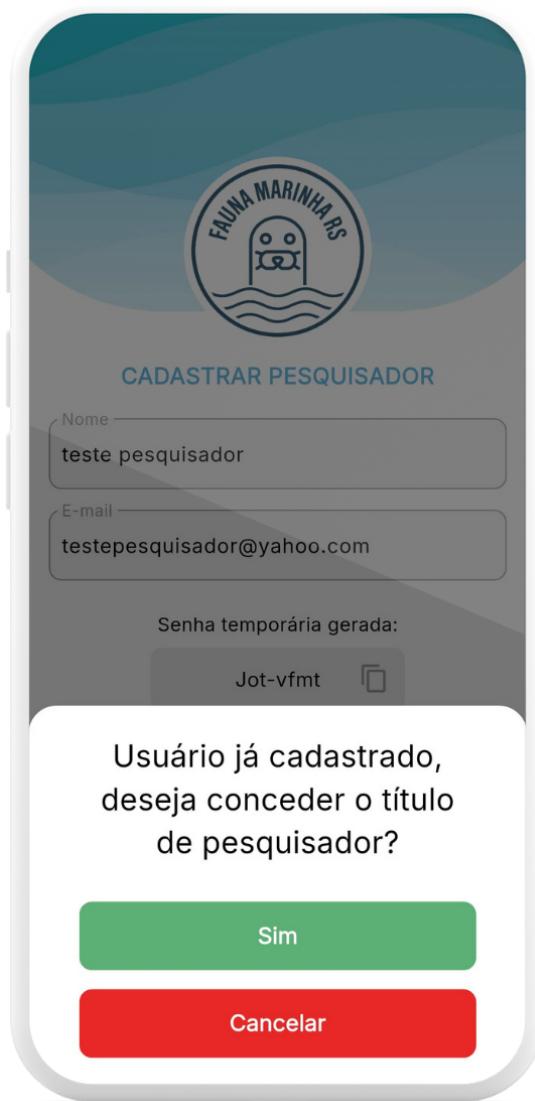


Figura 51 – *Bottomsheet* de confirmação para conceder *role* de pesquisador para o e-mail já cadastrado.

Fonte: Autor

5.5.9 Fauna Local

Para a seção de fauna local, foi decidido que, em um primeiro momento, seria utilizado o acervo externo do projeto Fauna Marinha RS, que já possui um catálogo bem

documentado de espécies registradas no litoral gaúcho, com material didático e imagens representativas de cada espécie. Sendo assim, o card de "Fauna Local" no menu inicial redireciona o usuário para o site do projeto para navegação externa (Figura 52).



Figura 52 – Tela da página de fauna local do projeto Fauna Marinha RS.

Fonte: Autor

5.5.10 Sobre o app

A tela de "Sobre o app" apresenta a versão da build do aplicativo e informações sobre o projeto Fauna Marinha RS, sobre o aplicativo e seu desenvolvimento separadas em *accordions*. Além disso, traz ícones com *links* externos para as redes sociais do projeto, como Instagram, Facebook, YouTube, Spotify, Telegram e para a caixa de e-mail, permitindo que os usuários acompanhem as atualizações e possam entrar em contato com a equipe do projeto (Figura 53).



Figura 53 – Tela de Sobre o app, apresentando *accordions* com informações do projeto e links para redes sociais.

Fonte: Autor

5.5.11 Armazenamento de Dados

A estrutura do banco de dados do aplicativo foi implementada no Firebase Firestore e organizada em coleções com responsabilidades individuais e índices de consulta específicos para melhor performance.

A coleção de *users* armazena os dados básicos dos usuários cadastrados. Cada documento dessa coleção representa um usuário e cada usuário possui atrelado a ele uma subcoleção com seus registros (9). Nessa coleção, foi utilizada uma técnica de desnormalização de dados, onde campos como *mammalsFound*, *reptilesFound* e *birdsFound* são armazenados diretamente no documento do usuário, para evitar a necessidade de uma consulta na subcoleção de registros para obter esses dados e, posteriormente, atualizá-los.

Essa abordagem foi vantajosa para a funcionalidade de Meu Perfil, onde o usuário pode visualizar rapidamente as estatísticas de sua conta com apenas uma consulta ao documento do usuário.

