UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

THIAGO ROBERTO DA CONCEIÇÃO
Orientador: Saul Delabrida

TEMPUS: APLICATIVO MULTIPLATAFORMA DE REGISTRO E DOCUMENTAÇÃO DE AMBIENTES PARA AUXÍLIO EM ESTUDOS DE CAMPO E CIÊNCIA CIDADÃ

Ouro Preto, MG
2021

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

THIAGO ROBERTO DA CONCEIÇÃO

TEMPUS:
APLICATIVO MULTIPLATAFORMA DE REGISTRO E DOCUMENTAÇÃO DE AMBIENTES PARA AUXÍLIO EM ESTUDOS DE CAMPO E CIÊNCIA CIDADÃ

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Saul Delabrida

Ouro Preto, MG
2021

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E BIOLOGICAS
DEPARTAMENTO DE COMPUTACAO

FOLHA DE APROVAÇÃO

Thiago Roberto da Conceição

APLICATIVO MULTIPLATAFORMA DE REGISTRO E DOCUMENTAÇÃO DE AMBIENTES PARA AUXÍLIO EM ESTUDOS DE CAMPO E CIÊNCIA CIDADÃ

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal
de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 25 de Agosto de 2021.

Membros da banca

Saul Emanuel Delabrida Silva (Orientador) - Doutor - Universidade Federal de Ouro Preto
Joubert de Castro Lima (Examinador) - Doutor - Universidade Federal de Ouro Preto
Jadson Castro Gertrudes (Examinador) - Doutor - Universidade Federal de Ouro Preto

Saul Emanuel Delabrida Silva, Orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 25/08/2021.

Documento assinado eletronicamente por **Saul Emanuel Delabrida Silva**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 25/08/2021, às 16:43, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0210870** e o código CRC **BFF21D9A**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000
Telefone: 3135591692 - www.ufop.br

Dedico este trabalho à meus famíliares e amigos, que sempre me deram a certeza de que eu conseguiria chegar até aqui.

Agradecimentos

Agradeço primeiramente a Deus, que sempre esteve do meu lado mesmo quando eu não estive ao lado Dele.

Aos meus pais que sempre me incentivaram nos estudos e me deram condições de ter ensino de qualidade.

Aos meus amigos de infância João Gabriel e Pedro Henrique, pelos bons momentos e memórias criadas durante os anos.

Aos amigos Douglas, Evair, Mateus, Débora, Júlia e Lucas, que fiz durante a graduação, por todo companheirismo e ajuda durante todo o curso.

Aos meus irmãos da República Artigo Quinto, por toda amizade e amor compartilhado durante todo esse tempo.

Ao meu orientador Saul Delabrida, por ter aceitado em me orientar nesse último passo.

E finalmente, agradeço à UFOP, meu segundo lar, o lugar onde me encontrei, onde fiz as amizades que vou levar pra toda minha vida e onde me tornei um ser humano muito melhor.

Resumo

A documentação e registro dos ambientes são feitos com o objetivo de compreender o comportamento dos elementos que compõem o meio e a relação entre si. Os estudos de campo e a prática da ciência cidadã se favorecem de uma coleta de dados bem feita. Portanto, o objetivo deste trabalho foi propor o desenvolvimento de uma aplicação móvel que auxilie os usuários a documentarem ambientes, registrarem seus elementos e etiquetá-los de maneira adequada. A escassez de aplicativos similares e a incompletude dos aplicativos existentes indica que o trabalho tem espaço e valor. Primeiramente, foi criado um protótipo utilizando boas práticas de design e usabilidade. Para o desenvolvimento do aplicativo, foi utilizado a ferramenta Ionic, amplamente utilizada para desenvolver aplicativos multiplataformas (Android e iOS). Os testes automáticos foram feitos utilizando o *test-runner* Karma para código em Javascript, enquanto testes manuais usaram como base a escala *System Usability Scale* - SUS, que mede a qualidade da usabilidade da aplicação. A implementação foi feita seguindo fielmente o protótipo.

Palavras-chave: Desenvolvimento móvel. Aplicativos móveis. Ciência cidadã. Estudo de campo. Aplicativos multiplataforma.

Abstract

The documentation and registration of the environments are made to understand the elements that make up the environment and the relationship between them. Field studies and the practice of citizen science benefit from well-done data collection. Therefore, the objective of this work was to propose the development of a mobile application that helps users document environments, register their elements, and label them appropriately. The scarcity of similar applications and the lack of completeness of existing ones indicates that the work has indicates the work has space and value. First, a prototype was created using good design and usability practices. We developed the application with the Ionic tool, widely used to develop cross-platform applications (Android and iOS). Automatic tests were performed using the *test-runner* Karma for Javascript code, while manual tests were based on the *System Usability Scale* - SUS, which measures the quality of the application's usability. The implementation was done following the prototype faithfully.

Keywords: Mobile development. Mobile applications. Citizen science. Field studies. Crossplatform applications.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

API	*Application Programming Interface*
HTTP	*HyperText Transfer Protocol*
RAFT	*Remote Accessible Field Trips*

Sumário

1 Introdução

Estudos ou pesquisas de campo são investigações onde ocorrem a coleta de informações a partir da observação de eventos da forma como ocorrem na natureza (TUMELERO, 2018). Essas informações são posteriormente analisadas, classificadas e interpretadas, com o intuito de compreender as causas e efeitos do objeto de estudo.

Atualmente não existem técnicas predefinidas para a coleta dos dados em pesquisas de campo. O que pode acontecer é que, a partir das técnicas selecionadas, a pesquisa pode tomar um caráter quantitativo ou qualitativo (TUMELERO, 2018). Independente do tipo de pesquisa, vale ressaltar a importância dessas, por permitirem o levantamento e coleta de dados que não são encontrados através de estudos teóricos.

Esse tipo de estudo requer vários equipamentos necessários para ser realizado com qualidade e precisão. Estudos da área da biologia e gelogia, por exemplo, requerem que os participantes utilizem roupas adequadas para se adaptarem desde as condições do relevo do local de estudo até as condições climáticas, como sol muito forte ou chuvas volumosas e ventos. Além da vestimenta, há os equipamentos necessários para a realização do estudo. Pranchetas, papéis, canetas, bússolas, entre outros itens.

A prática da ciência cidadã, por sua vez, conta com usuários - cientistas ou não - que também coletam dados de interesse. A diferença para um estudo de campo é que os envolvidos na prática não precisam ser cientistas e nem adotar uma metodologia específica de pesquisa. O interesse se encontra na coleta em si, tendo dados capazes de alimentar projetos com informações.

Segundo Gupta et al. (2013), com a proliferação de smartphones, conectividade móvel de baixo custo com boa cobertura e disponibilidade de vários aplicativos de coleta de dados que podem até contornar as questões de conectividade, as pesquisas em papel estão agora sendo substituídas pela coleta de dados baseada em dispositivos móveis. Esses aplicativos permitem a criação de formulários complexos que incorporam lógicas e verificações nas respostas e podem incluir respostas de várias formas, como texto, imagens, vídeos, áudio, localização GPS e código de barras, entre outros. (GUPTA et al., 2013).

O cenário para aplicativos nesse estilo é mais que ideal. Preece (2016) diz que o envolvimento dos cidadãos na ciência também útil devido à escassez de cientistas em algumas áreas de pesquisa. Em contraste, há uma grande número de cidadãos que, se motivados, podem contribuir para diferentes tipos de projetos científicos em todo o mundo. A disponibilidade de smartphones cada vez mais sofisticados, especialmente aqueles com câmeras de alta resolução, tem incentivado o interesse pela biodiversidade e projetos da natureza. (PREECE, 2016).

O sistema correto consegue garantir a participação de qualquer usuário e permitir sua

colaboração. Uma pesquisa doméstica realizada na África do Sul, de acordo com Tomlinson et al. (2009), teve os dados coletados por mulheres locais contratadas como profissionais de saúde da comunidade. Vinte e quatro *clusters* cobrindo a área de estudo proposta foram selecionados antes da aplicação da pesquisa. Cada casa em todos os 24 *clusters* foi listada usando a estratégia de listagem do censo nacional da África do Sul. O membro sênior de cada família em cada cluster foi entrevistado. Todas as entrevistas foram realizadas face a face, nas residências dos participantes do estudo. (TOMLINSON et al., 2009).

Em resumo, um estudo ou pesquisa de campo demanda preparo, tanto dos métodos que serão adotados nos estudos, quanto dos itens necessários para realizar as anotações e documentações necessárias para posterior análise e elaboração de conclusões. Já a ciência cidadã tem apenas interesse na coleta de dados, não se importando muito com o modo que a coleta irá ocorrer.

Através de um aplicativo, é possível substituir os itens necessários para um estudo de campo e automatizar os processos de registro e observação, além de beneficiar a prática da ciência cidadã ao promover uma melhor coleta de dados. O aplicativo proposto por esse trabalho visa atender essas melhoras, ao mesmo tempo que facilitará a reutilização dos dados coletados ao serem extraidos em JSON, um formato amigável e amplamente utilizado em várias plataformas.

1.1 Objetivos Geral e Específicos

Este trabalho tem como objetivo principal o desenvolvimento de um aplicativo móvel para o registro e documentação de dados de interesse em estudos de campo e ciência cidadã, tais como fotos, audios, anotações, além dos objetos de estudo em si.

1.1.1 Objetivos específicos

- Ser uma ferramenta que auxilie na realização de um estudo de campo e em atividades de ciência cidadã.
- Avaliar a interação do usuário com o sistema.
- Disponibilizar o aplicativo na Play Store.

1.2 Organização do Trabalho

No Capítulo 1, é apresentada a introdução do trabalho, bem como a justificativa e objetivos geral e específicos. No Capítulo 2, temos a fundamentação teórica, que envolveu pesquisas sobre aplicativos de documentação e registro. Logo após, no Capítulo 3 se encontram os trabalhos relacionados. O Capítulo 4 apresenta a metodologia de desenvolvimento do aplicativo. O Capítulo 5 trazem os resultados parciais. As conclusões preliminares são feitas no Capítulo 6, que também traz um cronograma de atividades.

2 Fundamentação Teórica

A seguir são apresentados conceitos básicos para a compreensão do trabalho desenvolvido, separado em quatro tópicos: aplicativos móveis, desenvolvimento móvel, observação e registro e JSON. Dessa forma, é possível compreender o que é uma aplicação móvel e o processo de desenvolvimento. Ainda, o conceito de observação e registro é introduzido de forma a apresentar a importância de estudos de campo, ao mesmo tempo que indica suas dificuldades que podem ser sanadas com tecnologia móvel. Também é introduzido o conceito de ciência cidadã e como a prática se beneficia através de uma coleta e classificação de dados bem executada. Na última sessão apresenta-se o formato JSON, que será o formato que as informações registradas poderão ser extraídas do aplicativo.

2.1 Aplicativos móveis

A presença de aplicativos móveis no cotidiano da maioria da população do planeta se deu a partir do surgimento dos aparelhos móveis inteligentes. Os aplicativos, como enfatizam Islam et al. (2010), são utilizados massivamente por facilitarem a vida de seus usuários, desde a comunicação até o entretenimento.

Segundo Silva e Santos (2014), o aquecimento do mercado de aparelhos cada vez mais inteligentes deu início ao mercado de aplicativos, que oferecem cada vez mais conteúdos e facilitadores.

Aplicativos móveis de acordo com Islam et al. (2010), estão sendo utilizados como fonte de renda por várias empresas, o que exige aparelhos com o máximo de performance possível, para criar um ambiente otimizado para os mesmos. Islam et al. (2010) apresentam diversos impactos positivos dos aplicativos móveis na socidedade, tais como rápida comunicação, economia de tempo e aumento de produtividade, novas vagas de emprego e redução de custos.

2.2 Desenvolvimento móvel

A diversidade de interesses, sistemas operacionais e empresas no mercado de aplicativos móveis deram origem a uma igual diversidade de formas de serem desenvolvidos os aplicativos móveis, como apontam Silva et al. (2015).

Segundo Smutnỳ (2012), existem quatros configurações técnicas de aplicativos:

- **Aplicativos nativos**: otimizados para a plataforma a qual é destinado, mas o desenvolvedor deve refazê-lo caso queira estender para outras plataformas.

- **Aplicativos híbridos**: com o auxílio de frameworks, o desenvolvimento é visado para multiplataformas e possui acesso a funcionalidades de hardware do dispositivo móvel.
- **Aplicativos *web* dedicado**: *website* móvel dedicado a uma plataforma específica.
- **Aplicativo móvel genérico**: um aplicativo móvel desenvolvido para qualquer plataforma com acesso a *web*.

Para desenvolver aplicativos nativos, Charland e LeRoux (2011) listam o conjunto de habilidades e conhecimentos que um desenvolvedor deve ter para cada sistema operacional:

Tabela 2.1 – Conhecimentos necessários para desenvolver aplicativos para sistemas operacionais distintos.

Conhecimentos necessários	
Sistema Operacional	Conhecimentos
Apple iOS	C, Objective C
Google Android	Java
RIM BlackBerry	Java
Symbian	C, C++, Python, HTML/CSS/JS
Windows Mobile	.NET
Windows 7 Phone	.NET
HP Palm webOS	HTML/CSS/JS
Samsung Bada	C++

Fonte: Charland e LeRoux (2011)

Através da Tabela 2.1, podemos verificar que estender um aplicativo para diversos sistemas operacionais já foi outrora um problema. Por isso hoje existe um crescimento da popularidade dos aplicativos híbridos, de acordo com Palmieri et al. (2012), pois atendem diversos sistemas operacionais. Além disso, o conhecimento necessário para atender todos os diferentes sistemas é reduzido.

Para desenvolver aplicativos nesse sentido, existem diversos *frameworks* que os desenvolvedores podem se utilizar. Smutnỳ (2012) lista as seguintes vantagens em utilizar tais ferramentas:

- **Peso leve**: arquivos de tamanhos pequenos são o foco devido a limites de conexão.
- **Otimização ao toque**: como os cursores de mouses são substituidos pelo toque humano em dispositivos móveis, os *frameworks* estão preparados para lidar com essa diferença através de elementos específicos.
- **Padrões HTML5 e CSS3**: a maioria dos dispositivos móveis contam com suporte a estes padrões, e por isso um dos focos dos frameworks é tirar proveito de funcionalidades baseadas nestes para aperfeiçoar os aplicativos.

Essa popularidade de aplicações WEB puras, desenvolvidas para rodarem no navegador *web* do aplicativo, também é apontada por Silva e Santos (2014). Tecnologias *web* padrão como HTML, CSS e *Javascript* são necessárias para construir a aplicação, mas ainda assim ela terá um comportamento muito similar a um aplicativo nativo. As aplicabilidades mais modernas de HTML5 e CSS3 tornam isso possível.

2.2.1 Ferramentas

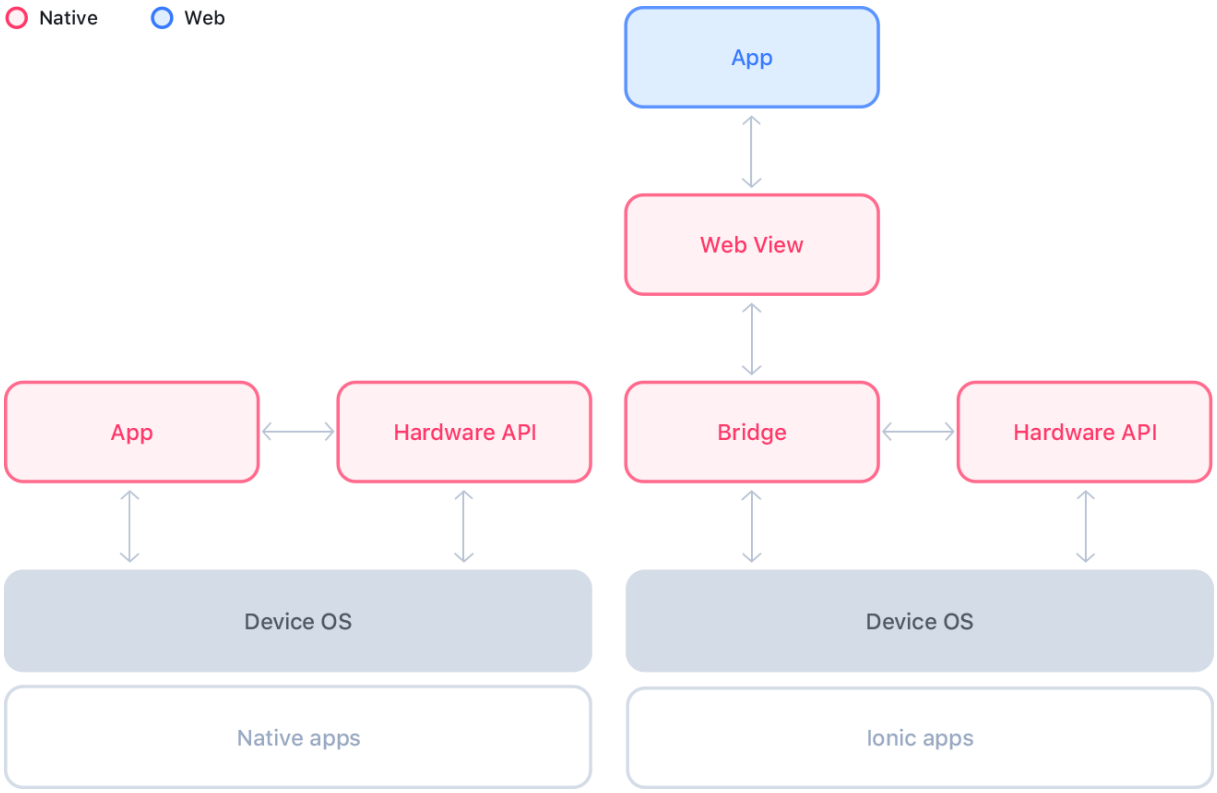
As ferramentas para desenvolvimento de aplicações em geral podem ser de complexo entendimento, fazendo-se necessário categorizá-las. Hartmann et al. (2011) as separa da seguinte forma:

- **Biblioteca**: pequenos códigos de ferramentas que oferecem funcionalidades muito específicas. Bibliotecas de máscaras para campos de entrada, acesso à câmera, ao microfone, à galeria, validações de formatos de *string* são alguns exemplos de bibliotecas.
- ***Framework***: com o auxílio das bibliotecas, arquiteturas e diretrizes, temos a ferramenta - ou conjunto de ferramentas - necessárias para se desenvolver uma aplicação móvel completa.
- **Plataforma**: O resultado da conjunção dos *frameworks*, serviços e ferramentas que servem de receptor para a aplicação desenvolvida, tanto para uso quanto para distribuição. Além disso podem fornecer ferramentas de documentação e automação.
- **Produto/Serviço**: Funcionalidade ou serviço pronto para ser usado através de uma aplicação em um dispositivo móvel.

2.2.2 Ionic

De acordo com Silva e Sotto (2018), Ionic é um *framework* de desenvolvimento de aplicativos móveis em HTML5, com foco em desenvolvimento híbrido, tendo o Angular - um *framework JavaScript* - como suporte às operações lógicas e aritméticas na plataforma. Na definição de Silva e Sotto (2018), os aplicativos desenvolvidos através do Ionic são uma camada de navegador embutida em aplicativo, que utiliza de bibliotecas adicionais para requisitar recursos nativos do dispositivo.

Figura 2.1 – Arquitetura de um aplicativo em Ionic.



Fonte: https://bit.ly/3pVqKSW. Acessado em: 05/02/2021

A Figura 2.1 compara a arquitetura de um aplicativo desenvolvido em Ionic com a de aplicativos nativos.

Como podemos verificar, utiliza-se a camada de navegador como ponte entre o aplicativo e o aparelho, o que torna possível desenvolver aplicativos para sistemas operacionais diversos.

Atualmente na sua quinta versão, o *framework* em questão conta com uma enorme quantidade de aplicativos desenvolvidos e uma comunidade extremamente ativa, assim sendo escolhido como ferramenta para desenvolvimento do Tempus.

2.3 Observação e registro

A seguir, descrevemos as áreas de estudo de campo, ciência cidadã e contextos em que o aplicativo se mostra uma ferramenta de auxílio.

2.3.1 Estudos de campo

A importância de estudos de campo que promovem observações e registros na área da Biologia - e em áreas correlatas - é apontada por Trevisan e Silva-Forsberg (2014), por tornar o ensino mais agradável e contextualizado, ao mesmo tempo que permite ao educador aplicar novos métodos de ensino e processos cognitivos que auxiliam na imersão nas atividades.

Na visão de Seniciato e Cavassan (2004), unir a educação com aspectos afetivos - neste caso a natureza - tem se mostrado um excelente método pedagógico de ensino, por inserir o aluno como elemento participante da natureza, e por consequência, responsável por esta.

De acordo com Marçal, Andrade e Viana (2017), os envolvidos em um estudo de campo encontram diversos obstáculos que podem ser prejudiciais à pratica. Dispersão e a necessidade de carregar equipamentos auxiliares - tais como GPS, bússola e caderneta de campo - são alguns deles.

Tendo concluído que estudos de campo são necessários e eficientes mas ainda assim imperfeitos, Marçal, Andrade e Viana (2017) apontam as tecnologias móveis como um meio para contornar tais dificuldades, por trazerem diversos benefícios aos estudos. Destes, destacamos a integração de ferramentas em um único dispositivo para angariação e armazenamento de informações cruciais. A documentação de um ambiente permite posterior análise de suas mais variadas características e fenômenos.

2.3.2 Ciência cidadã

Os projetos de pesquisa que estimulam usuários a coletar, classificar, analisar e transcrever dados científicos são conhecidos como ciência cidadã, segundo Bonney et al. (2014). Esses projetos podem tratar de vários tópicos relacionados à fauna, flora, qualidade das águas e até mesmo problemas sociais, como questões de asfaltamento e saneamento básico.

Bonney et al. (2014) explica que os dados são coletados por indíviduos (pesquisadores ou não) ou por pequenos times de pesquisa, que se comprometem a fornecer dados para os projetos, sejam voluntários ou contratados. A Figura 2.2 nos dá um exemplo de coleta de dados por indivíduos não pesquisadores, reafirmando que qualquer pessoa pode contribuir para um projeto de ciência cidadã.

Figura 2.2 – Mulheres de Komo (República do Congo) aprendendo a mapear em uma floresta, como parte do projeto Extreme Citizen Science (ExCiteS) Intelligent Maps.

Fonte: Bonney et al. (2014)

Ainda que seja uma prática benéfica para projetos sociais e científicos, a qualidade dos dados é questionada por alguns membros da sociedade científica (BONNEY et al., 2014). Já outros acreditam que essa área ainda não atingiu seu potencial completo e pode melhorar a forma como ajuda projetos científicos em geral.

A ideia é que, à medida que a prática seja difundida, as pessoas passem a utilizar o termo "ciência cidadã" para descrevê-la e também a confiar nos dados produzidos por ela, sejam cientistas, políticos e o público em geral (BONNEY et al., 2014).

Dessa forma, Tempus se propõe a ser uma ferramenta que auxilie também a prática de ciência cidadã, melhorando a forma em que esses dados são coletados, categorizados e exportados.

2.4 JSON

As informações após serem modeladas e registradas, se tornam mais interessantes se puderem ser extraídas e aproveitadas em outros sistemas e aplicações. Segundo Bourhis et al. (2017), um *JSON* (ou Notação de Objeto *Javascript*) é um objeto de pequeno tamanho e do padrão dos tipos de dados da linguagem de programação *Javascript*.

Basicamente, um *JSON* armazena informações através do padrão chave-valor, sendo este segundo passível de receber outra chave com outro valor, o que torna o objeto muito dinâmico.

Sua simplicidade, como aponta Bourhis et al. (2017), é responsável por ser um motivo para que esse formato seja adotado quase que universalmente para troca de dados na *web*.

Figura 2.3 – Objeto JSON simples.

```
{
  "name": {
     "first": "John",
     "last": "Doe"
     },
  "age": 32,
  "hobbies": ["fishing","yoga"]
}
```

Fonte: Bourhis et al. (2017)

Na Figura 2.3 temos a representação de uma notação de objeto *Javascript*, demonstrando sua simplicidade de reconhecimento tanto humano quanto lógico (BOURHIS et al., 2017). Isso é comprovado pela massiva utilização desse formato em API's (*Application Programming Interface*) e em seus processos de requisição e resposta através do protocolo HTTP (*HyperText Transfer Protocol*).

3 Trabalhos Relacionados

Nesse capítulo apresentamos alguns trabalhos correlatos. Os critérios adotados para seleção dos trabalhos correlatos foram, em caráter primário, artigos científicos. Na busca, foram combinadas as palavras aplicativo, estudo de campo, biologia, documentação. A pesquisa foi feita em português e inglês.

O trabalho encontrado de Kravcik et al. (2004), o sistema RAFT (*Remote Accessible Field Trips*), é uma aplicação para coleta de dados em estudos de campo. O aplicativo móvel de Marçal et al. (2014), o Geomóvel, se assemelha com o aplicativo proposto, ao permitir registro de coordenadas geográficas e registrar informações por áudio e fotos. O aplicativo Leafsnap, de Kress et al. (2018) permite a documentação de árvores e folhas através de uma interface bastante intuitiva. Já o aplicativo Scihub de Moresi et al. (2017) aborda a prática da ciência cidadã através de um maior engajamento em projetos científicos. A seguir, todas as aplicações são descritas com detalhes.

3.1 RAFT

O projeto RAFT é um aplicativo móvel que tem como objetivo auxiliar em ambas coleta e anotação de dados (KRAVCIK et al., 2004). Ele atende as seguintes demandas:

- **Criação**: Criação de objetos de dados relacionados ao estudo de campo.
- **Coleta de dados**: coletar dados do campo - fotos, por exemplo - e a possibilidade de armazenamento em um repositório compartilhado.
- **Anotações**: Possibilidade de anotação de dados coletados, através de áudios e escrita.
- **Recuperação**: Navegação no sistema para procura de dados previamente catalogados.
- **Proteção de direitos autorais**: usuários podem editar seu conteúdo e apenas visualizar o conteúdo de outros usuários.

Segundo Kravcik et al. (2004), por atender demandas diversas de coleta e análise de dados, cada funcionalidade no RAFT necessita uma tela e dispositivo. O público alvo também não deve se estressar com dificuldades e designs não intuitivos, pois não são obrigatoriamente acostumados com tecnologias (KRAVCIK et al., 2004).

A Figura 3.1 mostra como é feita a captura de uma foto através do sistema RAFT. Uma prévia de vídeo aparece do lado direito e, após tirar a foto, ela aparece debaixo do botão de

captura. Logo após, ele pode salvá-la em sua coleção, onde a imagem aparece com uma miniatura que a identifica (KRAVCIK et al., 2004).

Figura 3.1 – Captura de foto em RAFT.

Fonte: Kravcik et al. (2004)

De acordo com Kravcik et al. (2004), a imagem também é identificada por outros detalhes, tais como tópico, tarefa, coleção e título. Palavras chaves também são adicionadas para facilitar posterior loacalização.

Em resumo, o sistema RAFT é muito intuitivo e de fácil uso. Pelo tamanho da interface, podemos identificar que a aplicação é destinada a *tablets*, como nos mostra a Figura 3.2. Os *tablets* são dispositivos móveis com certa popularidade mas ainda assim não tão comuns quanto aparelhos celulares. Outro fator negativo é seu idioma, que não é o português.

Figura 3.2 – Sistema RAFT em uso.

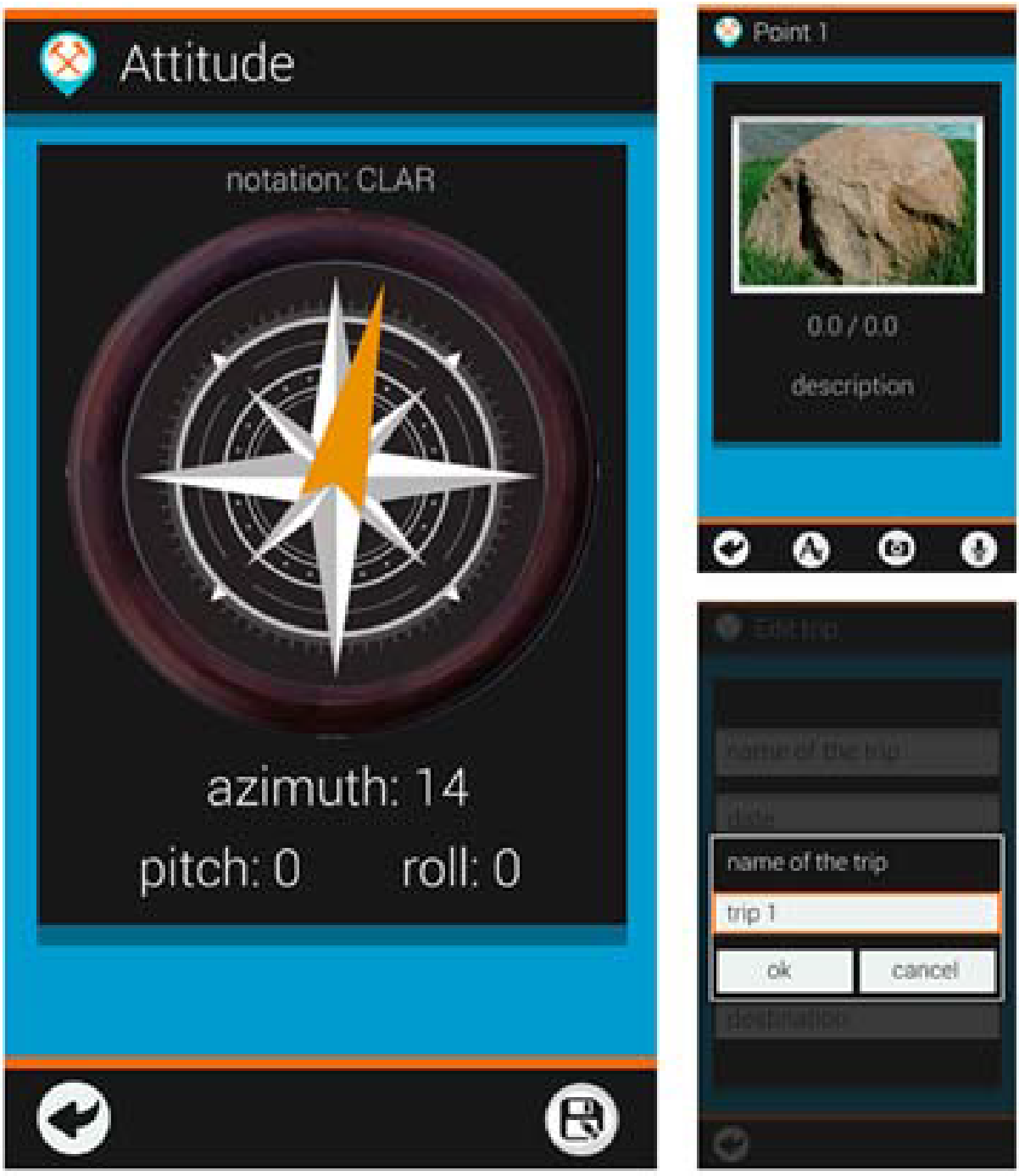
Fonte: Kravcik et al. (2004)

3.2 Geomóvel

O Geomóvel, como mostra Marçal et al. (2014), é um aplicativo móvel que une ferramentas de mídia e sensores para auxiliar estudantes em aulas de campo. Medição de ângulos, registro de coordenadas geográficas e captura de fotos, gravação de áudios e escritas de textos são as principais funções da aplicação.

Segundo Marçal et al. (2014), o Geomóvel é compatível com dispositivos Android. As funções de registro incluem escrita, gravação e captura. As informações são armazenadas em um banco de dados local e coordenadas geográficas obtidas através do sensor GPS do aparelho. A Figura 3.3 nos mostra algumas capturas de tela do aplicativo.

Figura 3.3 – Aplicativo Geomóvel.



Fonte: Marçal et al. (2014)

GPS, acelerômetro e magnetômetro são os 3 sensores utilizados para registro de informações de localização no Geomóvel (MARÇAL et al., 2014). O GPS registra as coordenadas geográficas, latitude e longitutde. Medições de ângulos ficam por conta do acelerômetro e magnetômetro, que simulam um compasso.

A compatibilidade exclusiva com o Android é um ponto negativo do Geomóvel, pois limita a sua extensão a outros sistemas operacionais. Ainda assim, sua captura de coordenadas geográficas se mostra muito interessante, pois é um dado importante em extrações que visam

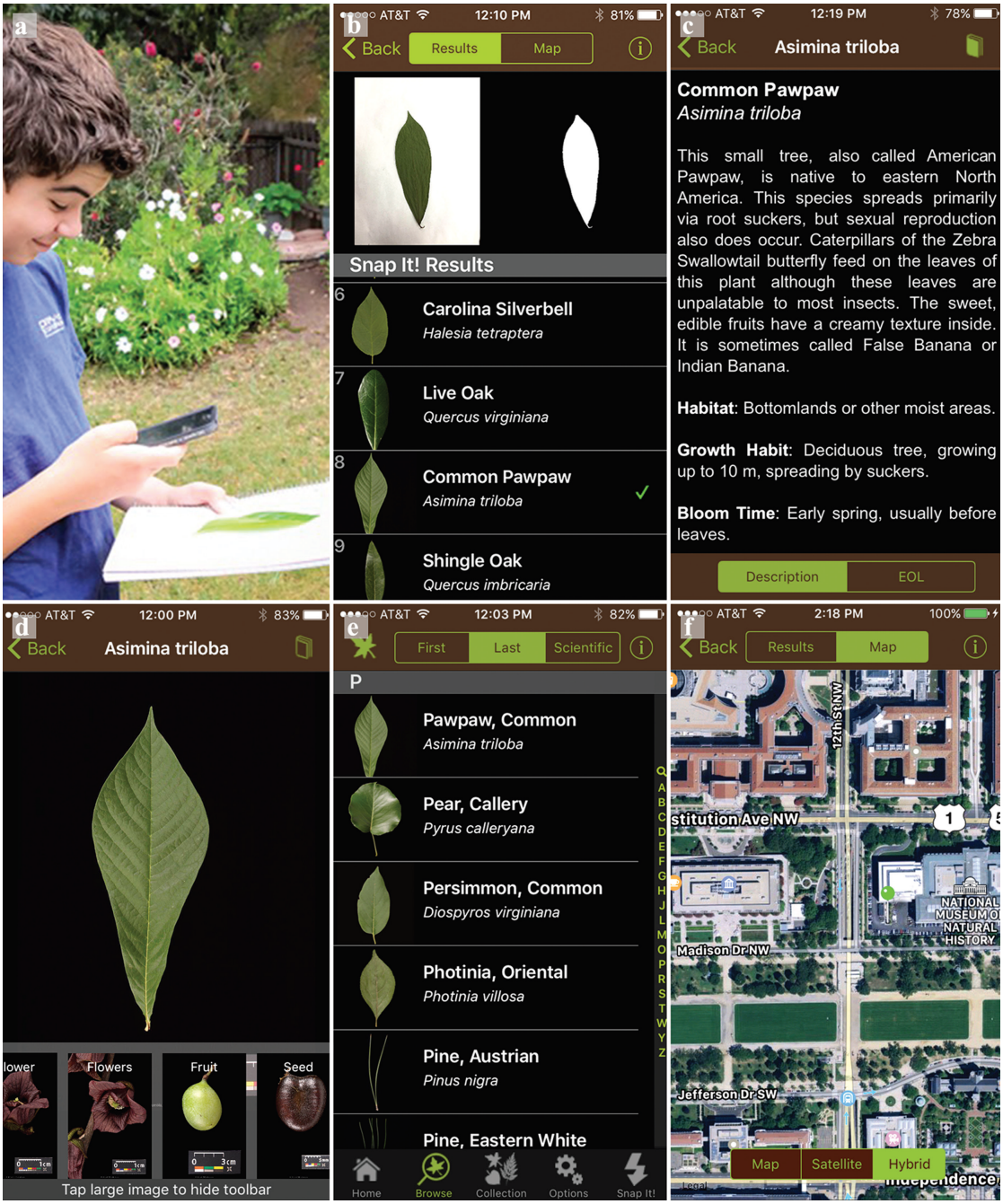
reutilizar estes dados em outros sistemas.

3.3 Leafsnap

Segundo Kress et al. (2018), o aplicativo Leafsnap foi originalmente desenvolvido como uma ferramenta de ajuda para botânicos de taxonomia para encontrar informações sobre novas espécies localizadas em habitats pouco conhecidos ao redor do mundo.

A popularização de dispositivos móveis acabou levando à evolução do aplicativo, que ganhou novas funções como mapeamento de locais, distribuição geográfica e outras informações relevantes, mas sempre voltadas para a área de botânica (KRESS et al., 2018). A Figura mostra algumas telas do aplicativo e suas funcionalidades.

Figura 3.4 – Aplicativo Leafsnap.



Fonte: Kress et al. (2018)

Depois que o usuário usa o Leafsnap para tirar uma foto da folha, o algoritmo de reconhecimento compara a forma da folha com a forma das milhares de folhas identificadas na biblioteca de imagens do banco de dados do Leafsnap. Uma lista de espécies candidatas aparecere na tela e, para cada espécie candidata, o usuário pode navegar entre breves descrições botânicas e imagens de estruturas reprodutivas e vegetativas. Ele também pode navegar por árvores registradas por outros usuários no Leafsnap, para encontrar espécies de árvores adequadas. Então, o usuário marca seu ID e envia sua imagem e dados de localização para o servidor. (KRESS et al., 2018).

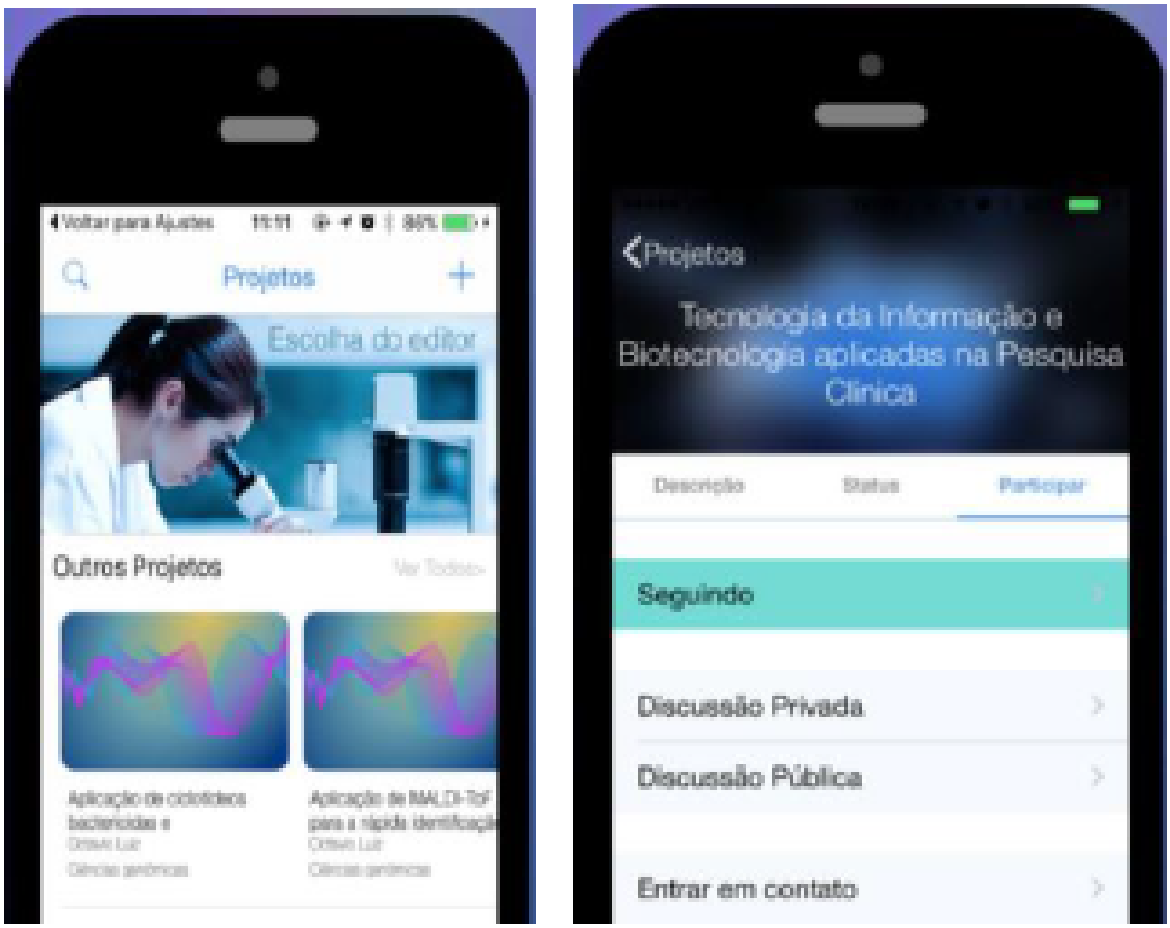
Podemos perceber que o aplicativo é voltado apenas para a plataforma iOS, além de ser

voltado para área da botânica, o que restringe o tipo de usuário que utiliza o sistema.

3.4 SciHub

O SciHub é um aplicativo desenvolvido para a plataforma iOS, e seu principal objetivo é aumentar a visibilidade, a exposição e a comunicação entre pesquisadores e partes interessadas no projeto (MORESI et al., 2017). Seu diferencial se encontra em reunir e listar projetos de pesquisa científica em um único lugar. Ao abrir o aplicativo, o usuário pode visualizar projetos científicos e participar de discussões sobre eles. A Figura 3.5 mostra algumas telas do aplicativo.

Figura 3.5 – Aplicativo Scihub [1].



Fonte: Moresi et al. (2017)

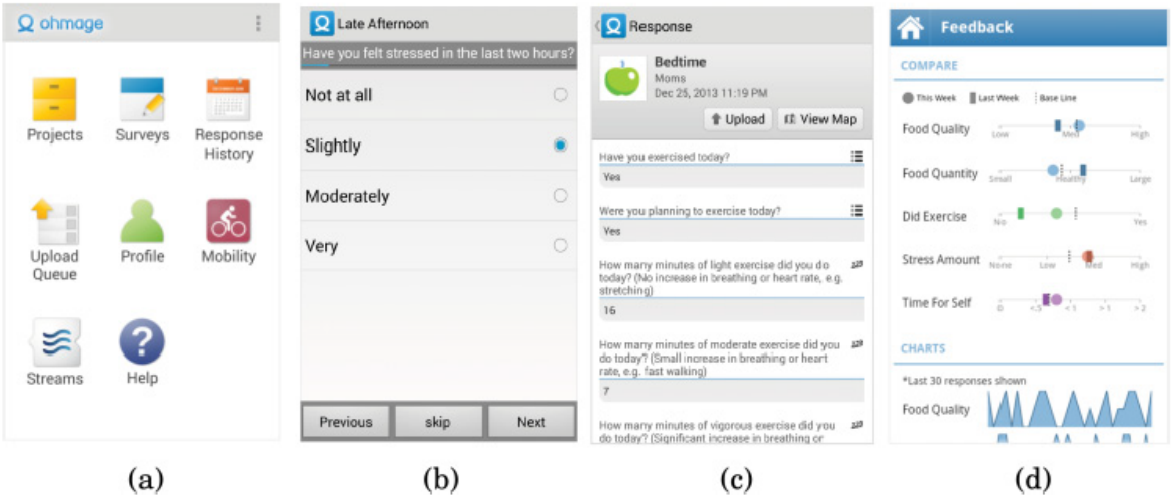
Através do aplicativo, usuários podem engajar nas discussões dos projetos que estão seguindo, e contribuir com informações relevantes. Ainda assim, o aplicativo não é multiplataforma e não permite uma coleta individualizada de dados, além de a ação de inserir imagens nas discussões não ter sido mencionada no artigo.