Tabela 9 – Campos da coleção *users*

Campo	Tipo	Descrição
<i>name</i>	string	Nome completo do usuário
<i>email</i>	string	E-mail do usuário
<i>profileImageUrl</i>	string	URL da imagem de perfil
<i>createdAt</i>	timestamp	Data e hora de criação do perfil
<i>role</i>	string	Papel do usuário no sistema
<i>birdsFound</i>	number	Número de registros com aves validados que foram enviados pelo usuário
<i>mammalsFound</i>	number	Número de registros com mamíferos validados que foram enviados pelo usuário
<i>reptilesFound</i>	number	Número de registros com répteis validados que foram enviados pelo usuário
<i>registers</i>	subcoleção	Coleção com os registros enviados pelo usuário

A subcoleção de *registers* se encontra dentro de cada documento de usuário e armazena todos os registros enviados por aquele usuário. Cada registro é armazenado

como um documento dentro dessa subcoleção, permitindo que cada usuário tenha seus registros organizados de forma individualizada (Tabela 10).

Essa abordagem foi adotada com o intuito de possibilitar consultas mais rápidas e eficientes quando o sistema precisar buscar por registros de um usuário específico, uma vez que o Firestore não precisará iterar sobre uma coleção global de registros, mas sim acessar diretamente a subcoleção pelo path `user/{id}/registers`, onde `{id}` é o ID do usuário. Essa medida também traz escalabilidade ao sistema, pois cada usuário terá sua própria subcoleção de registros, e as consultas não serão afetadas pelo número total de registros, mas sim pelo número de registros de cada usuário individualmente.

Uma inicial desvantagem dessa abordagem seria a necessidade de realizar consultas para obter todos os registros do sistema já que eles estão distribuídos em várias subcoleções. Porém, o Firestore oferece a funcionalidade de *collection group queries*, que permite a realização de consultas em todas as subcoleções de um tipo específico. Essa funcionalidade foi especialmente usada para implementação do painel de registros e avaliar registros, onde é necessário acessar todos os registros enviados por todos os usuários.

Outro ponto onde a desnormalização foi empregada para economizar o número de leituras e trazer mais rapidez nas consultas foi na adição dos campos de `authorName` e `user Id` em cada registro, que são preenchidos automaticamente com os dados do usuário que enviou o registro. Essa abordagem faz com que, ao listar registros para um usuário, não seja necessário realizar uma nova consulta no documento do usuário para cada item da lista.

Tabela 10 – Campos da subcoleção users/{id}/registers

Campo	Tipo	Descrição
registerNumber	string	Número identificador único do registro
animal	map	Dados taxonômicos do animal registrado
authorName	string	Nome do usuário que realizou o registro
userId	string	ID do usuário que realizou o registro (referência à coleção users)
date	string	Data e hora do registro no formato ISO
location.latitude	string	Latitude geográfica do local do registro
location.longitude	string	Longitude geográfica do local do registro
status	string	Status do registro
city	string	Cidade onde ocorreu o registro
beachSpot	string	Número da guarita
witnessed	boolean	Indica se o animal foi testemunhado encalhando
hour	string	Hora do encalhe
obs	string	Observações adicionais feitas pelo usuário
referencePoint	string	Ponto de referência
registerImageUrl	string	URL da imagem principal do animal
registerImageUrl2	string	URL de uma imagem adicional
sampleState	number	Estado de decomposição do animal
specialistReturn	string	Retorno de um especialista

Fonte: Autor

Além dos índices padrão criados pelo Firestore para os campos utilizados em filtros (*where*) e ordenações (*orderBy*), foram criados índices personalizados para a subcoleção registers, com o objetivo de garantir melhor desempenho em consultas específicas e operações de filtragem. Esses índices incluem:

- register.animal.species: para habilitar buscas por espécie de animal dentro da estrutura do objeto animal. Permitindo identificar todos os registros de uma determinada espécie.
- register.date: para habilitar buscas de registros ordenados por data em ordem crescente ou decrescente, auxiliando na exibição de listas de registros.

- `register.registerNumber`: para permitir buscas por número de registro, possibilitando uma consulta rápida de um registro específico.

A coleção de *animals* armazena informações sobre as espécies de animais registradas no sistema, incluindo dados taxonômicos e imagens representativas de cada espécie (Tabela 11). Essa coleção é utilizada para fornecer informações detalhadas sobre as espécies e para gerar uma padronização nas avaliações dos registros.