3.5 Ohmage

O Ohmage é um aplicativo híbrido, disponível para Android e iOS, que promove a coleta, armazenamento, análise e visualização compartilhada de dados.

Ohmage foi projetado especificamente para permitir uma configuração fácil de coletas de dados sistemáticos em que os indivíduos capturam tanto dados de pesquisa - fotos, áudios e vídeos, enquetes -, quanto dados passivos - coordenadas de GPS, acelerômetros - ao longo do uso do aplicativo em seus smartphones. (RAMANATHAN et al., 2012). A Figura 3.6 ilustra as telas do aplicativo em funcionamento.

Figura 3.6 – Aplicativo Ohmage. (a) A tela inicial do aplicativo mostrando diferentes opções de menu. (b) Um prompt para entrada do usuário. (c) Uma pesquisa concluída. (d) Feedback do usuário mostrando seu status atual versus sua linha de base em um estudo.



Fonte: Ramanathan et al. (2012)

Os usuários de Ohmage usam seus dispositivos móveis para capturar diversos dados espaço-temporais por meio de ações, como tirar uma foto, gravar um vídeo, quanto através de fluxos de dados capturados passivamente (GPS, acelerômetros). Os usuários esperam ser capazes de coletar dados em qualquer lugar e a qualquer hora com ou sem conectividade de rede. Portanto, os aplicativos precisam operar sem problemas em uma variedade de ambientes. Além disso, segundo Ramanathan et al. (2012), o consumo de bateria deve ser consciente para que a coleta de dados passivos não prejudique a duração da carga, o que pode acarretar num mau funcionamento da aplicação.

Atualmente, Ohmage se encontra tanto na App Store quanto na Play Store sob o nome de UCLA MobilizingCS, sendo mantido pela Universidade da Califórnia, Los Angeles, e sua última atualização foi há 3 anos. A língua do aplicativo é apenas em inglês. Apesar destes pontos, é um aplicativo muito completo e mais próximo do que queremos chegar.

3.6 Sensr

O aplicativo Sensr se apresenta como uma ferramenta para criação de eventos de coletas de dados, onde temos os autores do evento, e voluntários que se inscrevem no evento para contribuir com seus próprios dados coletados.

De acordo com Kim, Mankoff e Paulos (2013), Sensr conta com três opções principais: Meus Eventos, Meus Dados e Configurações. A primeira permite ver os eventos que o usuário está participando, além de permitir procurar eventos por categorias. A segunda mostra todos os dados coletados pelo usuário. Nas configurações, o usuário pode adicionar algumas informações pessoais, se desejar.

Figura 3.7 – Capturas de tela do aplicativo móvel. Da esquerda para a direita, uma lista de eventos que um usuário adicionou, uma categoria para encontrar e adicionar um evento, uma página principal do evento que pode ser visualizada por mapa ou visualização de lista, um repositório dos dados e um formulário personalizado para relatar dados

Fonte: Kim, Mankoff e Paulos (2013)

A ideia de eventos compartilhados de coleta de dados do Sensr é bem interessante, permitindo que um usuário autor veja os dados coletado por um ou mais membros de seu evento. Ainda assim, o aplicativo não se encontra na App Store, mesmo sendo exclusivo para iOS, além de estar somente em inglês.

3.7 Discussão

A Tabela 3.1 destaca as diferenças dos trabalhos correlatos com a aplicação móvel proposta. As métricas para comparação utilizadas são encontradas na primeira coluna, e as restantes representam respectivamente cada um dos trabalhos e o aplicativo proposto. As métricas dizem respeito ao armazenamento, criação e coleta dos dados registrados. Ainda, temos mais duas métricas que avaliam se o idioma do aplicativo é o português do Brasil e se o mesmo é

multiplataforma (Android e iOS). A métrica denominada Ativo diz respeito se um aplicativo continua ativo, ou seja, se não foi descontinuado e continua recebendo atualizações constantes. A métrica de Personalização fala da capacidade da aplicação em criar categorias personalizadas para seus elementos. O aplicativo proposto tem a intenção de atender todas as métricas de avaliação.

Tabela 3.1 – Comparação entre aplicativos

Comparação entre os aplicativos							
Métricas	**RAFT**	**Geomóvel**	**Leafsnap**	**Scihub**	**Ohmage**	**Sensr**	**Aplicativo proposto**
Criação e coleta de dados	X	X	X		X	X	X
Armazenamento de dados	X	X	X	X	X	X	X
Idioma do aplicativo em Português do Brasil		X		X			X
Aplicativo multiplataforma (Android e iOS)					X		X
Ativo							X
Personalização							X

Fonte: Elaborado pelo autor

4 Desenvolvimento

4.1 Metodologia

O primeiro estágio desse trabalho envolveu a pesquisa de estudos de campo e como esses funcionam. Marçal, Andrade e Viana (2017) lista equipamentos auxiliares como um obstáculo a ser superado para alguns campistas, e como a tecnologia móvel pode auxiliar.

Após a pesquisa, foi realizada uma revisão bibliográfica que pode se dividir em duas partes: paradigmas de desenvolvimento móvel e trabalhos relacionados. A primeira visa dar um contexto mais amplo ao trabalho, que se trata de um aplicativo desenvolvido por meio de tecnologia híbrida (Ionic *framework*), enquanto a segunda busca trabalhos semelhantes. O *framework* estará detalhado na Seção 4.4. Assim, foram encontrados 4 aplicativos relacionados. Foram abordadas as formas que cada aplicativo trata o problema. Logo em seguida, foi proposto um aplicativo que reúne as ferramentas em comum, visando a criação de um sistema mais completo. A Tabela 3.1 mostra a comparação entre eles.

Após a revisão bibliográfica, os requisitos funcionais e não funcionais foram levantados. Dentre os requisitos funcionais estão as funções comuns de todo aplicativo móvel, tais como login, cadastro de usuário e recuperação de senha. Além desses, temos a criação de variáveis de ambiente e registros de fotos. Já os não funcionais se referem a questões de usabilidade, acessibilidade e segurança. O levantamento está detalhado na Seção 4.2.

O protótipo do aplicativo foi realizado pela plataforma *Figma*. Foram desenhados telas, campos, botões e menus. Na Seção 4.3 haverão mais detalhes do protótipo e as respectivas telas.

O desenvolvimento do aplicativo é descritos na Seção 4.4, discorrendo sobre o framework utilizado e as telas. Os testes, tanto automatizados quanto o SUS (System Usability Scale) são descritos na Seção 4.5.

4.2 Levantamento de Requisitos

Para entendermos de maneira clara o funcionamento de um aplicativo, assim como suas limitações, é importante listarmos os seus requisitos, que se dividem entre funcionais e não funcionais. Figueiredo (2011) descreve os requisitos funcionais como descritores explícitos das funções e serviços do sistema, além de definirem como o sistema deve reagir e o que não pode fazer. Possuem como atributos uma definição consistente e assertiva para cada um de seus serviços, embora na prática seja quase impossível alcançar taís características (FIGUEIREDO, 2011). Já os requisitos não funcionais se referem a questões de usabilidade, velocidade, confiabilidade, facilidade de uso e robustez (FIGUEIREDO, 2011).

A Tabela 4.1 lista os requisitos funcionais da última versão do aplicativo. Além disso, uma breve descrição de cada um dos requisitos é apresentada. Os requisitos RF01, RF02 e RF03 são fundamentais de todo aplicativo móvel. Os requisitos RF04 e RF05 permitem ao usuário compreender o funcionamento do aplicativo e sua finalidade.

Tabela 4.1 – Requisitos Funcionais.

	REQUISITO	DESCRIÇÃO
RF01	Login	O usuário deve ser capaz de fazer login no aplicativo
RF02	Cadastro	O usuário deve ser capaz de se cadastrar no sistema
RF03	Recuperação de senha	O usuário deve ser capaz de recuperar sua senha
RF04	CRUD de Elementos	O usuário deve ser capaz de criar, visualizar, editar e excluir elementos
RF05	CRUD de Elementos personalizados	O usuário deve ser capaz de criar elementos personalizados (com nome e ícone próprio), além de visualizar, editar e excluir

Fonte: Elaborado pelo autor

A Tabela 4.2 lista os requisitos não funcionais. O RNF01 garante a característica híbrida do aplicativo, podendo ser utilizado tanto em dispositivos Android quanto iOS.

Tabela 4.2 – Requisitos Não Funcionais.

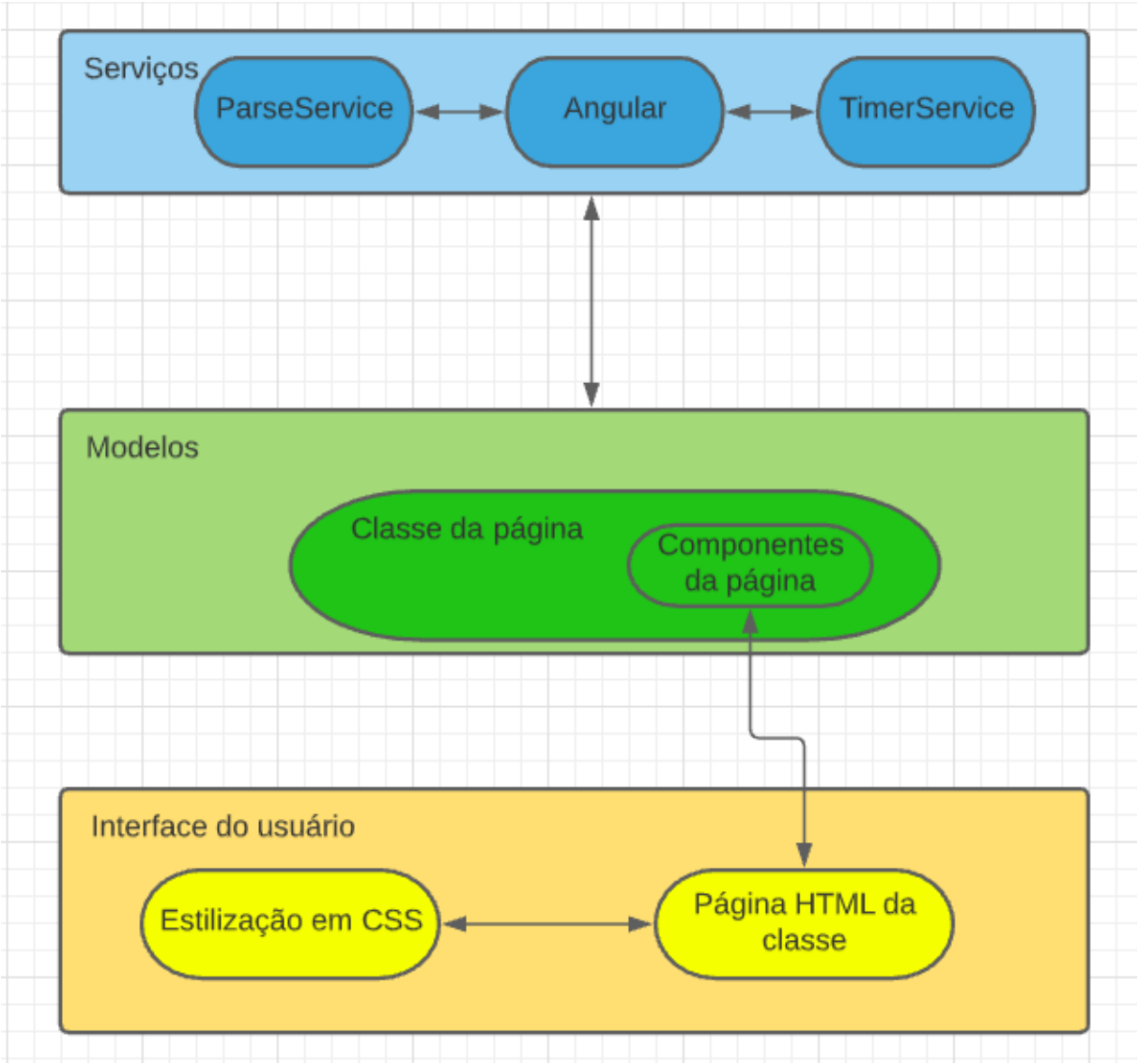
	REQUISITO	DESCRIÇÃO
RNF01	Plataforma Android e iOS	O aplicativo deve ser compatível com os sistemas operacionais Android e iOS.
RNF02	Integridade	Os dados fornecidos pelo usuário devem estar protegidos e seguros.

Fonte: Elaborado pelo autor

4.3 Protótipo

Amoretti (2001) define protótipo como um conjunto de propriedades mais relevantes de um determinado conceito. Essas propriedades auxiliam em uma construção mental e reconhecimento através de comparação de categorias e padrões, o que possibilita o sujeito saber o que está vendo ao comparar com o protótipo que tem na memória (AMORETTI, 2001). Arquiteturar a solução a ser desenvolvida é tão importante quanto o desenvolvimento em si, já que norteia o desenvolvimento, esclarece onde se quer chegar e quais padrões seguir para realizar o desenvolvimento. A Figura 4.1 detalha a arquitetura da aplicação.

Figura 4.1 – Protótipo: Arquitetura da solução.



Fonte: Elaborado pelo autor

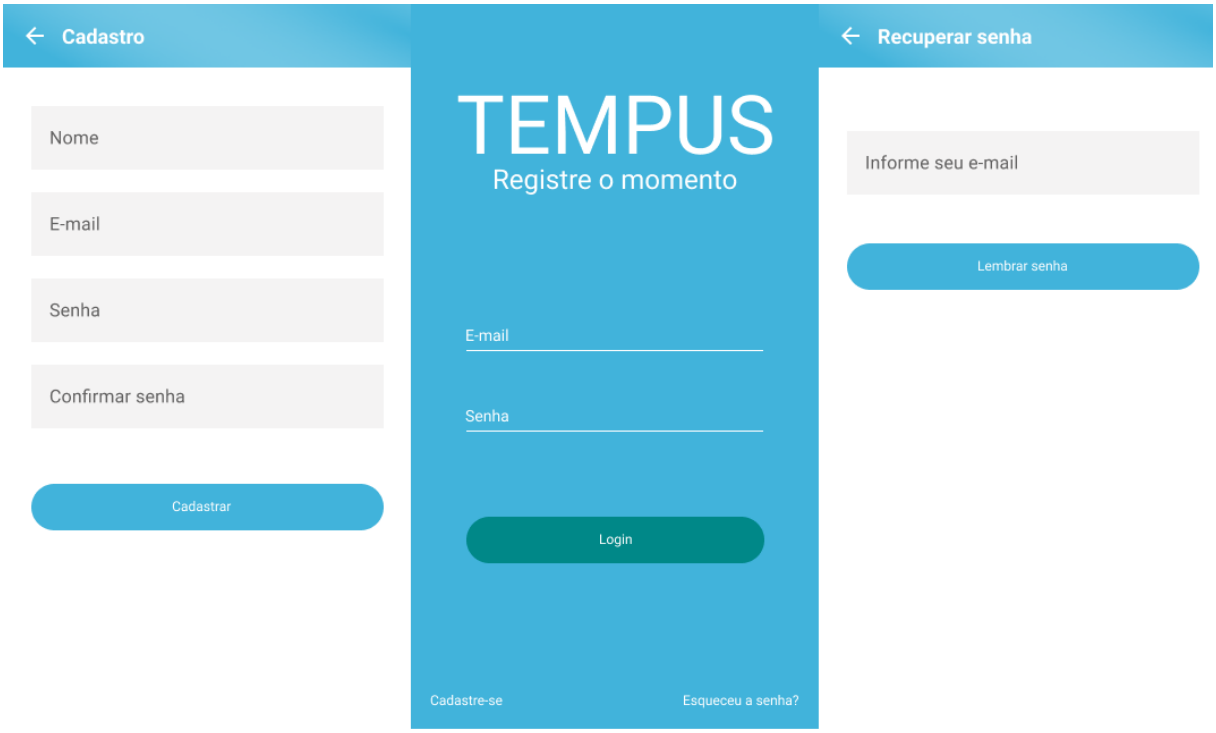
A camada de Serviços fornece toda a estrutura necessária para realizar requisições, seja ao banco de dados, seja às bibliotecas do Angular que realizam funções como navegação e criação de modais, por exemplo. O Serviço TimerService foi implementado para medir o tempo que os usuários gastaram em um roteiro de testes.

A camada de Modelos contém todas as páginas da aplicação. Elas são criadas através de classes e construtores, que podem incluir elementos da camada de Serviços. Os componentes da página dizem respeito a propriedades exclusivas de cada página, tais como formulários e outras variáveis privadas das mesmas.

A camada de Interface do usuário engloba toda a estrutura de visualização. As páginas são desenhadas através da linguagem de marcação HTML, e estilizada utilizando-se CSS. Ações podem ser tomadas através dos elementos das páginas, que avisam aos componentes que, por sua vez, solicitam que serviços sejam executados. Assim, temos a arquitetura da solução.

A Figura 4.2 mostra o protótipo de algumas telas. A tela de Login é exibida quando o usuário abre o aplicativo pela primeira vez e onde o mesmo insere seu e-mail e senha para autenticar sua entrada no sistema. Ela garante que o requisito RF01 é atendido. Caso o mesmo ainda não possua uma conta, é possível se cadastrar, como mostra a tela de cadastro, que atende ao requisito RF02. Se a senha for esquecida, pode ser requisitada uma recuperação através da tela de recuperação de senha, cumprindo a exigência do RF03.

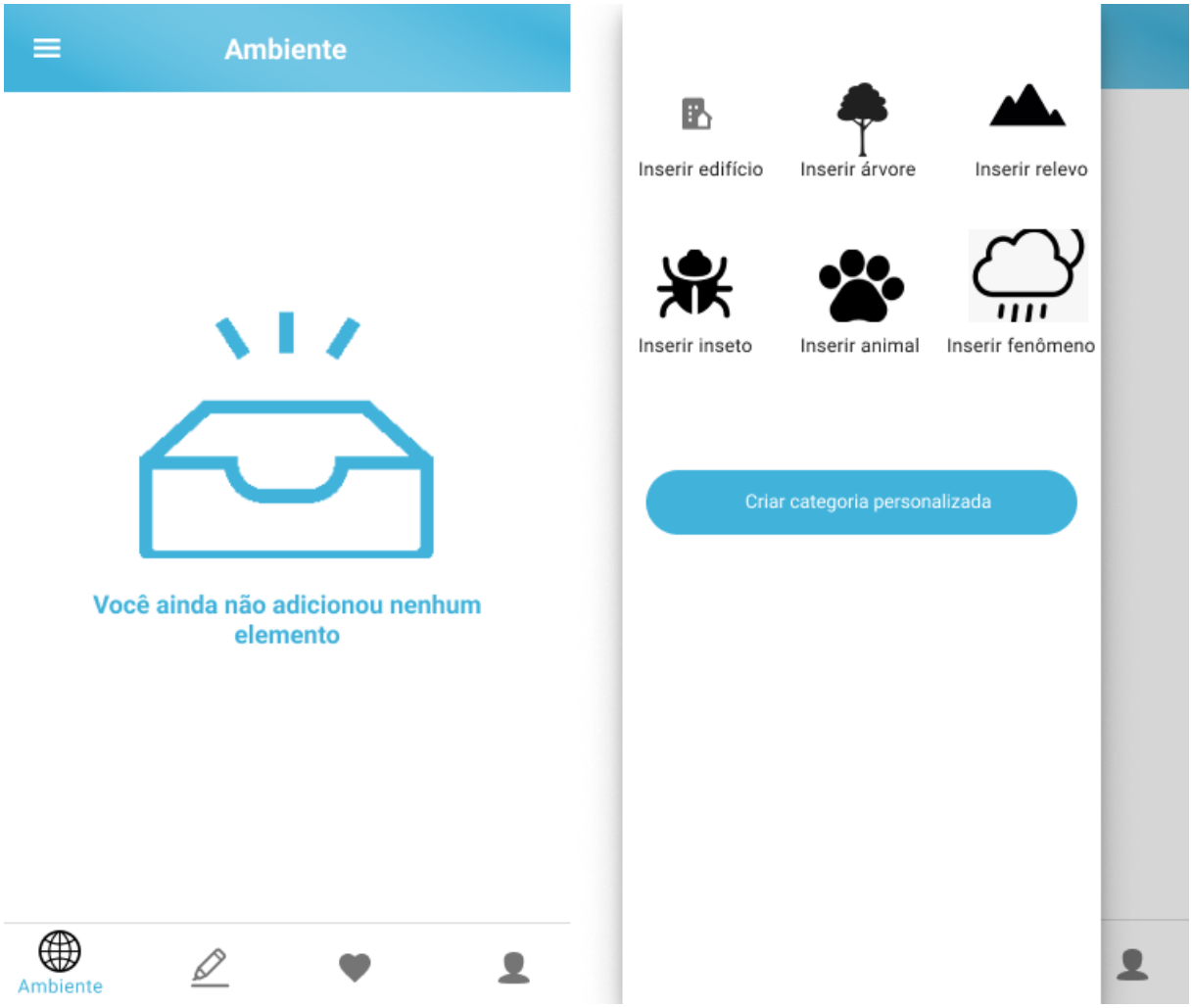
Figura 4.2 – Protótipo: Telas iniciais.



Fonte: Elaborado pelo autor

A Figura 4.3 mostra a tela principal quando o usuário abre o aplicativo e já está autenticado. Quando não possui nenhum objeto registrado, é exibido um aviso. Ainda, é possível abrir um menu lateral que exibe as opções de categorias de objetos a serem adicionados.

Figura 4.3 – Protótipo: Tela inicial.



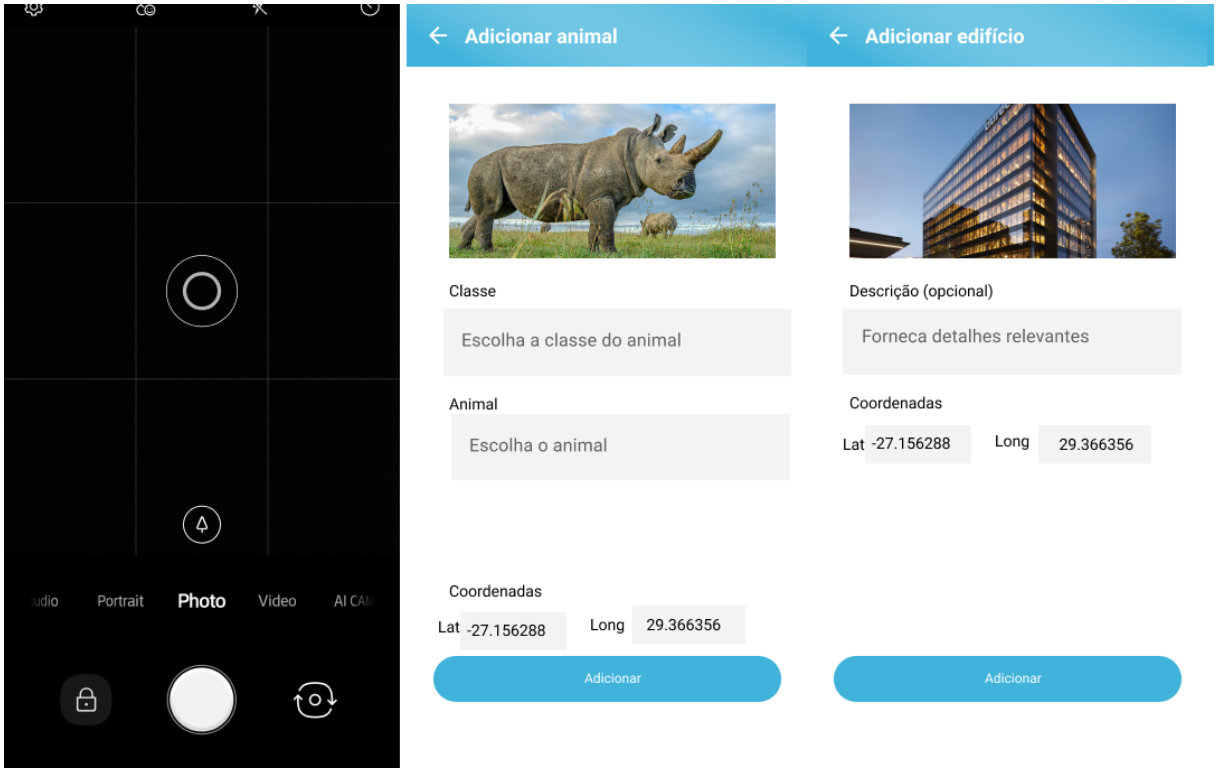
Fonte: Elaborado pelo autor

4.3.1 Inserção de objetos pré-definidos

As duas primeiras formas de inserção são mostradas pela Figura 4.4. Ao se escolher o registro de qualquer variável, é solicitado que o usuário tire uma foto ou utilize uma foto da galeria para identificar o objeto. Além disso, todos os objetos que serão registrados devem ter suas coordenadas de localização fornecidas. As telas a seguir garante adesão ao RF04.

Registrar um animal exige o fornecimento de sua classe - peixes, anfíbios, répteis, aves e mamíferos - e qual o animal em específico. Já ao escolher gravar um objeto edifício pode se registrar descrições de detalhes relevantes sobre o mesmo.

Figura 4.4 – Protótipo: Formas de inserção 1.

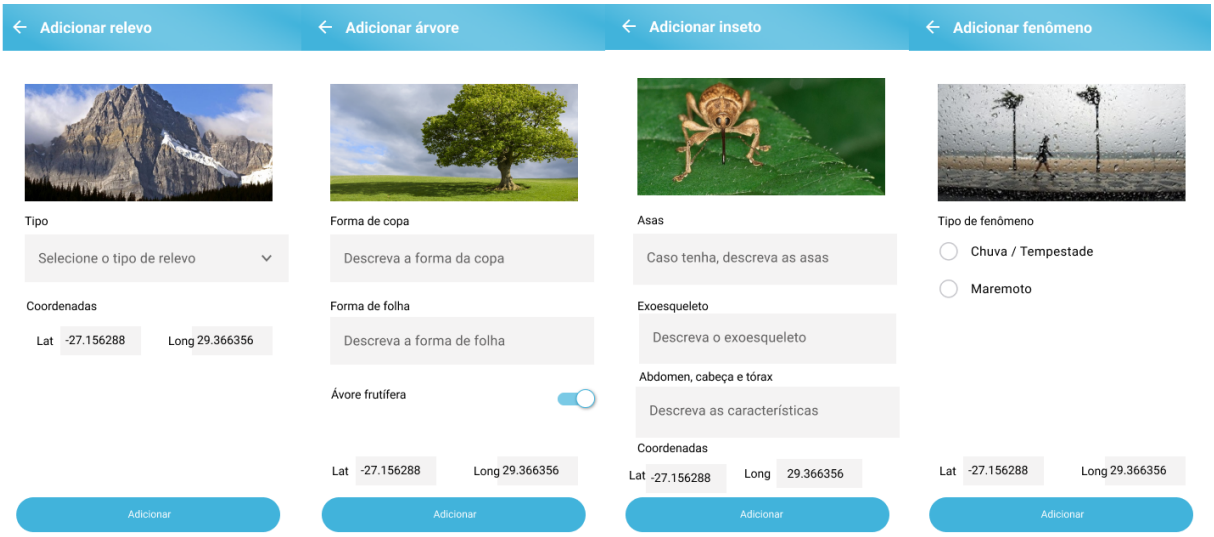


Fonte: Elaborado pelo autor

A Figura 4.5 mostra as outras formas de registro de elementos. Adicionar um relevo requer a classificação do mesmo como montanha, planalto, planície ou depressão. A gravação de uma árvore pode ser mais detalhada, com campos para descrição das formas da copa e da folha. Ainda é possível informar se a mesma é frutífera ou não.

O registro de insetos também segue a mesma proposta, com campos para descrever o exoesquesleto, asas (caso tenha), além de cabeça, abdômen e tórax. Por fim, a opção de adicionar um fenômeno - meteorológico ou tectônico - se dá pela classificação do mesmo pelas opções de chuva/tempestade, neve, maremoto, furacão ou terremoto.

Figura 4.5 – Protótipo: Formas de inserção 2.



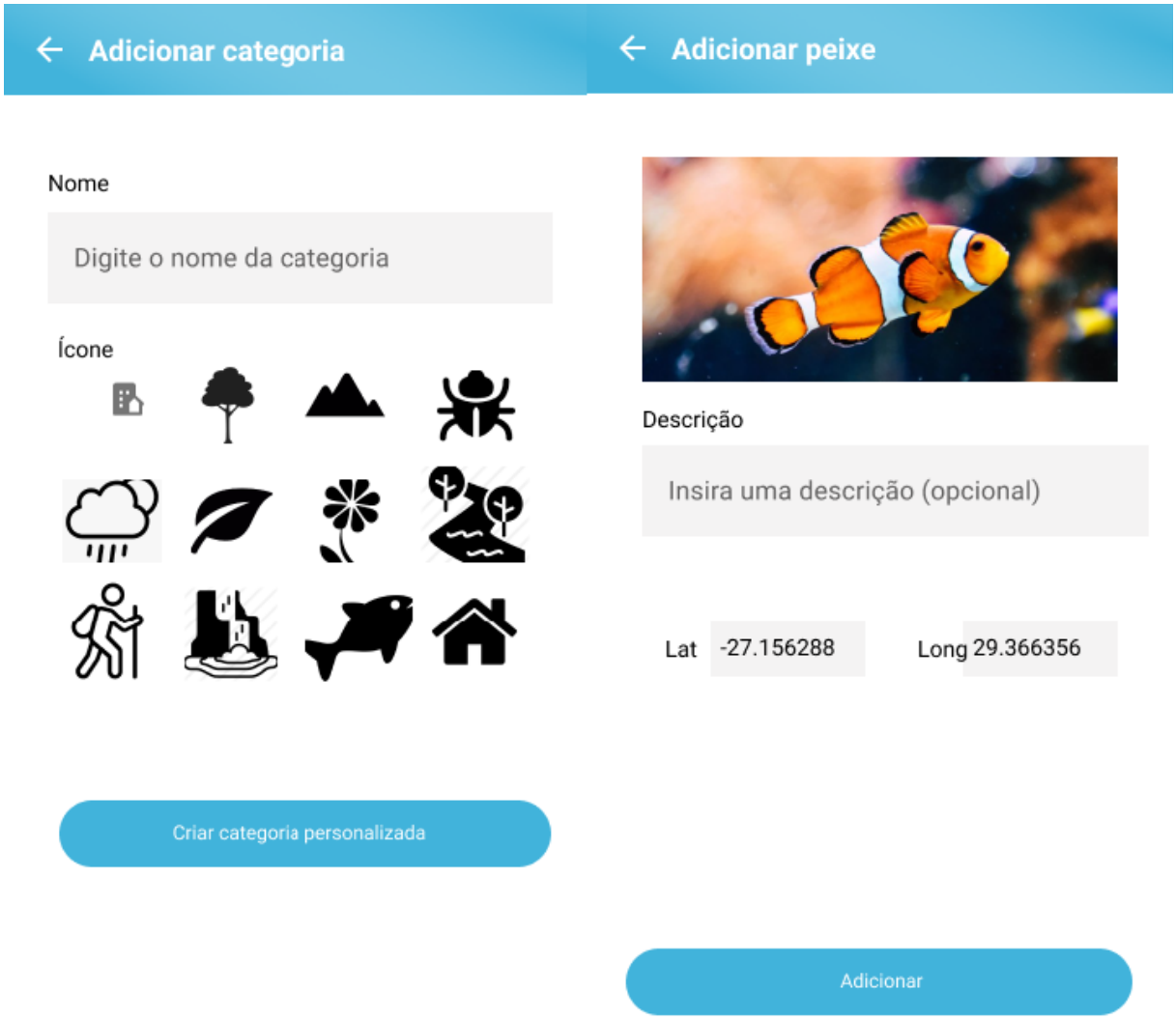
Fonte: Elaborado pelo autor

4.3.2 Inserção de objetos personalizados

Os objetos pré-definidos correspondem a elementos comuns do cotidiano e que são, em suas essências, os tipos de objeto que o aplicativo se propõe a registrar. Ainda assim, é possível que o usuário queira criar objetos personalizados que não se encaixem nos padrões. A Figura 4.6 mostra que é possível criar uma categoria personalizada.

É importante que o usuário tenha a liberdade limitada, pois objetos muito personalizados poderiam causar problemas na modelagem do banco de dados. Assim, só é possível dar um nome à categoria e escolher um ícone para a mesma. Depois, tira-se a foto para adicionar um objeto da nova categoria de objeto. Por fim, há um campo para adicionar uma breve descrição, e as coordenadas geográficas obrigatórias. O requisito RF05 é atendido por meio dessas telas.

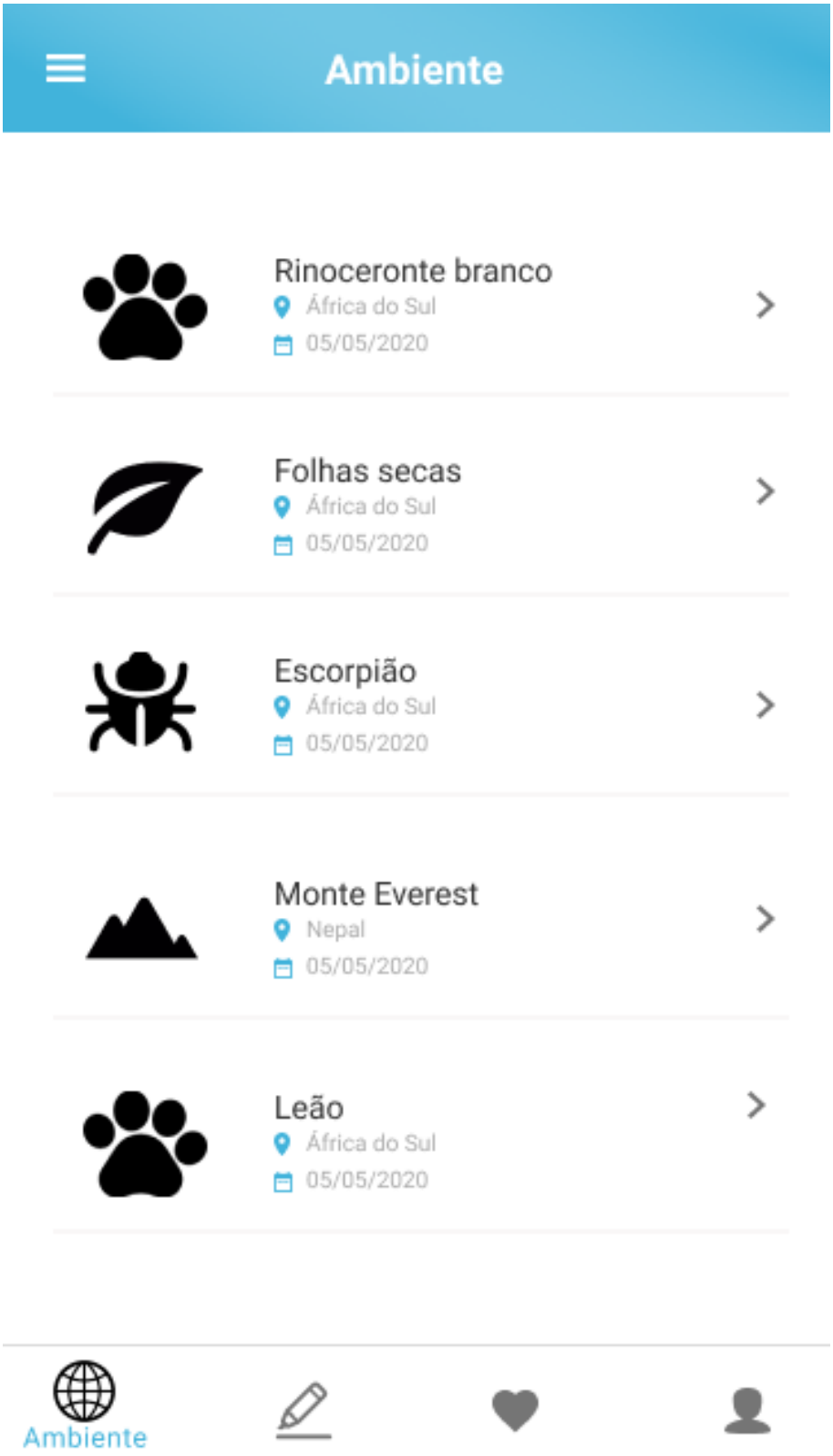
Figura 4.6 – Protótipo: Categoria personalizada.



Fonte: Elaborado pelo autor

Finalmente, quando um ou mais objetos são adicionados, eles podem ser vistos na lista de elementos de ambiente, na tela inicial, como ilustrado pela Figura 4.7. O ícone de sua categoria aparece na primeira coluna, em destaque. Na segunda coluna, temos o nome do objeto, sua localização, dessa vez nomeada por extenso, e a data de registro do objeto.

Figura 4.7 – Protótipo: Lista de elementos.



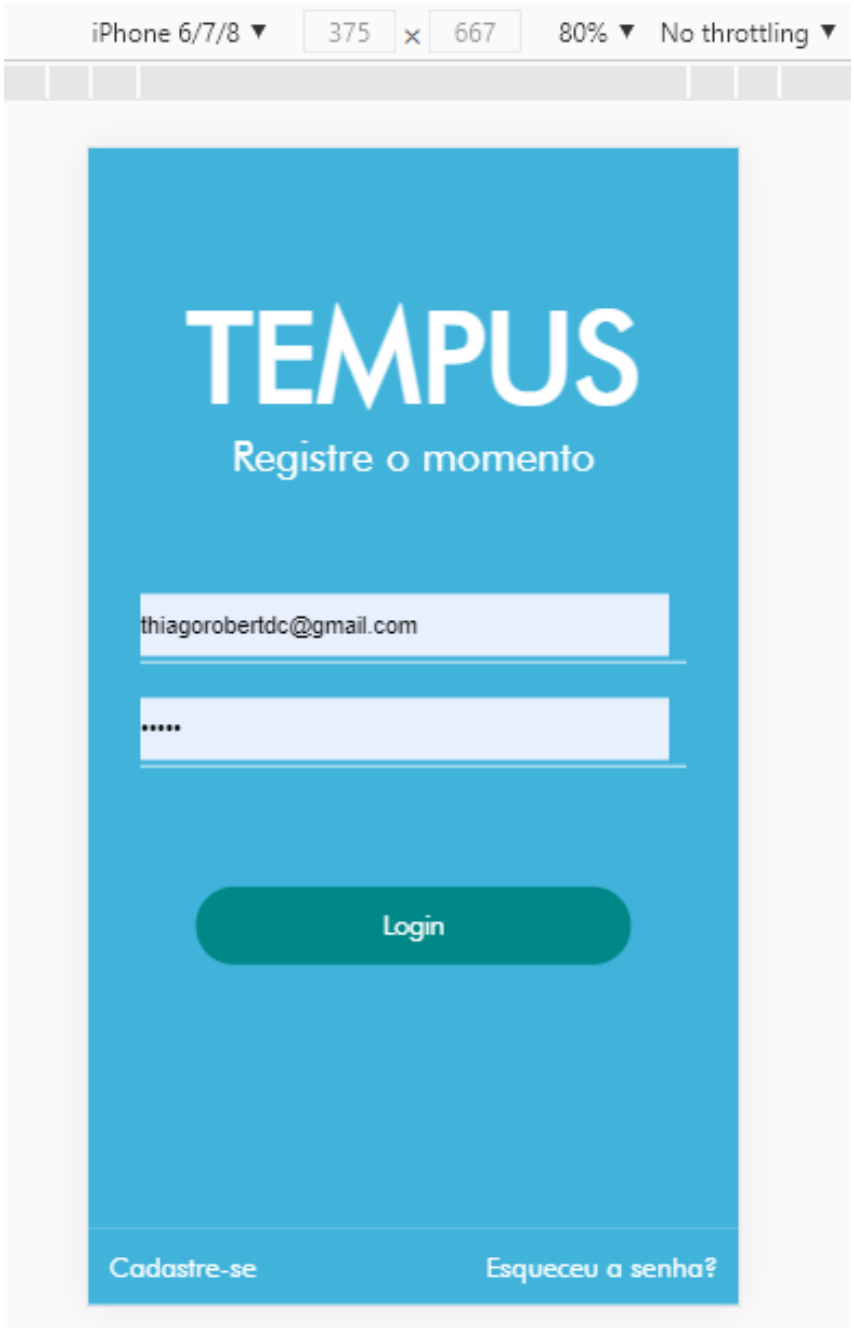
Fonte: Elaborado pelo autor

A seta na terceira coluna, alinhada ao final à direita, indica que o usuário pode visualizar o elemento com mais detalhes.

4.4 Desenvolvimento móvel

As Figuras a seguir ilustram todas as telas implementadas do aplicativo. Todas as telas seguiram os protótipos apresentados pela Sessão 4.3 e, aquelas que não possuiam protótipos prévios, seguiram os padrões propostos das telas anteriores. Algumas telas sofreram alterações em relação à sua correspondente no protótipo do Figma e, nestes casos, as alterações são justificadas.

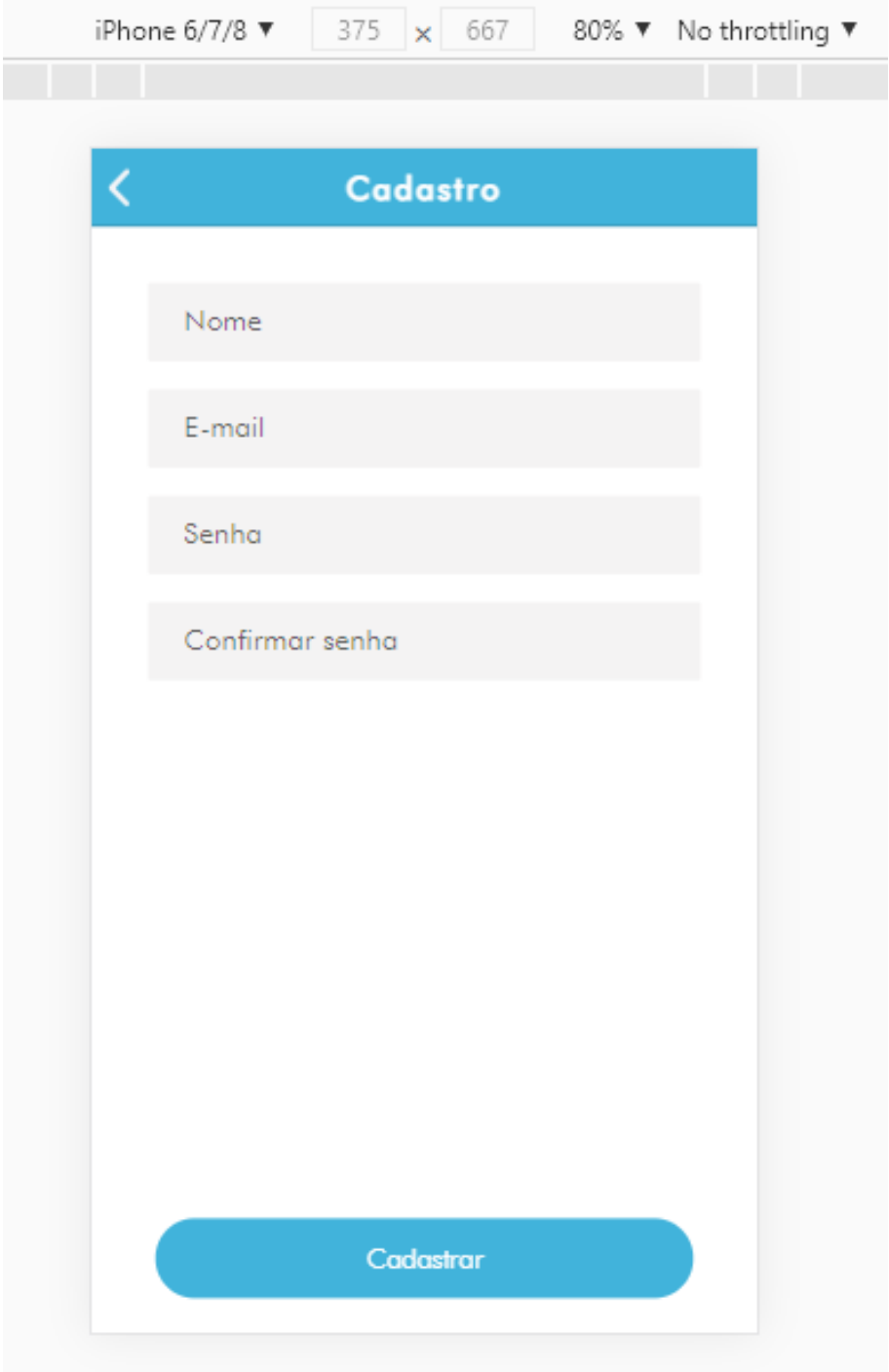
Figura 4.8 – Desenvolvimento: Tela de Login.