Os dados inicialmente inseridos nessa coleção são provenientes de materiais internos fornecidos pelo CECLIMAR e são utilizados no sistema como carga para os inputs de seleção dos campos taxonômicos nos formulários de envio e avaliação de registros. O sistema também permite que novos animais sejam adicionados à coleção por pesquisadores, caso uma espécie nova seja encontrada e não esteja presente na base de dados.

Tabela 11 – Campos da coleção *animals*

Campo	Tipo	Descrição
<code>id</code>	<code>number</code>	Identificador do animal
<code>popularName</code>	<code>string</code>	Nome popular da espécie
<code>scientificName</code>	<code>string</code>	Nome científico da espécie
<code>class</code>	<code>string</code>	Classe taxonômica
<code>order</code>	<code>string</code>	Ordem taxonômica
<code>family</code>	<code>string</code>	Família taxonômica
<code>genus</code>	<code>string</code>	Gênero taxonômico
<code>species</code>	<code>string</code>	Espécie taxonômica
<code>quantity</code>	<code>number</code>	Quantidade de vezes que a espécie foi identificada em registros
<code>imageUrl</code>	<code>string</code>	URL de imagem da espécie

A coleção de *counters* é utilizada para armazenar contadores gerais do sistema, como o número total de registros feitos e o número total de espécies cadastradas (Tabela 12).

Tabela 12 – Campos da coleção counters

Campo	Tipo	Descrição
registerCounter.count	number	Contador total de registros feitos no sistema
animalCounter.generalCounter	number	Contador total de espécies cadastradas

5.6 EXECUÇÃO DE TESTES E VERIFICAÇÃO DE QUALIDADE

CONSIDERAÇÕES FINAIS

O capítulo consolidou os resultados até aqui obtidos, alinhando-os ao planejamento inicial do projeto, garantindo rastreabilidade entre requisitos, projeto e implementação.

6 CONCLUSÃO

Sed consequat tellus et tortor. Ut tempor laoreet quam. Nullam id wisi a libero tristique semper. Nullam nisl massa, rutrum ut, egestas semper, mollis id, leo. Nulla ac massa eu risus blandit mattis. Mauris ut nunc. In hac habitasse platea dictumst. Aliquam eget tortor. Quisque dapibus pede in erat. Nunc enim. In dui nulla, commodo at, consecetur nec, malesuada nec, elit. Aliquam ornare tellus eu urna. Sed nec metus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Phasellus id magna. Duis malesuada interdum arcu. Integer metus. Morbi pulvinar pellentesque mi. Suspendisse sed est eu magna molestie egestas. Quisque mi lorem, pulvinar eget, egestas quis, luctus at, ante. Proin auctor vehicula purus. Fusce ac nisl aliquam ante hendrerit pellentesque. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Morbi wisi. Etiam arcu mauris, facilisis sed, eleifend non, nonummy ut, pede. Cras ut lacus tempor metus mollis placerat. Vivamus eu tortor vel metus interdum malesuada.

Sed eleifend, eros sit amet faucibus elementum, urna sapien consectetur mauris, quis egestas leo justo non risus. Morbi non felis ac libero vulputate fringilla. Mauris libero eros, lacinia non, sodales quis, dapibus porttitor, pede. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Morbi dapibus mauris condimentum nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam sit amet erat. Nulla varius. Etiam tincidunt dui vitae turpis. Donec leo. Morbi vulputate convallis est. Integer aliquet. Pellentesque aliquet sodales urna.

REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 6028: Resumo - apresentação*. Rio de Janeiro, 2003. 2 p. Citado na página 4.

Atlassian. *Jira | Software para acompanhamento de itens e projetos*. 2025. Acesso em 20 de maio de 2025. Disponível em: <<https://www.atlassian.com/br/software/jira>>. Citado 2 vezes nas páginas 18 e 35.

Brasil. *Lei n.º 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD)*. 2018. Acesso em: 7 de junho de 2025. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709compilado.htm>. Citado 2 vezes nas páginas 82 e 84.

CASTRO Érika P.; BAGER, A. Sistema urubu: a ciência cidadã em prol da conservação da biodiversidade. *Revista Brasileira de Tecnologias Sociais*, Editora UNIVALI, v. 6, n. 2, p. 111–130, 2019. Acesso em 08 de maio de 2024. Disponível em: <<https://periodicos.univali.br/index.php/rbts/article/view/15264>>. Citado na página 26.

CHAME, M. et al. Sistema de informação em saúde silvestre - “siss-geo”. *Grandes Desafios da Computação no Brasil*, Sociedade Brasileira de Computação, v. 3, 2015. Citado na página 24.

CHEN, L. *graphic: A Flutter data visualization library*. 2024. <<https://pub.dev/packages/graphic>>. MIT License. Accessed: 10 May 2025. Citado na página 42.

Dart. *Dart Documentation*. 2025. Acesso em 08 de maio de 2025. Disponível em: <<https://dart.dev/docs>>. Citado na página 37.

Facebook. *Facebook*. 2025. <<https://www.facebook.com/>>. Acesso em 10 de maio de 2025. Citado na página 44.

Fauna Marinha RS. *Fauna Marinha RS – Animais do litoral gaúcho*. 2025. Acesso em 12 de maio de 2025. Disponível em: <<https://www.ufrgs.br/faunamarinhars/>>. Citado na página 52.

Firebase Open Source. *Firebase Scrypt - Password hashing algorithm*. 2023. <<https://firebaseopensource.com/projects/firebase/scrypt>>. Acesso em: 8 jun. 2025. Citado na página 82.

Flutter. *Flutter Documentation*. 2025. Acesso em 08 de maio de 2025. Disponível em: <<https://docs.flutter.dev/>>. Citado na página 36.

FLUTTER_MAP. *flutter_map: Interactive maps for Flutter based on Leaflet*. 2025. <https://pub.dev/packages/flutter_map>. BSD 3-Clause License. Accessed: 10 May 2025. Citado na página 41.

GIL, A. C. *Como elaborar projetos de pesquisa*. 4. ed. São Paulo: Atlas, 2002. Disponível em: <https://files.cercomp.ufg.br/weby/up/150/o/Anexo_C1_como_elaborar_projeto_de_pesquisa_-_antonio_carlos_gil.pdf>. ISBN 85-224-3169-8. Disponível em: <https://files.cercomp.ufg.br/weby/up/150/o/Anexo_C1_como_elaborar_projeto_de_pesquisa_-_antonio_carlos_gil.pdf>. Citado na página 17.

Google. *Android*. 2024. Acesso em 15 de abril de 2024. Disponível em: <<https://www.android.com>>. Citado na página 23.

Google. *Google Drive*. 2024. <<https://www.google.com/drive/>>. Acesso em 07 de maio de 2025. Citado na página 15.

Google Firebase. *Firebase Documentation*. 2025. Acesso em 10 de maio de 2025. Disponível em: <<https://firebase.google.com/docs?authuser=0&hl=pt-br>>. Citado 2 vezes nas páginas 37 e 38.

Google Play Store. *Google Play Store*. 2024. Acesso em 10 de abril de 2024. Disponível em: <<https://play.google.com/store>>. Citado 2 vezes nas páginas 18 e 23.

Google Scholar. *Google Scholar*. 2025. Acesso em 20 de maio de 2025. Disponível em: <<https://scholar.google.com.br/>>. Citado na página 17.

Governo Federal do Brasil. *Gov.br*. 2024. Acesso em 18 de abril de 2024. Disponível em: <<https://www.gov.br>>. Citado na página 23.

GUEDES, G. *UML 2 - Uma Abordagem Prática - 3ª Edição*. Novatec Editora, 2018. ISBN 9788575226469. Disponível em: <<https://books.google.com.br/books?id=RUDLDwAAQBAJ>>. Citado na página 46.

Instagram. *Instagram*. 2025. <<https://www.instagram.com/>>. Acesso em 10 de maio de 2025. Citado na página 44.

LEIER, S.; CONTRIBUTORS. *Hive - Lightweight and blazing fast key-value database written in pure Dart*. 2025. <<https://pub.dev/documentation/hive/latest/>>. Acesso em 08 de maio de 2025. Citado na página 41.

MARTINS, D. G. d. M.; CABRAL, E. H. d. S. Panorama dos principais estudos sobre ciência cidadã. *ForScience*, v. 9, n. 2, p. e01030, out. 2021. Disponível em: <<https://forscience.ifmg.edu.br/index.php/forscience/article/view/1030>>. Citado na página 15.