Fonte: Elaborado pelo autor

A tela de cadastro é mostrada pela Figura 4.9. Novamente, seguimos os padrões do protótipo, mas podemos notar que o cabeçalho está seguindo os padrões do iOS no botão de voltar à tela de Login. Entretanto, no protótipo o estilo segue os padrões do Android. Isso foi proposital para reafirmar a característica de multiplataforma do aplicativo.

Figura 4.9 – Desenvolvimento: Tela de Cadastro.

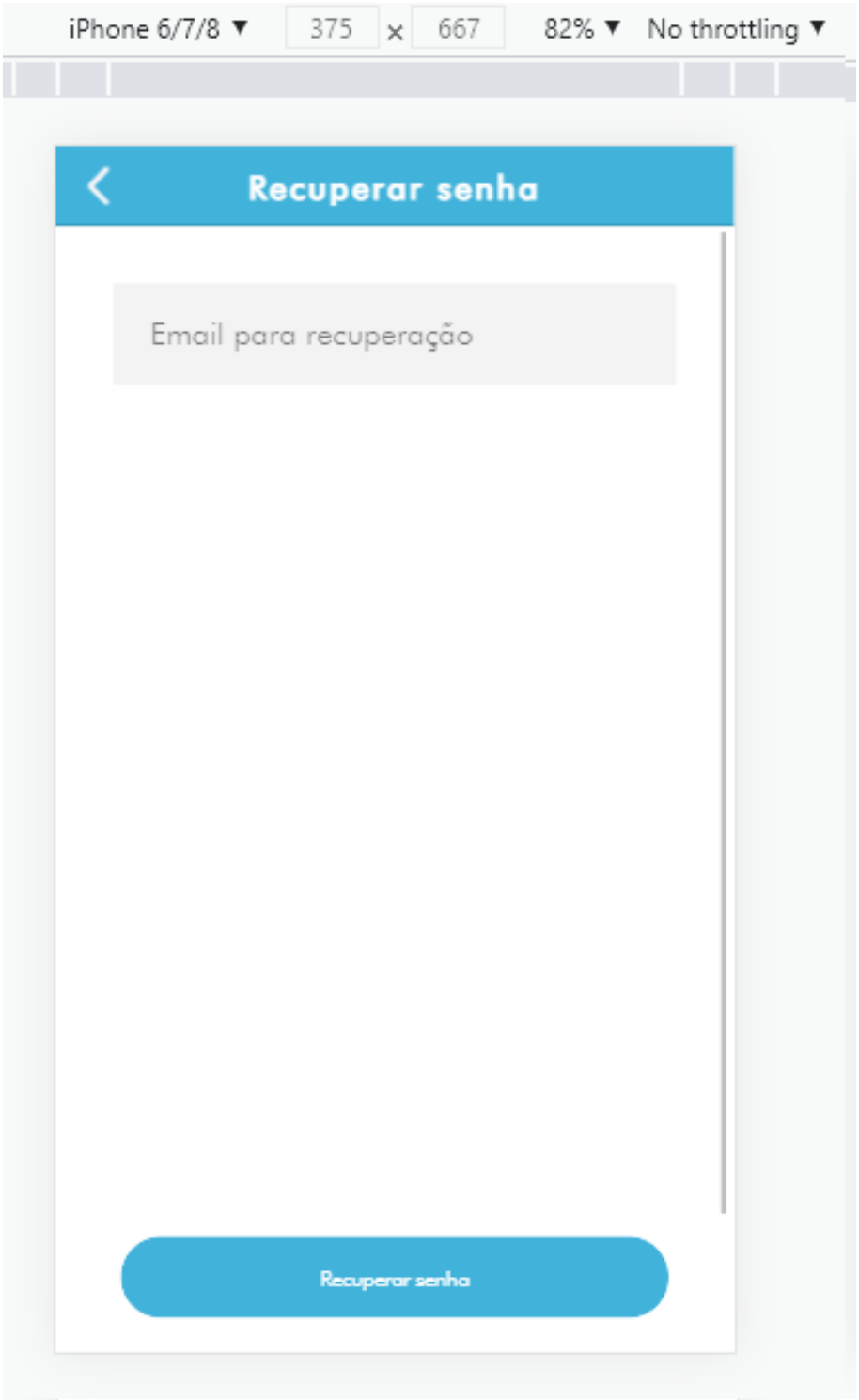


Fonte: Elaborado pelo autor

Usuários podem esquecer suas senhas e, para isso, é necessário ter uma função de recuperação de senha. Dessa forma, a Figura 4.10 mostra a tela de recuperação de senha, onde o

usuário fornece seu email e recebe um email com instruções para a criação de uma nova senha.

Figura 4.10 – Desenvolvimento: Recuperação de Senha.

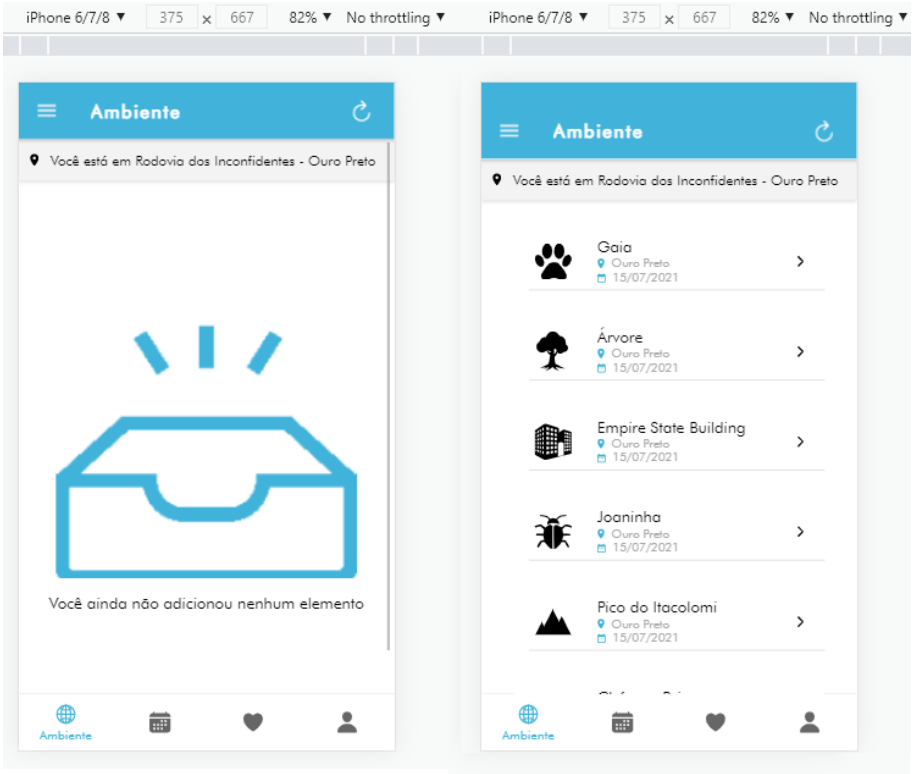


Fonte: Elaborado pelo autor

A tela inicial após a autenticação do usuário segue os padrões do protótipo, adicionando uma tag no topo da página informando ao usuário sua atual localização. A Figura 4.11 mostra a tela de ambiente quando o usuário se autentica e faz login no sistema. Temos duas possibilidades,

uma tela indicando que nenhum objeto foi adicionado, e uma lista de objetos registrados pelo usuário. Há um botão no canto superior direito, para atualizar a lista de objetos registrados quando desejado.

Figura 4.11 – Desenvolvimento: Tela de Ambiente - Vazio e Com Objetos Registrados.



Fonte: Elaborado pelo autor

A Figura 4.12 mostra a tela de menu aberta. Nela, temos as opções padrões de objetos possíveis de serem registrados pelo usuário. Além disso, todas as categorias personalizadas pelo usuário são mostradas na seção "Personalizadas por você", única para cada usuário. A partir desse menu, o usuário pode escolher qual tipo de objeto quer adicionar.

O botão "Criar categoria personalizada"é autoexplicativo e fornece mais autonomia do usuário em relação ao sistema. É possível escolher um ícone personalizado para a categoria e dar um nome, como nos mostra a Figura 4.13.

Figura 4.12 – Desenvolvimento: Menu.

Fonte: Elaborado pelo autor

Figura 4.13 – Desenvolvimento: Adicionar Categoria Personalizada.

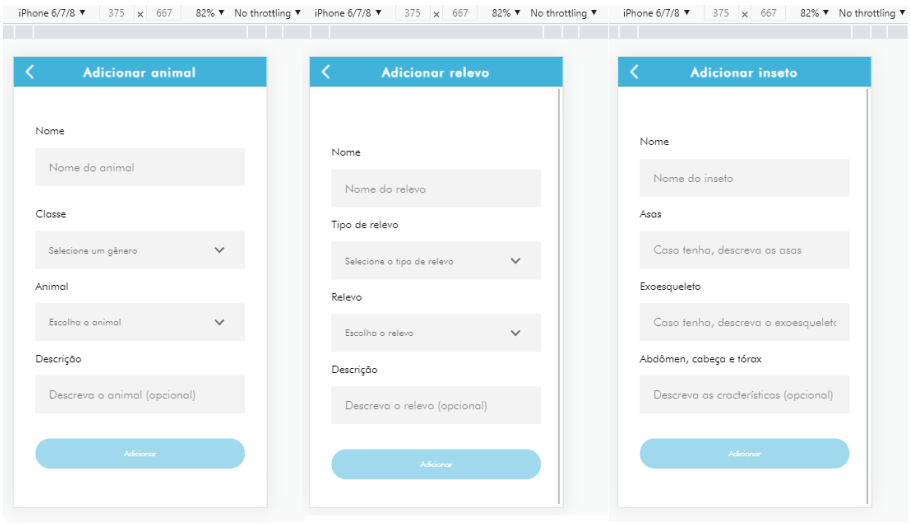
Fonte: Elaborado pelo autor

O protótipo fornecia a opção de adicionar seis categorias padrões, sendo uma delas fenômenos meteorológicos. Essa categoria foi removida por não ser tão relevante de ser registrada. Assim, é possível através do aplicativo registrar cinco categorias padrões, sendo elas Animais,

Insetos, Árvores, Edifícios e Relevos. Ainda, é possível adicionar objetos de categorias personalizadas. As Figuras mostram os modais de adição destes objetos. Como é um desenvolvimento feito em computador, a foto tirada pelo usuário não aparece, por não ser possível tirar uma foto com o computador através do Ionic sem ser com a *webcam*.

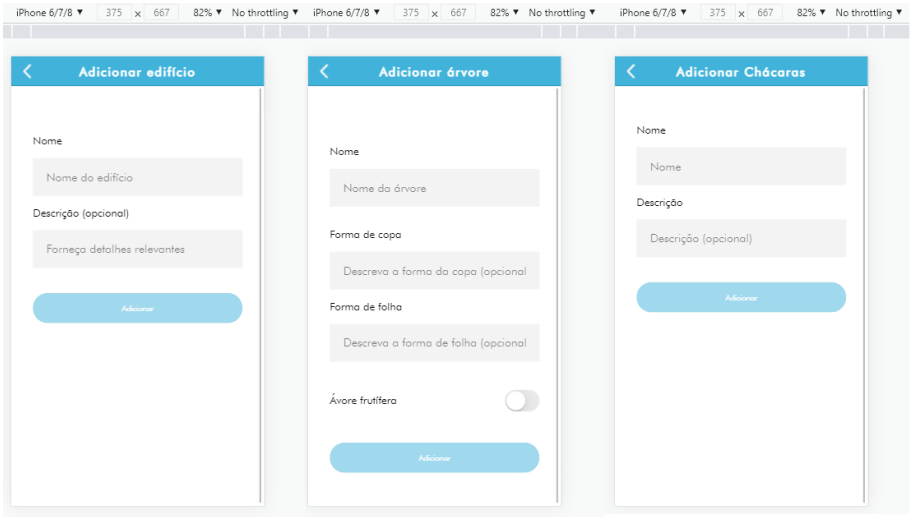
As coordenadas geográficas que poderiam ser fornecidas pelo usuário foram removidas, pois isso poderia afetar a qualidade dos dados coletados, possibilitando a existência de documentações inverídicas e/ou inexatas. As coordenadas são fornecidas automaticamente pelos controladores, sem qualquer interferência do usuário.

Figura 4.14 – Desenvolvimento: Adicionar objetos.



Fonte: Elaborado pelo autor

Figura 4.15 – Desenvolvimento: Adicionar objetos.

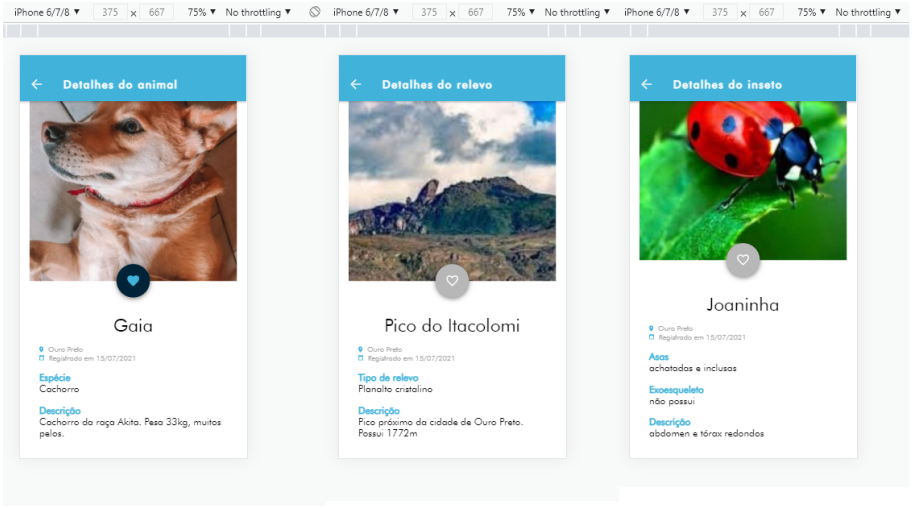


Fonte: Elaborado pelo autor

Os objetos registrados em Tempus possuem uma tela onde seus detalhes, assim como as fotos registradas podem ser vistos. Os cabeçalhos das telas indicam qual o tipo do objeto está

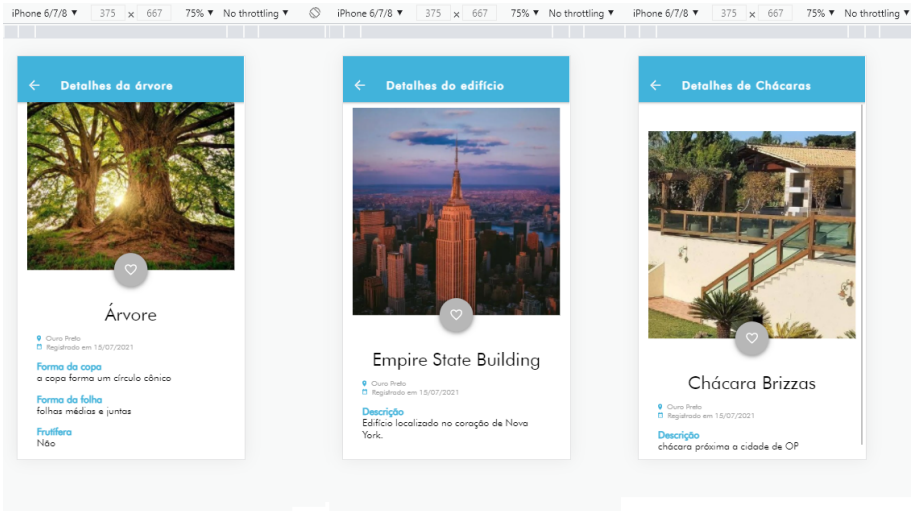
sendo detalhado, e o corpo da página trás as demais informações. É possível favoritar os objetos clicando no coração branco dentro do círculo cinza que fica centralizado na parte inferior da foto. Uma vez clicado, o coração se torna azul claro e seu círculo em volta se preenche de azul escuro. As Figuras 4.16 e 4.17 ilustram esse cenário.

Figura 4.16 – Desenvolvimento: Detalhes de objetos.



Fonte: Elaborado pelo autor

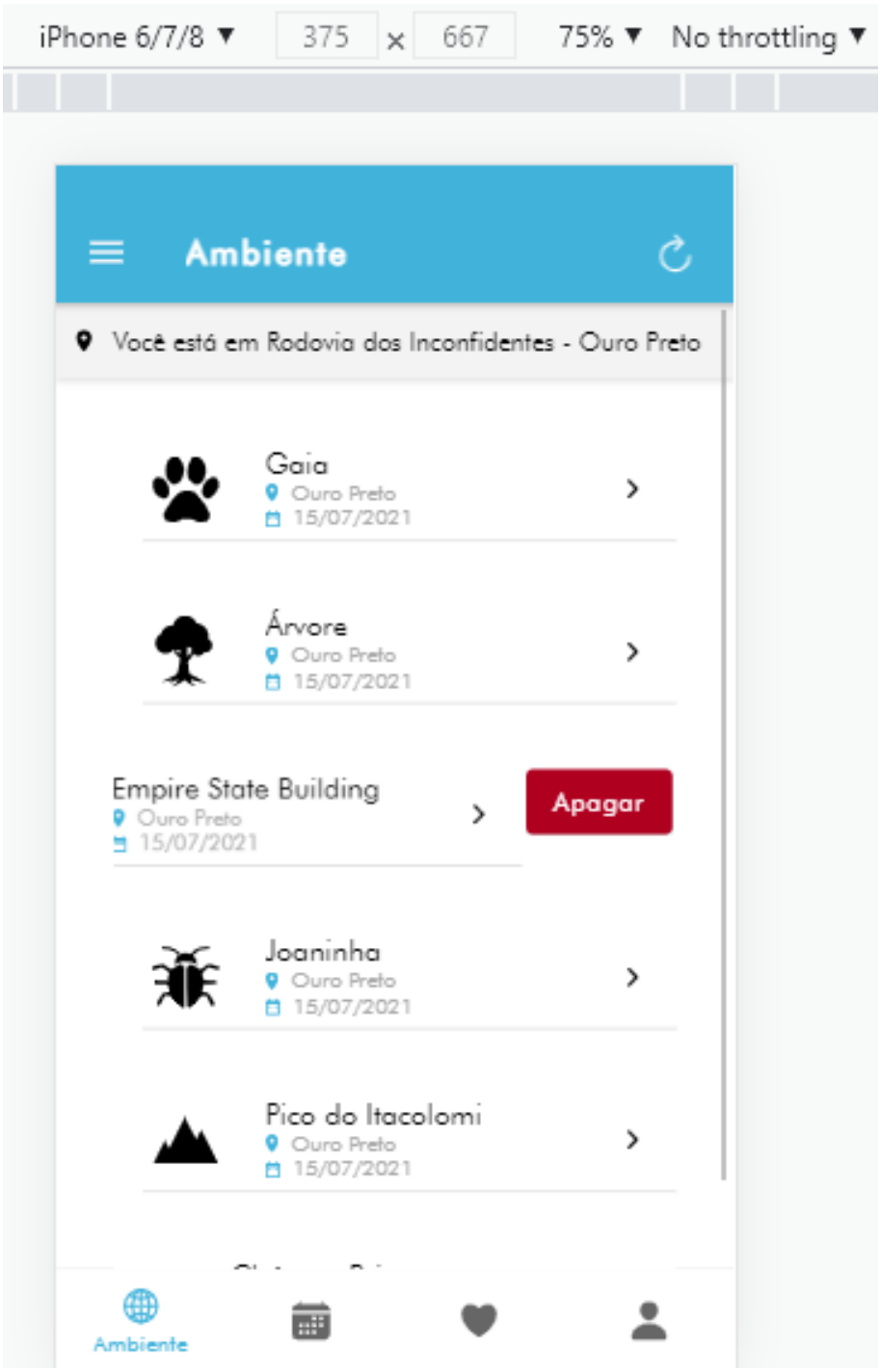
Figura 4.17 – Desenvolvimento: Detalhes de objetos.



Fonte: Elaborado pelo autor

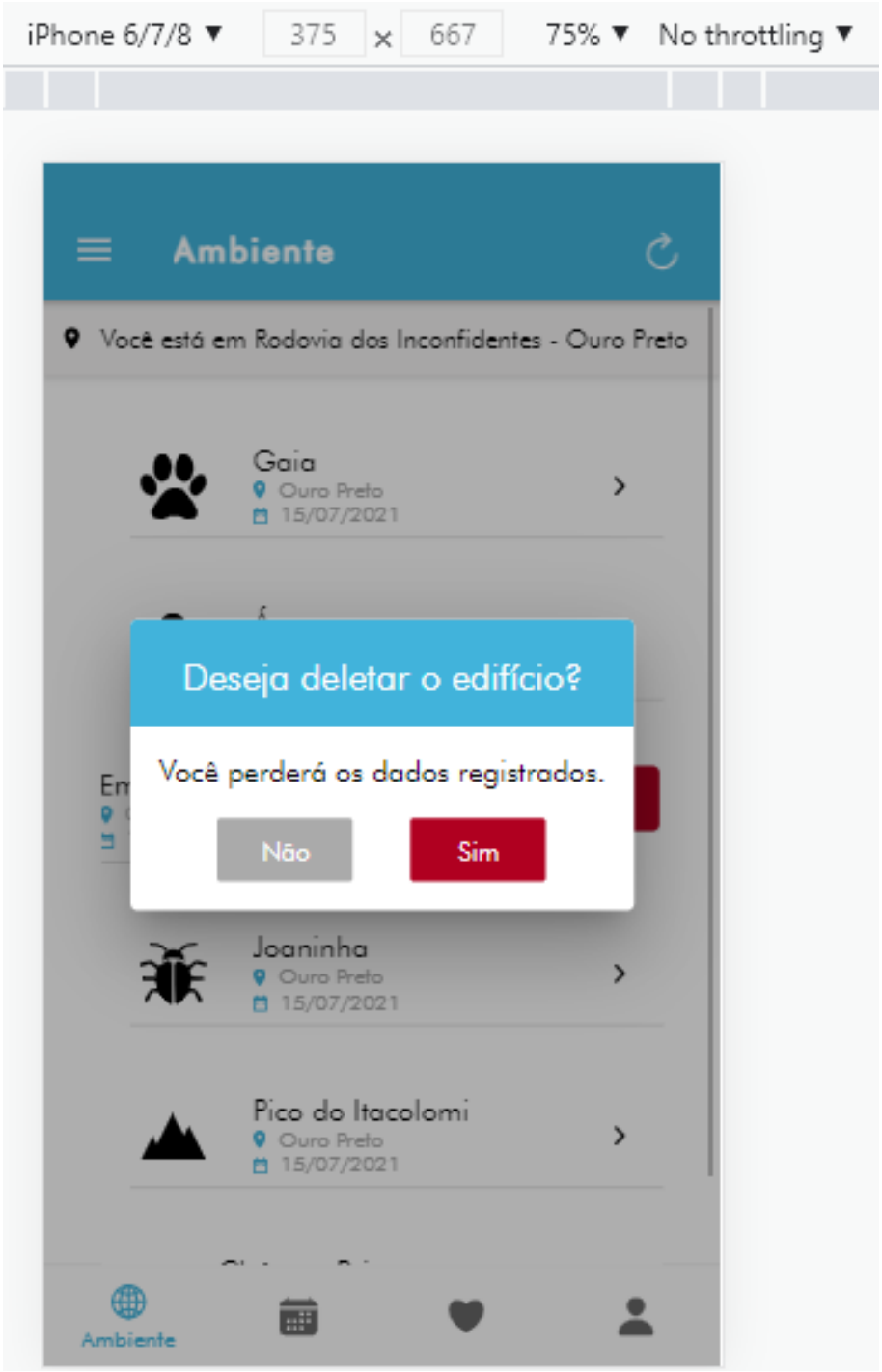
Os objetos registrados podem ser excluídos caso desejado. O usuário precisa arrastar o mesmo para a esquerda, na listagem de variáveis de ambiente e clicar em apagar e, depois, confirmar o aviso de exclusão, como mostram as Figuras 4.18 e 4.19.

Figura 4.18 – Desenvolvimento: Exclusão.



Fonte: Elaborado pelo autor

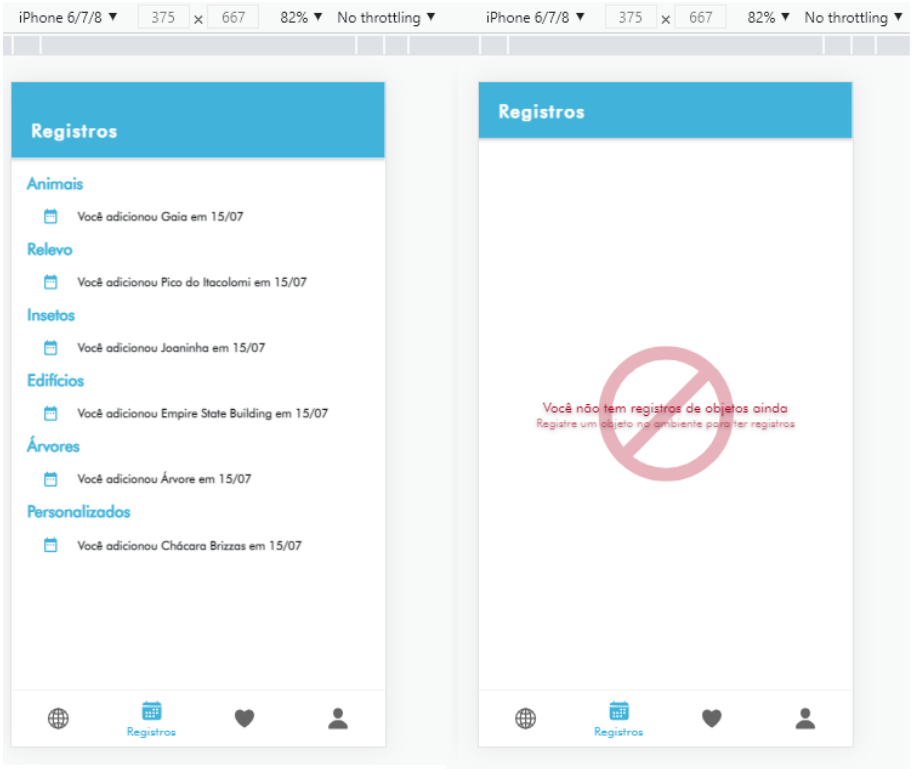
Figura 4.19 – Desenvolvimento: Aviso.



Fonte: Elaborado pelo autor

A tela que fornece o histórico de registros é ilustrada pela Figura 4.20. As duas possibilidades são uma listagem de registros vazia, quando o usuário é novo no sistema. A outra é uma listagem de registros de acordo com a categoria dos objetos, fornecendo o nome e a data em que o objeto foi registrado.

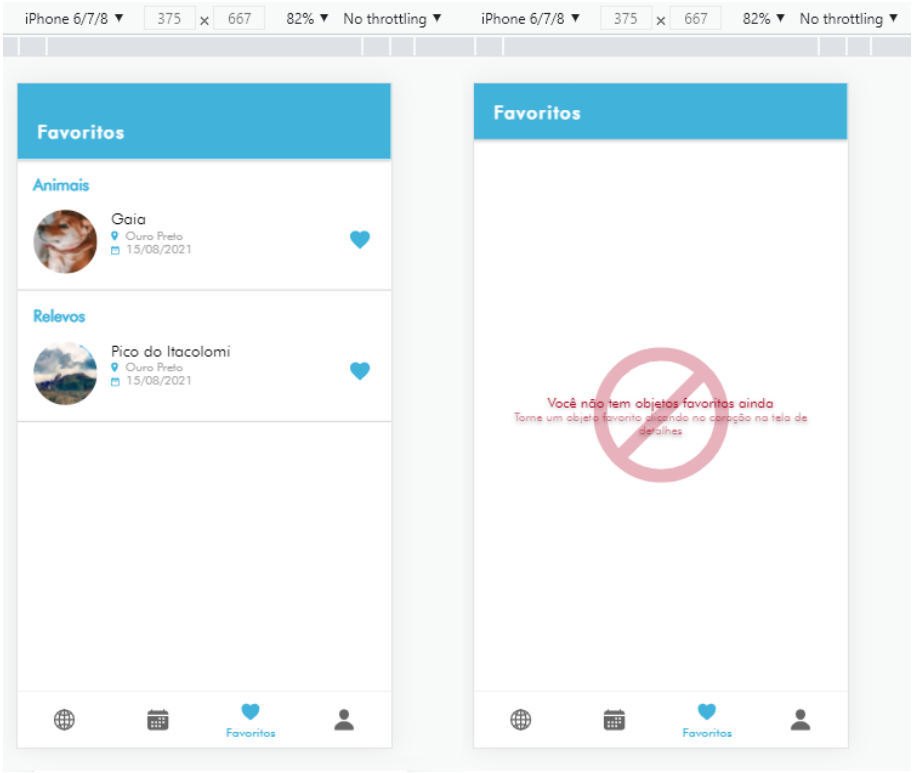
Figura 4.20 – Desenvolvimento: Registros.



Fonte: Elaborado pelo autor

Objetos podem ser favoritados para serem encontrados mais rapidamente pelo usuário. Para isso temos a seção de Favoritos, que nos mostra uma listagem de objetos favoritos ou uma tela indicando que nenhum objeto ainda foi favoritado pelo usuário. Podemos ver as duas situações na Figura 4.21.

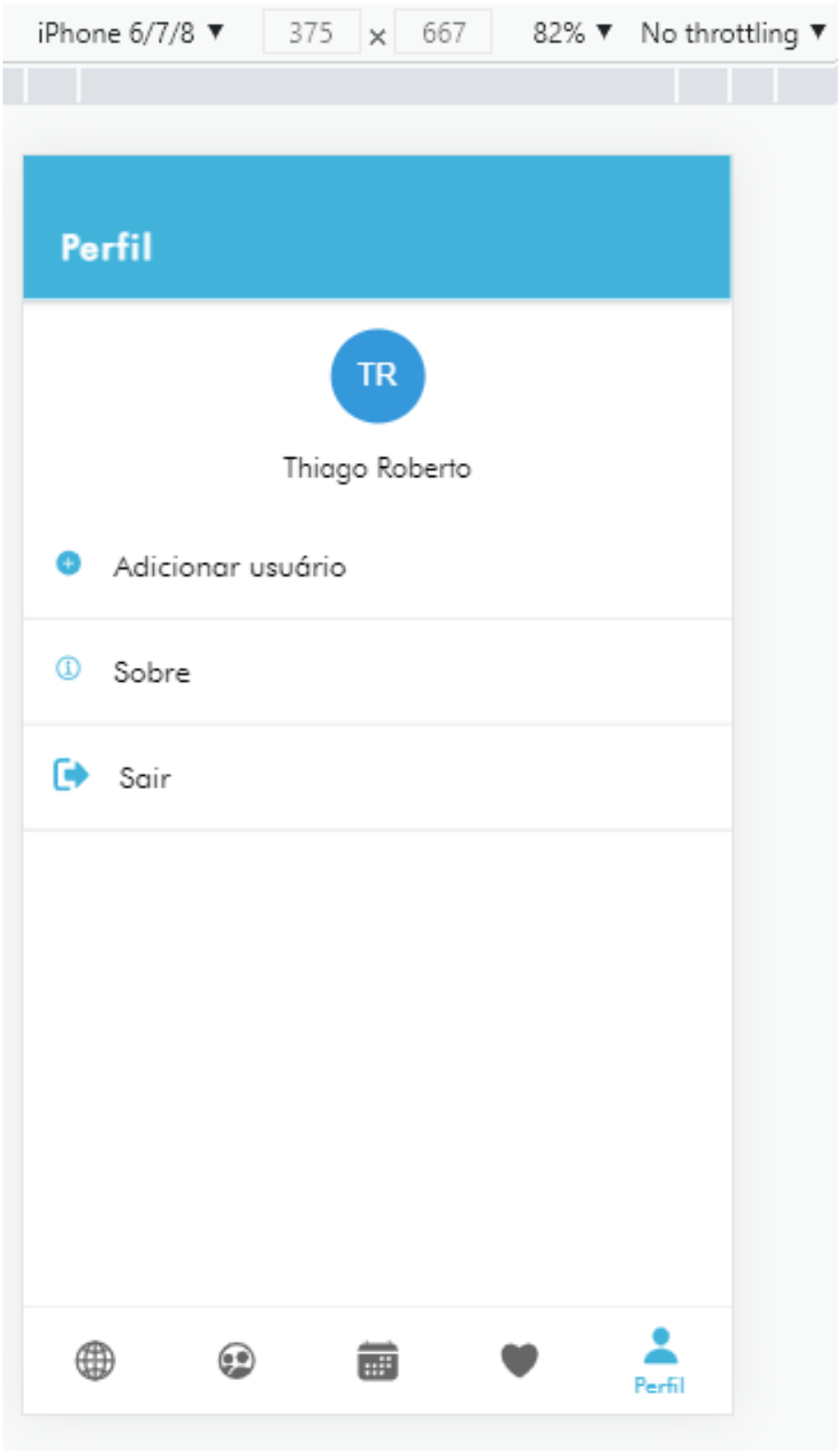
Figura 4.21 – Desenvolvimento: Favoritos.



Fonte: Elaborado pelo autor

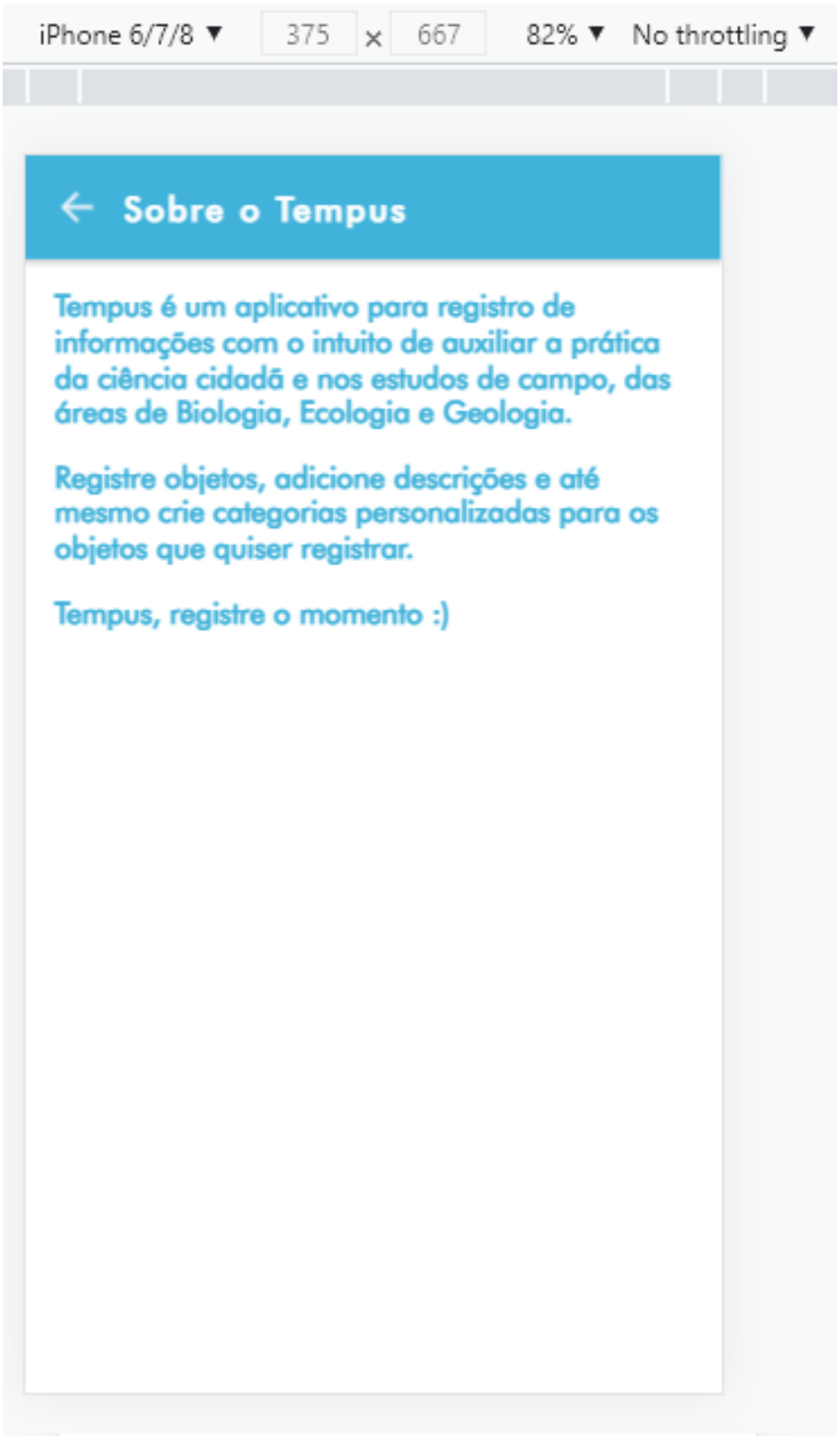
A tela de Perfil possui informações breves. Um avatar circular no topo mostra as iniciais capitalizadas do usuário, com seu nome por extenso embaixo. Há três opções de ação a partir dessa tela: realizar o *logout* do sistema, ler algumas informações sobre o sistema ou adicionar um usuário em nosso ambiente social. A primeira ação nos leva de volta à tela de *login*, enquanto a segunda nos leva para uma tela explicando um pouco sobre o aplicativo e o seu propósito. A opção de adicionar um usuário nos permite inserir um novo usuário ao nosso ambiente social, tornando possível a visualização dos objetos coletados por este. A Figura 4.22 mostra a tela de perfil, enquanto a Figura 4.23 mostra a tela de informações. A tela de adição de usuários, assim como a listagem dos usuários adicionados é mostrada pela Figura 4.24.

Figura 4.22 – Desenvolvimento: Perfil.



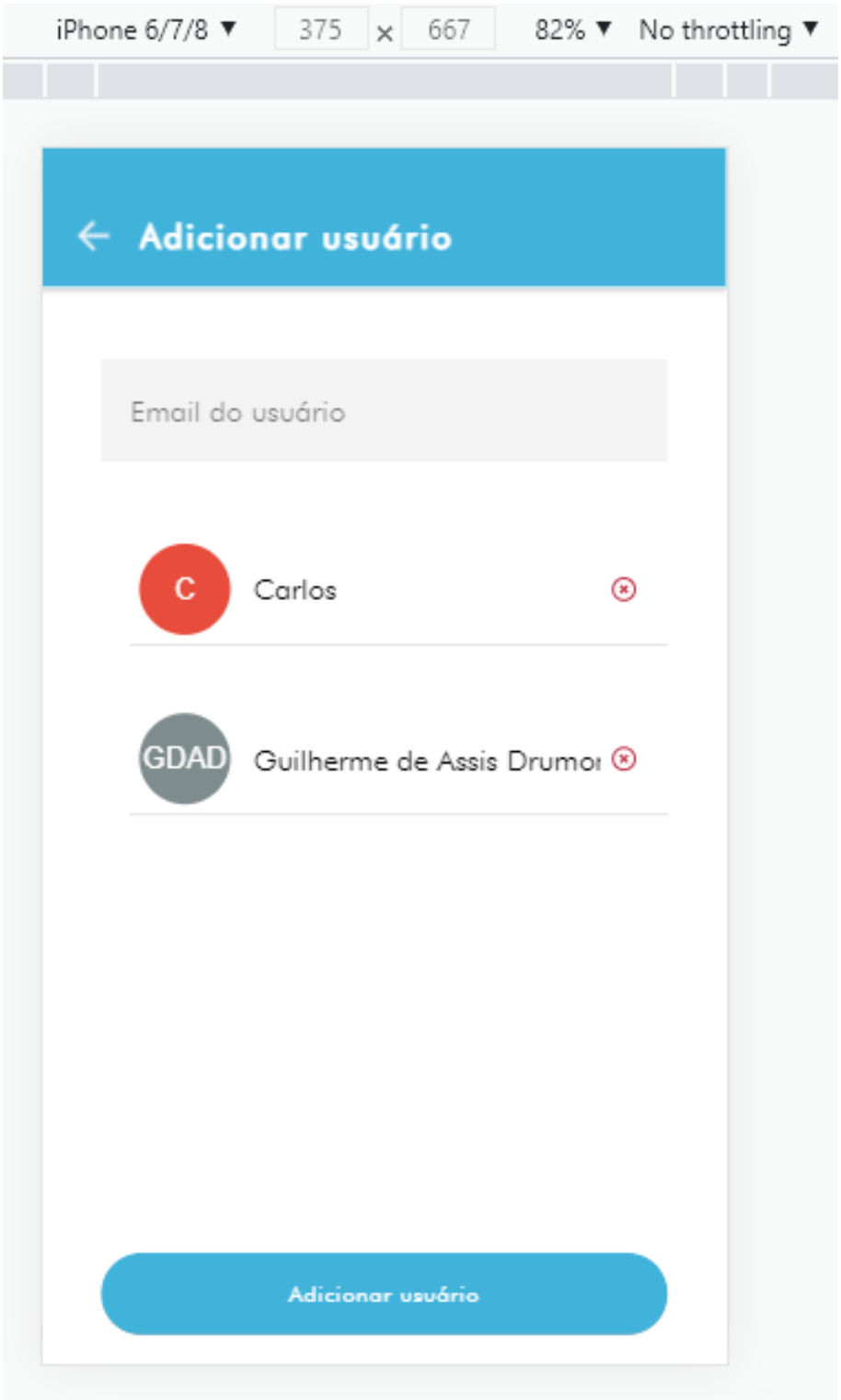
Fonte: Elaborado pelo autor

Figura 4.23 – Desenvolvimento: Sobre.



Fonte: Elaborado pelo autor

Figura 4.24 – Desenvolvimento: Adicionar usuários.