NAÇÕES UNIDAS. *Agenda 2030 para o Desenvolvimento Sustentável*. 2015. <<https://brasil.un.org/pt-br/91863-agenda-2030-para-o-desenvolvimento-sustentável>>. Acesso em 10 de maio de 2024. Citado 2 vezes nas páginas 16 e 32.

NAÇÕES UNIDAS. *Sobre o nosso trabalho para alcançar os Objetivos de Desenvolvimento Sustentável no Brasil*. 2015. <<https://brasil.un.org/pt-br/sdgs>>. Acesso em 10 de maio de 2024. Citado 2 vezes nas páginas 32 e 34.

OHNO, T. *Toyota Production System: Beyond Large-Scale Production*. [S.l.]: Productivity Press, 1988. 176 p. Citado 2 vezes nas páginas 18 e 35.

PALMA, D. A. *Monitoramento de qualidade da água com o enfoque ciência cidadã: estudo de cem Brazilândia*. 76 p. Monografia — Universidade de Brasília, Brasília, 2016. Citado na página 31.

Petrobras. *SIMBA - Sistema de Monitoramento da Biodiversidade Aquática*. 2024. Acesso em 12 de abril de 2024. Disponível em: <<https://simba.petrobras.com.br/simba/web/>>. Citado 4 vezes nas páginas 27, 28, 29 e 30.

PRESSMAN, R. S. *Engenharia de Software – uma Abordagem Profissional*. 7. ed. [S.l.]: AMGH Editora LTDA, 2011. Citado 4 vezes nas páginas 18, 34, 35 e 51.

SERAFIM, V. *Introdução à Criptografia: Funções Criptográficas de Hash*. 2012. Acesso em 08 de junho de 2025. Disponível em: <http://www.serafim.eti.br/academia/recursos/Roteiro_08-Funcoes_de_Hash.pdf>. Citado na página 82.

SIBBR. *SIBBr: Sistema de Informação sobre a Biodiversidade Brasileira*. 2024. <<https://sibbr.gov.br/>>. Acesso em 08 de abril de 2024. Citado 2 vezes nas páginas 17 e 23.

SINGH, K. *excel: A Flutter and Dart library for reading and writing Excel files*. 2021. <<https://pub.dev/packages/excel>>. MIT License. Accessed: 10 May 2025. Citado na página 41.

SISS-Geo. *Plataforma Institucional Biodiversidade e Saúde Silvestre. Sistema de Informação em Saúde Silvestre*. 2024. <<https://sissgeo.lncc.br/apresentacao.xhtml>>. Acesso em 12 de abril de 2024. Citado 2 vezes nas páginas 24 e 25.

SOMMERVILLE, I. *Engenharia de Software*. 9. ed. São Paulo: Pearson Prentice Hall, 2011. Tradução da obra original: Software Engineering. Revisão técnica de Kechi Hirama. 3^a reimpressão – dez. 2013. ISBN 978-85-7936-108-1. Disponível em: <<https://www.facom.ufu.br/~william/Disciplinas%202018-2/BSI-GSI030-EngenhariaSoftware/Livro/engenhariaSoftwareSommerville.pdf>>. Citado na página 46.

WhatsApp. *WhatsApp*. 2024. <<https://www.whatsapp.com/>>. Acesso em 07 de maio de 2025. Citado 2 vezes nas páginas 15 e 44.

WILDSCHUT, D. The need for citizen science in the transition to a sustainable peer-to-peer-society. *Futures*, v. 91, p. 46–52, 2017. Citado na página 31.

ZAYAT, W.; SENVAR, O. Framework study for agile software development via scrum and kanban. *International Journal Of Innovation And Technology Management*, World Scientific Pub Co Pte Ltd, v. 17, n. 04, p. 203002–1–203002–24, 2020. Acesso em 07 de maio de 2025. Disponível em: <<https://www.worldscientific.com/doi/10.1142/S0219877020300025>>. Citado na página 35.

APÊNDICE A – QUISQUE LIBERO JUSTO

Quisque facilisis auctor sapien. Pellentesque gravida hendrerit lectus. Mauris rutrum sodales sapien. Fusce hendrerit sem vel lorem. Integer pellentesque massa vel augue. Integer elit tortor, feugiat quis, sagittis et, ornare non, lacus. Vestibulum posuere pellentesque eros. Quisque venenatis ipsum dictum nulla. Aliquam quis quam non metus eleifend interdum. Nam eget sapien ac mauris malesuada adipiscing. Etiam eleifend neque sed quam. Nulla facilisi. Proin a ligula. Sed id dui eu nibh egestas tincidunt. Suspendisse arcu.