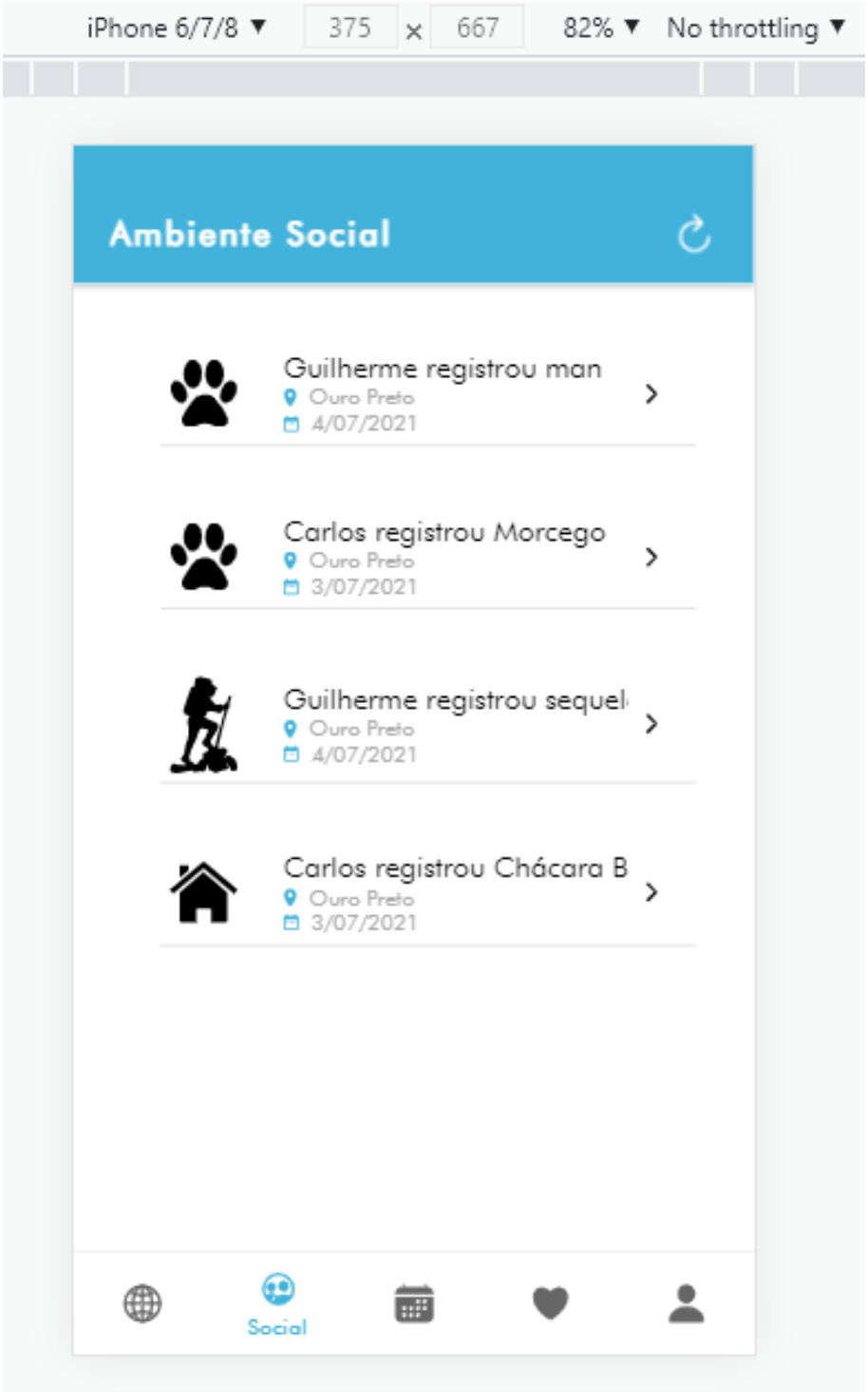


Fonte: Elaborado pelo autor

A tela de Ambiente Social possui informações de objetos adicionados por outros usuários. Assim como na tela de Ambiente, os objetos são listados mostrando um ícone personalizado e informações básicas. Um clique no objeto também leva à uma tela de detalhes. Essa funcionalidade deixa o aplicativo mais interessante, promovendo uma coleta de dados compartilhada. A Figura

4.25 ilustra o ambiente de dados compartilhados.

Figura 4.25 – Desenvolvimento: Ambiente Social.



Fonte: Elaborado pelo autor

4.5 Testes

Após o desenvolvimento completo do aplicativo, foram realizados dois tipos de testes que visam garantir a qualidade do software e uma boa usabilidade ao usuário, respectivamente. Essa

Seção discorre sobre as definições e implementações. Os resultados obtidos através da aplicação destes testes são descritos no Capítulo 5.

4.5.1 Testes automatizados

Srivastava, Kumar e Singh (2021) definem a testagem automatizada como uma parte crucial do desenvolvimento de *software*, pois garantem alta qualidade e que está livre de bugs e defeitos. O teste de *software* consiste na execução de tratamentos incorretos de códigos de entrada e de estados escolhido para revelar bugs.

Dessa forma, testes automatizados foram escolhidos para garantir a qualidade do *software* Tempus, tratando alguns casos de dados de entrada e chamadas de funções. A ferramenta escolhida foi o *test runner* Karma.

O principal objetivo do Karma é automatizar os testes em diversos navegadores web com um único comando. Mesmo ele tendo sido criado para o Angular, atualmente ele é usado em outros frameworks JavaScript. O Karma suporta diversos tipos de testes, como: unitários, integração, E2E.

Com o Karma, foram escritos testes unitários que trataram de dados de entrada e de chamadas de funções.

A estrutura de um teste é ilustrada pela Figura 4.26. Ele se inicia com a palavra chave *it*, que abre a função do teste. O primeiro parâmetro é o texto que descreve a ação que o teste deve fazer e normalmente se inicia com a palavra *should*, em inglês, que significa "deve".

Figura 4.26 – Desenvolvimento: Teste.

```
Run | Debug | Show in Test Explorer
it('should correctly render the passed name Input value', () => {
  const name = faker.name.findName();
  fixture.debugElement.query(By.css('#name')).nativeElement.value = name;
  fixture.debugElement.query(By.css('#name')).nativeElement.dispatchEvent(new Event('input'));
  fixture.detectChanges();
  fixture.whenStable().then(() => {
    expect(component.form.get('name').value).toBe(name); // this pass
  });
});
```

Fonte: Elaborado pelo autor

A biblioteca *faker* foi utilizada para gerar dados falsos, simulando os dados sendo inseridos por um usuário. Os campos de inserção são procurados via *queries* em HTML e, ao encontrarmos os mesmos, inserimos os dados através de um evento de inserção. Por fim, ao atualizarmos o estado do componente, verificamos que o valor inserido se encontra no formulário e, assim, validamos a inserção de um dado, através da função *expect*, palavra em inglês para "expectativa". Assim, esperamos que algum valor se encontre no formulário e que seja igual ao dado falso criado no início do teste.

Se um dado não for inserido e o usuário tentar prosseguir, ou seja, despachar o formulário para validação, renderizamos um aviso solicitando que aquele campo seja fornecido. A Figura 4.27 descreve essa situação.

Figura 4.27 – Desenvolvimento: Teste.

```
Run | Debug | Show in Test Explorer
it('should correctly render the Password error message when Password is empty', () => {
  fixture.debugElement.query(By.css('#password')).nativeElement.value = '';
  fixture.debugElement.query(By.css('#password')).nativeElement.dispatchEvent(new Event('input'));
  fixture.detectChanges();
  fixture.whenStable().then(() => {
    expect(fixture.debugElement.query(By.css('#password-error')).nativeElement.value).toBe('Preencha sua senha'); // this pass
  });
});
```

Fonte: Elaborado pelo autor

Da mesma maneira, iniciamos o teste com *it*, escrevemos a asserção a ser feita e, ao final, esperamos que se encontre renderizado na tela um aviso solicitando ao usuário que preencha o campo que falta.

As chamadas de funções são testadas da mesma maneira, com o diferencial que precisamos criar um *spy* - espião em inglês - para simular as chamadas de funções dos pacotes do Angular. A Figura 4.28 mostra dois testes de chamada de função com pacotes do Angular, chamando o serviço do Parse e o serviço de navegação, respectivamente.

Figura 4.28 – Desenvolvimento: Teste.

```
Run | Debug | Show in Test Explorer
it("should call parse Login Service on button click", () => {
  let parse = fixture.debugElement.injector.get(ParseService);
  spyOn(parse, "logIn");

  de = fixture.debugElement.query(By.css("ion-button"));

  de.triggerEventHandler("click", null);

  expect(parse.logIn).toHaveBeenCalled();
});

Run | Debug | Show in Test Explorer
it("should call Navigation on register tag click", () => {
  let navCtrl = fixture.debugElement.injector.get(NavController);
  spyOn(navCtrl, "navigateForward");

  de = fixture.debugElement.query(By.css("#register-tag"));

  de.triggerEventHandler("click", null);

  expect(navCtrl.navigateForward).toHaveBeenCalled();
});
```

Fonte: Elaborado pelo autor

As funções que não fazem parte do pacote do Angular, mas que formam funções próprias através de combinações das funções do pacote, também tem suas chamadas testadas através de *spys*, como mostra a Figura 4.29.

Figura 4.29 – Desenvolvimento: Teste.

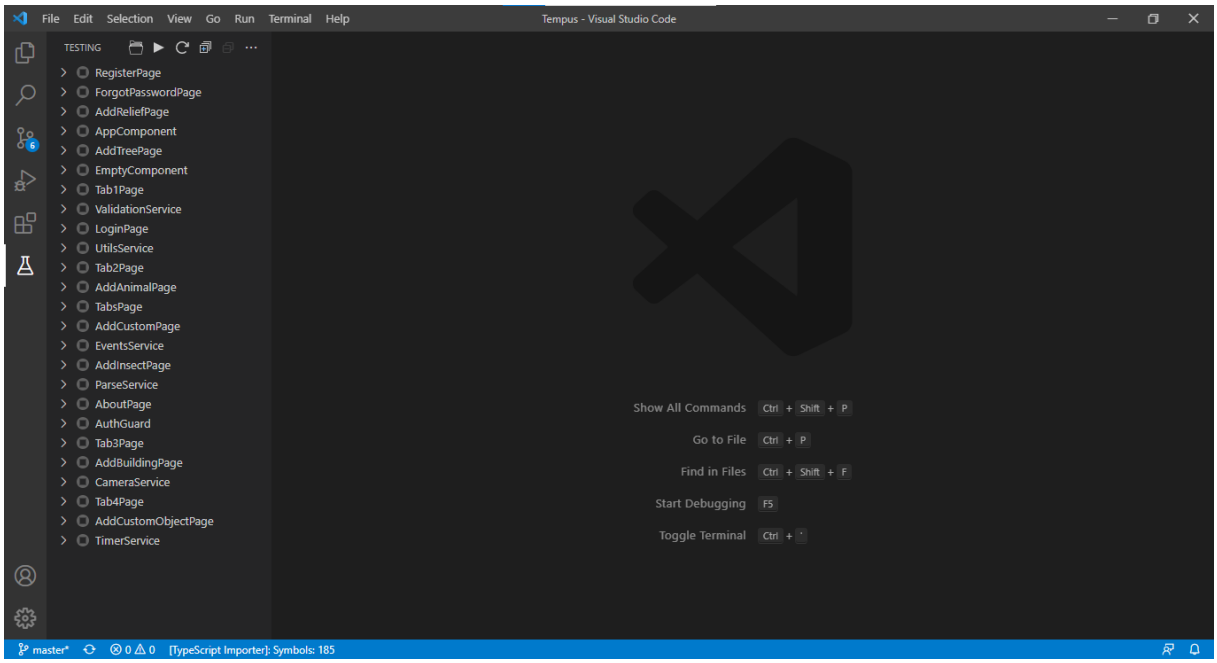
```
Run | Debug | Show in Test Explorer
it("should call parse SaveItem on button click", () => {
  let addAnimal = fixture.debugElement.injector.get(AddAnimalPage);
  spyOn(addAnimal, "addAnimal");
  const name = faker.name.findName();
  fixture.debugElement.query(By.css('#name')).nativeElement.value = name;
  fixture.debugElement.query(By.css('#name')).nativeElement.dispatchEvent(new Event('input'));


  const description = faker.name.findName();
  fixture.debugElement.query(By.css('#description')).nativeElement.value = description;
  fixture.debugElement.query(By.css('#description')).nativeElement.dispatchEvent(new Event('input'));
  fixture.detectChanges();
  fixture.whenStable().then(() => {
    de = fixture.debugElement.query(By.css("#add-animal"));
    de.triggerEventHandler("click", null);
    expect(addAnimal.addAnimal).toHaveBeenCalled();
  });
});
});
```

Fonte: Elaborado pelo autor

O ambiente de desenvolvimento do Visual Studio Code, editor de código fonte que contém um terminal integrado além de suporte à extensões personalizadas que permitem complementação inteligente de código, por exemplo, fornece uma interface visual para rodar os testes com uma interface amigável e intuitiva, chamada de laboratório de testes. Todos os testes são listados em uma coluna à esquerda, com a opção de rodar todos de uma vez ou escolher um bloco para ser rodado. Cada bloco corresponde a uma tela do aplicativo. A Figura 4.30 é uma imagem desse laboratório de testes e de sua estrutura.

Figura 4.30 – Desenvolvimento: Laboratório de Testes.



Fonte: Elaborado pelo autor

4.5.2 Testes de usabilidade - método SUS

Esta Seção detalha os testes de usabilidade realizado para a aplicação. A Subseção 4.5.2.1 descreve o método escolhido e a Subseção 4.5.2.2 descreve o processo de avaliação adotado.

4.5.2.1 Método

O teste de código manual é a técnica mais primitiva de todas as variedades de teste e ajuda a encontrar bugs no aplicativo de código (SRIVASTAVA; KUMAR; SINGH, 2021). Dessa forma, testes voltados para a usabilidade do aplicativo foram adotados como forma de avaliar a qualidade de Tempus enquanto software.

O *System Usability Scale* (SUS) é um questionário padronizado amplamente utilizado para a avaliação da usabilidade (LEWIS, 2018). As citações do Google Scholar (examinadas em 13/03/2018) mostraram 5.664 citações para o artigo que introduziu o SUS pela primeira vez, segundo Lewis (2018).

Em sua forma padrão (mais usada), o SUS tem dez itens de cinco pontos com alternância de itens positivos e itens negativos. As pontuações vão de 1 para "Discordo fortemente"e 5 para "Concordo fortemente". Além disso, as questões de números ímpares possuem um tom positivo, enquanto as questões de números pares possuem afirmações negativas sobre o uso da aplicação.

Para avaliarmos os resultados, precisamos converter para uma escala de 0 (menor nota) a 10 (maior nota), as notas dadas pelos usuários em cada uma das perguntas. Nas respostas ímpares, subtraímos um ponto à resposta dada pelo usuário. Nas respostas pares, retiramos de 5 a resposta fornecida pelo usuário e, assim, teremos respostas entre 0 e 4. Após esse cálculo, somamos as pontuações das 10 questões e multiplicamos por 2,5 o que refletirá numa pontuação que pode ir de 0 a 100 (em vez do 0 a 40, caso não multiplicássemos por 2,5). O valor obtido será a pontuação individual na escala do SUS e, no caso de termos mais de um usuário, fazemos uma média das pontuações de cada um, obtendo um valor final. De acordo com Lewis (2018), o ideal é que esse valor esteja acima de 68, valor encontrado como média nos experimentos utilizados para criar a escala.

4.5.2.2 Processo

Os convites para testar a aplicação foram feitos via redes sociais de mensagem de texto, tendo adesão de 12 usuários. Antes de responder às perguntas, os usuários foram convidados a instalar uma versão do aplicativo em seus dispositivos móveis e a seguir o seguinte roteiro:

1. Criar sua conta
2. Adicionar 1 imagem capturada da categoria Animal
3. Adicionar, 1 categoria de sua escolha, tirar sua foto e preencher seus dados

4. Criar uma categoria personalizada
5. Adicionar 1 imagem da categoria personalizada
6. Visualizar detalhes específicos das capturas criadas
7. Ao entrar nos detalhes específicos da captura, torná-la como favorita
8. Apagar um objeto da lista de objetos
9. Visualizar as datas de registro dos objetos criados
10. Visualizar seus objetos favoritos
11. Visualizar informações sobre o App
12. Realizar logout do sistema

O roteiro foi montado seguindo as instruções da Tabela 4.3:

Tabela 4.3 – Testes de usabilidade

Atividade	**Tarefa**
Preparação	• definir tarefas para os participantes executarem • definir o perfil dos participantes e recrutá-los • preparar material para observar e registrar o uso • executar um teste-piloto
Coleta de dados	• observar e registrar a performance e a opinião dos participantes durante durante sessões de uso controladas
Interpretação e consolidação dos resultados	• reunir, contabilizar e sumarizar os dados coletados dos participantes
Relato dos resultados	• relatar a performance e a opinião dos participantes

Fonte: Disponível em https://bit.ly/3eHUGxs

A tarefa 1 garantiu o acesso de todos os usuários ao sistema, sendo redirecionados à tela principal após o cadastro. Sem conhecimento dos usuários, foram cronometrados os tempos gastos em cada uma das tarefas do roteiro, com objetivo de obter-se uma métrica além dos testes automatizados e a escala SUS: o tempo. Através da medição do tempo também podemos fazer aferições sobre a usabilidade da aplicação.

Após concluir todo o roteiro, os usuários foram orientados a responder o formulário SUS através da plataforma do Google Formulários.

A aplicação do método SUS se deu por meio do Google Formulários, e as 10 perguntas foram as seguintes:

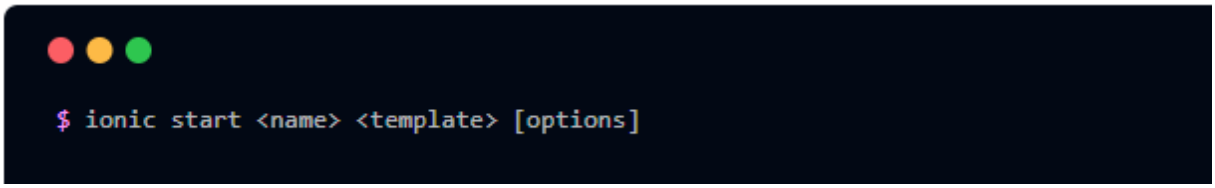
1. Eu acho que gostaria de usar este sistema com frequência
2. Eu acho este sistema desnecessariamente complexo
3. Eu achei o sistema fácil de usar
4. Eu acho que precisaria do suporte de um técnico para poder usar o sistema

5. Eu acho que as várias funções do sistema estão bem integradas
6. Eu acho que o sistema apresenta muitas inconsistências
7. Eu imagino que as pessoas aprenderão a usar este sistema rapidamente
8. Eu achei que o sistema era complicado de usar
9. Eu senti-me confiante a utilizar o sistema
10. Eu irei necessitar de aprender muitas coisas novas antes de conseguir utilizar o sistema.

4.6 Estrutura do Projeto

A criação do projeto inicial foi feita utilizando uma gama de ferramentas. O framework Ionic foi a ferramenta escolhida para dar início ao aplicativo multiplataforma. A Figura 4.31 mostra a utilização do comando de inicialização, que resulta no esqueleto do projeto pronto com uma página inicial.

Figura 4.31 – Estrutura: Inicialização.



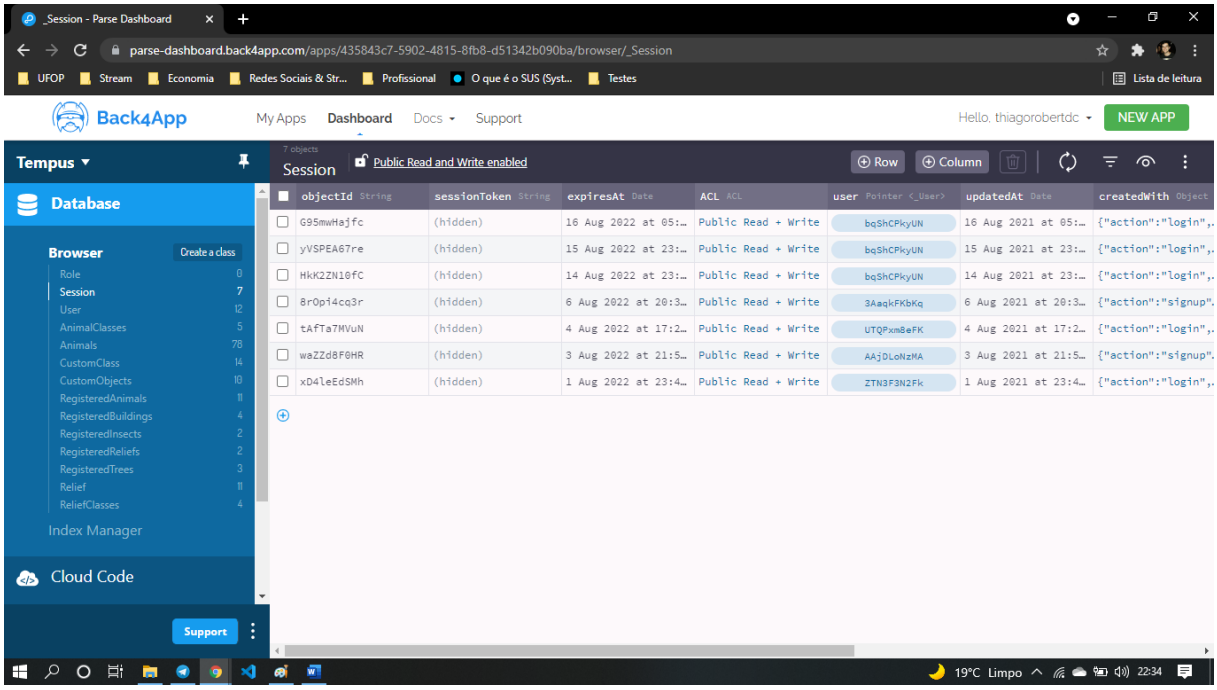
Fonte: Disponível em https://bit.ly/3uP3Dvl

4.6.1 Banco de Dados

O banco de dados é responsável pelo armazenamento e integridade dos dados dos usuários que utilizam a aplicação. O Back4App é uma plataforma *online* que abstrai toda a implementação de um banco de dados, fornecendo uma chave de acesso que deve ser inserida no código, e em poucos minutos o banco está pronto e apto a armazenar os dados. Suas funções são baseadas no Parse, antigo banco de dados do Facebook que se tornou público e gratuito.

A interface do Back4App é bem intuitiva e auxilia o administrador a criar as classes para armazenar os dados. A Figura 4.32 mostra que há uma coluna na lateral esquerda, onde as classes criadas se encontam, junto com um botão para criar uma nova classe. Os dados armazenados daquela classe são mostrados em formato de tabelas. As colunas são as propriedades do objeto, e as linhas são os objetos em si.

Figura 4.32 – Estrutura: Banco de dados.



Fonte: Elaborado pelo autor

As classes de nossa aplicação, e suas respectivas propriedades são descritas a seguir. Todos os objetos das classes possuem as propriedades *createdAt* e *updatedAt*, que indicam o momento de criação e atualização dos mesmos.

4.6.1.1 User

Classe correspondente ao usuário.