**APÊNDICE B – NULLAM ELEMENTUM URNA VEL IMPERDIET
SODALES ELIT IPSUM PHARETRA LIGULA AC PRETIUM ANTE JUSTO A
NULLA CURABITUR TRISTIQUE ARCU EU METUS**

Nunc velit. Nullam elit sapien, eleifend eu, commodo nec, semper sit amet, elit. Nulla lectus risus, condimentum ut, laoreet eget, viverra nec, odio. Proin lobortis. Curabitur dictum arcu vel wisi. Cras id nulla venenatis tortor congue ultrices. Pellentesque eget pede. Sed eleifend sagittis elit. Nam sed tellus sit amet lectus ullamcorper tristique. Mauris enim sem, tristique eu, accumsan at, scelerisque vulputate, neque. Quisque lacus. Donec et ipsum sit amet elit nonummy aliquet. Sed viverra nisl at sem. Nam diam. Mauris ut dolor. Curabitur ornare tortor cursus velit.

Morbi tincidunt posuere arcu. Cras venenatis est vitae dolor. Vivamus scelerisque semper mi. Donec ipsum arcu, consequat scelerisque, viverra id, dictum at, metus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut pede sem, tempus ut, porttitor bibendum, molestie eu, elit. Suspendisse potenti. Sed id lectus sit amet purus faucibus vehicula. Praesent sed sem non dui pharetra interdum. Nam viverra ultrices magna.

Aenean laoreet aliquam orci. Nunc interdum elementum urna. Quisque erat. Nullam tempor neque. Maecenas velit nibh, scelerisque a, consequat ut, viverra in, enim. Duis magna. Donec odio neque, tristique et, tincidunt eu, rhoncus ac, nunc. Mauris malesuada malesuada elit. Etiam lacus mauris, pretium vel, blandit in, ultricies id, libero. Phasellus bibendum erat ut diam. In congue imperdiet lectus.

ANEXO A – MORBI ULTRICES RUTRUM LOREM.

Sed mattis, erat sit amet gravida malesuada, elit augue egestas diam, tempus scelerisque nunc nisl vitae libero. Sed consequat feugiat massa. Nunc porta, eros in eleifend varius, erat leo rutrum dui, non convallis lectus orci ut nibh. Sed lorem massa, nonummy quis, egestas id, condimentum at, nisl. Maecenas at nibh. Aliquam et augue at nunc pellentesque ullamcorper. Duis nisl nibh, laoreet suscipit, convallis ut, rutrum id, enim. Phasellus odio. Nulla nulla elit, molestie non, scelerisque at, vestibulum eu, nulla. Ut odio nisl, facilisis id, mollis et, scelerisque nec, enim. Aenean sem leo, pellentesque sit amet, scelerisque sit amet, vehicula pellentesque, sapien.

**ANEXO B – CRAS NON URNA SED FEUGIAT CUM SOCIIS NATOQUE
PENATIBUS ET MAGNIS DIS PARTURIENT MONTES NASCETUR
RIDICULUS MUS**

Sed consequat tellus et tortor. Ut tempor laoreet quam. Nullam id wisi a libero tristique semper. Nullam nisl massa, rutrum ut, egestas semper, mollis id, leo. Nulla ac massa eu risus blandit mattis. Mauris ut nunc. In hac habitasse platea dictumst. Aliquam eget tortor. Quisque dapibus pede in erat. Nunc enim. In dui nulla, commodo at, consectetur nec, malesuada nec, elit. Aliquam ornare tellus eu urna. Sed nec metus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

ANEXO C – FUSCE FACILISIS LACINIA DUI

Phasellus id magna. Duis malesuada interdum arcu. Integer metus. Morbi pulvinar pellentesque mi. Suspendisse sed est eu magna molestie egestas. Quisque mi lorem, pulvinar eget, egestas quis, luctus at, ante. Proin auctor vehicula purus. Fusce ac nisl aliquam ante hendrerit pellentesque. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Morbi wisi. Etiam arcu mauris, facilisis sed, eleifend non, nonummy ut, pede. Cras ut lacus tempor metus mollis placerat. Vivamus eu tortor vel metus interdum malesuada.