- **objectId**: ID correspondente do objeto.
- **emailVerified**: Variável boleana que indica se o usuário verificou ou não seu email após cadastro.
- **name**: Nome do usuário
- **username**: email utilizado pelo usuário
- **password**: senha do usuário - invisível para o administrador
- **tasks**: objeto contendo os tempos gastos pelo usuário ao realizar as tarefas do roteiro
- **likedTrees**: Relação que engloba todas as árvores favoritadas pelo usuário.
- **likedAnimals**: Relação que engloba todos os animais favoritados pelo usuário.
- **likedBuildings**: Relação que engloba todos os edifícios favoritados pelo usuário.

- **likedReliefs**: Relação que engloba todos os relevos favoritados pelo usuário.
- **likedInsects**: Relação que engloba todos os insetos favoritados pelo usuário.
- **likedCustoms**: Relação que engloba todos os objetos customizados favoritados pelo usuário.

4.6.1.2 AnimalClasses

Classe correspondente às classes de animais.

- **name**: Nome das classes de animais.

4.6.1.3 Animals

Classe correspondente aos animais disponíveis no banco.

- **name**: Nome do animal.
- **myClass**: Ponteiro para um AnimalClasses.

4.6.1.4 ReliefClasses

Classe correspondente às classes de relevos.

- **name**: Nome das classes de animais.

4.6.1.5 Relief

Classe correspondente aos relevos disponíveis no banco.

- **name**: Nome do relevo.
- **myClass**: Ponteiro para um ReliefClasses.

4.6.1.6 CustomClass

Classe correspondente às classes personalizadas criadas pelos usuários.

- **name**: Nome da classe.
- **type**: inteiro que indica o tipo da classe, utilizado para definir seu ícone.
- **user**: ponteiro para um User, indicando qual usuário criou a classe.

4.6.1.7 CustomObjects

Classe correspondente aos objetos customizados adicionados pelos usuários.

- **name**: Nome do objeto.
- **photo**: foto registrada do objeto personalizado.
- **user**: ponteiro para um User, indicando qual usuário registrou o objeto.
- **location**: Localização por extenso do objeto.
- **class**: ponteiro para um CustomClass, indicando qual classe personalizada o objeto pertence.
- **type**: inteiro que indica o tipo da classe, utilizado para definir seu ícone.
- **description**: descrição inserida pelo usuário ao registrar objeto.
- **coords**: contém as coordenadas geográficas do objeto registrado

4.6.1.8 RegisteredAnimals

Classe correspondente aos animais adicionados pelos usuários.

- **name**: Nome do animal.
- **photo**: foto registrada do animal.
- **user**: ponteiro para um User, indicando qual usuário registrou o animal.
- **animalType**: ponteiro para um Animals, indicando qual animal foi registrado.
- **location**: Localização por extenso do animal.
- **description**: descrição inserida pelo usuário ao registrar o animal.
- **coords**: contém as coordenadas geográficas do animal registrado

4.6.1.9 RegisteredReliefs

Classe correspondente aos relevos adicionados pelos usuários.

- **name**: Nome do relevo.
- **photo**: foto registrada do relevo.
- **user**: ponteiro para um User, indicando qual usuário registrou o relevo.

- **reliefType**: ponteiro para um Relief, indicando qual relevo foi registrado.
- **location**: Localização por extenso do relevo.
- **description**: descrição inserida pelo usuário ao registrar o relevo.
- **coords**: contém as coordenadas geográficas do relevo registrado

4.6.1.10 RegisteredBuildings

Classe correspondente aos edifícios adicionados pelos usuários.

- **name**: Nome do edifício.
- **photo**: foto registrada do edifício.
- **user**: ponteiro para um User, indicando qual usuário registrou o edifício.
- **location**: Localização por extenso do edifício.
- **description**: descrição inserida pelo usuário ao registrar o edifício.
- **coords**: contém as coordenadas geográficas do relevo registrado

4.6.1.11 RegisteredInsects

Classe correspondente aos insetos adicionados pelos usuários.

- **name**: Nome do inseto.
- **photo**: foto registrada do inseto.
- **user**: ponteiro para um User, indicando qual usuário registrou o inseto.
- **location**: Localização por extenso do inseto.
- **wings**: descrição das asas inserida pelo usuário ao registrar o inseto.
- **exoskeleton**: descrição do exoesqueleto inserida pelo usuário ao registrar o inseto.
- **parts**: descrição do corpo inserida pelo usuário ao registrar o inseto.
- **coords**: contém as coordenadas geográficas do inseto registrado

4.6.1.12 RegisteredTrees

Classe correspondente às árvores adicionados pelos usuários.

- **name**: Nome da árvore.
- **photo**: foto registrada da árvore.
- **user**: ponteiro para um User, indicando qual usuário registrou a árvore.
- **location**: Localização por extenso da árvore.
- **leafShape**: descrição das folhas inserida pelo usuário ao registrar a árvore.
- **topShape**: descrição da copa inserida pelo usuário ao registrar a árvore.
- **coords**: contém as coordenadas geográficas do inseto registrado

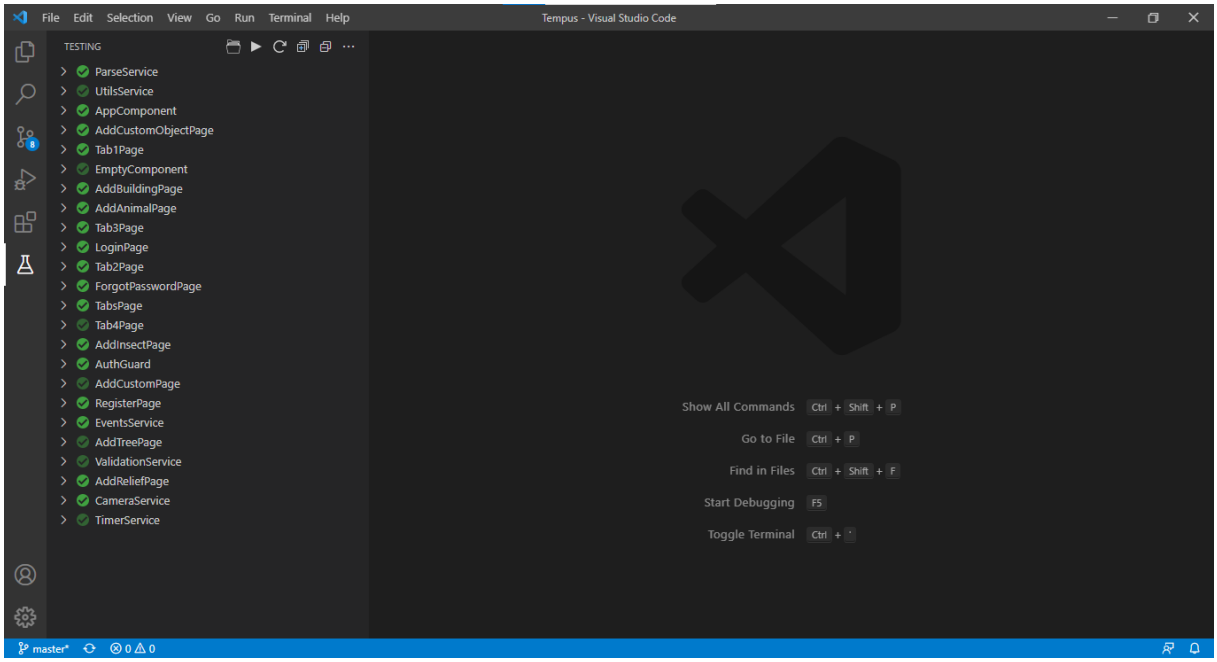
5 Resultados

Neste Capítulo são descritos os resultados finais dos testes aplicados. A Seção 5.1 descreve alguns dos testes automatizados implementados e seus resultados. A Seção 5.2 descreve os resultados obtidos através da aplicação dos testes referentes ao *System Usability Scale* - SUS e as inferências realizadas a partir das análises.

5.1 Testes automatizados

A Figura 5.2 são apresentados todos os testes automatizados após serem executados e passarem com sucesso.

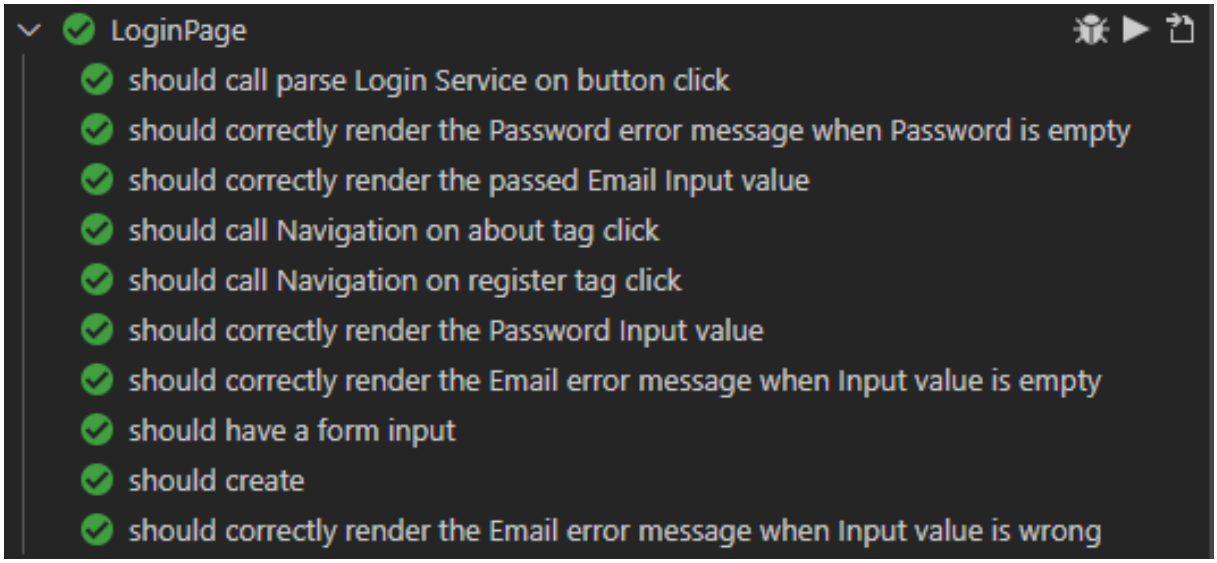
Figura 5.1 – Desenvolvimento: Laboratório de Testes.



Fonte: Elaborado pelo autor

Cada bloco possui testes próprios, correspondentes ao contexto da página que o mesmo representa. A Figura 5.2 mostra os testes da página de Login, tratando sobre as entradas no formulário de Login, da navegação para as páginas de cadastro e recuperação de senha, além da função de autenticação.

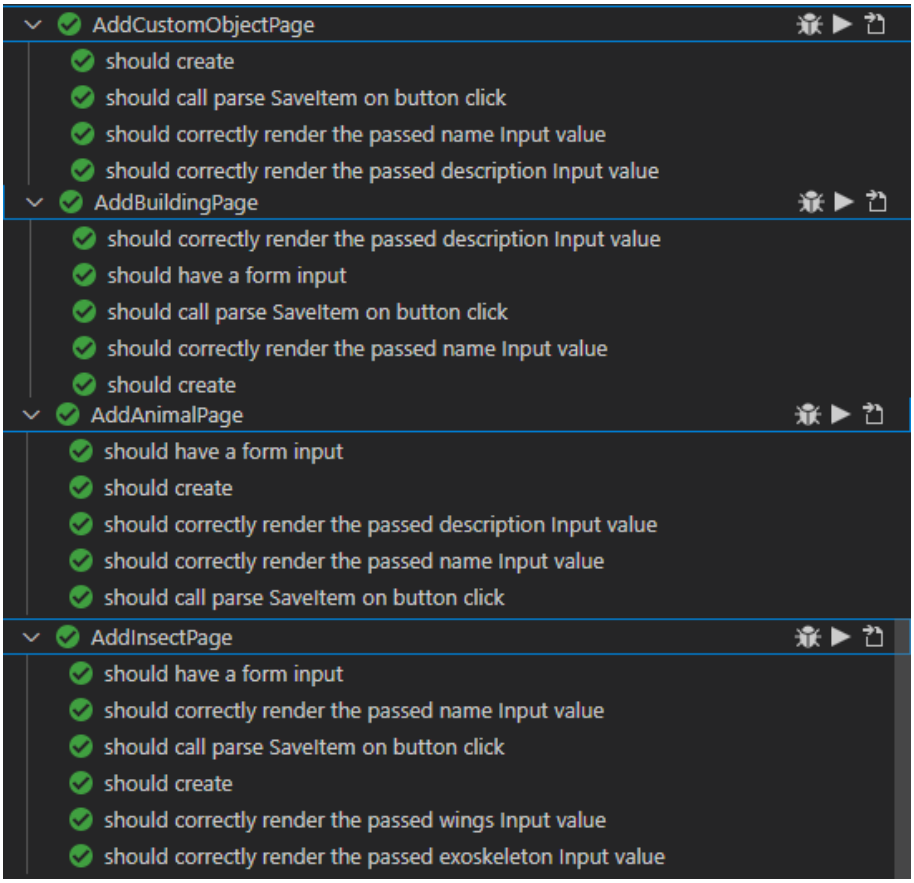
Figura 5.2 – Desenvolvimento: Bloco de Testes.



Fonte: Elaborado pelo autor

Os testes para as páginas de adição de objetos e de recuperação de senha seguem a mesma estrutura da página de Login, pois possuem contextos similares - formulários, inserções de dados e chamadas de funções. Assim, a Figura 5.3 ilustram os testes automatizados para os modais de adição de objetos customizados e 3 objetos padrões do sistema.

Figura 5.3 – Desenvolvimento: Bloco de Testes - Geral.



Fonte: Elaborado pelo autor

Exibidos os testes, conseguimos garantir um software consistente e de boa usabilidade, dado que os testes automatizados cobrem casos de uso de formulários, botões e chamadas de funções, tratando da ausência de dados e sempre informando ao usuário quando algo deu errado.

Ainda, os testes tornam a manutenção do código mais fácil, pois permite que, em um futuro problema, verifiquemos o teste que cobre o caso daquele problema ou que mais se aproxima do mesmo, sempre permitindo evoluir o código para algo mais limpo e eficiente.

5.2 Testes de usabilidade - SUS

A Tabela 5.1 apresenta o dia e os horários que cada usuário, representado por uma linha na horizontal, respondeu ao questionário do *System Usability Scale*, comumente chamado de SUS. O valor bruto do SUS (escala de 0 a 40) é calculado na coluna Score Bruto SUS para cada participante, e sua avaliação final - escala de 0 a 100 - se encontra na coluna Score Final SUS.

O questionário do SUS consistiu de 10 perguntas, sendo as seguintes:

1. Eu acho que gostaria de usar esse sistema com frequência.
2. Eu acho o sistema desnecessariamente complexo.

3. Eu achei o sistema fácil de usar.
4. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.
5. Eu acho que as várias funções do sistema estão muito bem integradas.
6. Eu acho que o sistema apresenta muita inconsistência.
7. Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.
8. Eu achei o sistema atrapalhado de usar.
9. Eu me senti confiante ao usar o sistema.
10. Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

As respostas foram obtidas através dos formulários do Google e registradas em planilha pelo próprio sistema.

Por fim, temos a média final SUS, que é a média final das avaliações finais de cada usuário. O valor final da média para a aplicação Tempus nesta bateria de testes foi de 86,25. Lembrando que o valor mínimo para considerarmos uma aplicação com boa usabilidade é de 68, temos que nossa aplicação alcançou não só o necessário mas também foi além, apresentando bons resultados de usabilidade e um alto valor na escala.

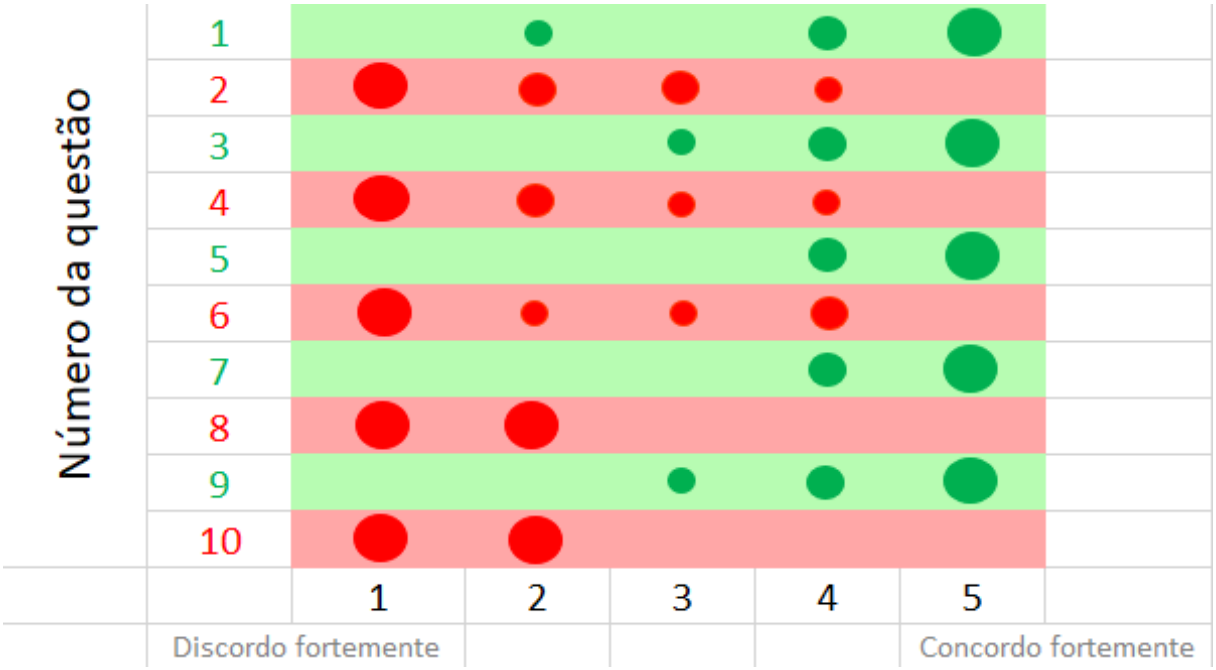
Tabela 5.1 – Testes de usabilidade realizados junto ao usuário

Carimbo de data/hora	Pergunta 1	Pergunta 2	Pergunta 3	Pergunta 4	Pergunta 5	Pergunta 6	Pergunta 7	Pergunta 8	Pergunta 9	Pergunta 10	Score Bruto SUS	Score Final SUS
8/3/2021 22:54:08	4	1	5	1	5	1	4	1	5	1	38	95
8/3/2021 23:29:51	5	2	5	1	5	1	5	2	5	1	38	95
8/3/2021 23:32:54	2	3	4	2	5	4	4	2	3	1	26	65
8/4/2021 0:03:44	4	1	5	1	5	4	5	1	4	1	35	87.5
8/4/2021 0:28:36	5	1	5	1	5	1	5	1	5	1	40	100
8/4/2021 7:33:20	5	1	5	1	5	2	5	1	5	1	39	97.5
8/4/2021 8:46:04	5	2	3	3	4	2	4	2	3	1	29	72.5
8/4/2021 20:59:09	5	4	3	4	4	3	5	2	3	2	25	62.5
8/4/2021 22:03:45	2	1	5	1	5	4	5	2	3	1	31	77.5
8/4/2021 23:48:46	5	1	4	1	5	1	5	1	5	1	39	97.5
8/6/2021 17:38:54	5	2	5	2	5	1	5	1	5	2	37	92.5
8/16/2021 17:15:17	4	1	5	1	5	1	5	2	4	1	37	92.5
											Média final SUS	**86.25**

Fonte: Elaborado pelo autor

Ainda, para uma melhor visualização do significado das respostas do teste SUS aplicado, podemos interpretar assim como a Figura 5.4 mostra.

Figura 5.4 – Resultados: Gráfico de bolhas.



Fonte: Elaborado pelo autor

As perguntas de número ímpar que reforçam características positivas da aplicação, possui maior concentração de respostas alinhadas à direita, orientação horizontal que significa que o usuário concorda fortemente. Já as perguntas de números pares que afirma negativamente sobre o software sob teste, possui maior concentração de respostas à esquerda.

Salvas as devidas proporções, dado que os usuários não necessariamente faziam parte do público alvo da aplicação, o gráfico mostra que os usuários tiveram uma boa perspectiva da aplicação.

5.3 Tempo e *feebacks*

A última métrica coletada foi o tempo que cada usuário levou para realizar as tarefas do roteiro. A Tabela 5.2 faz uma relação de tarefa e tempo gasto por usuário. As que aparecem sem a contagem de tempo foram tarefas em que o algoritmo falhou em cronometrar o tempo gasto pelo usuário em realizar a tarefa ou tarefas que o usuário não conseguiu completar.

Tabela 5.2 – Tempos gastos por usuário para realizar cada tarefa

Segundo por tarefa	Usuário 1	Usuário 2	Usuário 3	Usuário 4	Usuário 5	Usuário 6	Usuário 7	Usuário 8	Usuário 9	Usuário 10	Usuário 11	Usuário 12
Tarefa 1	20.69s	32.027s	15.977s	25.509s	30.944s	27.882s	33.028s	19.046s	27.595s	51.298s	32.036s	41.392s
Tarefa 2	-	-	-	-	-	-	-	-	-	-	-	-
Tarefa 3	-	-	-	-		-	-	-	-	-	-	-
Tarefa 4	17.021s	10.619s	7.031s	11.963s	20.842s	14.187s	10.375s	11.82s	5.559s	18.827s	11.658s	18.263s
Tarefa 5	10.544s	30.04s	10.826s	20.26s	13.983s	6.283s	11.079s	55.324s	9.583s	10.399s	19.368s	13.266s
Tarefa 6	33.732s	33.413s	120.444s	45.465s	151.15s	77.24	6.537s	4.885s	27.153s	3.873s	0s	17.91s
Tarefa 7	35.893s	1.304s		53.654s	73.049s	5.385s	1.997s	4.915s		51.882s	6.866s	0.804s
Tarefa 8	-	966.052s	-	-	-	207.222s	-	35.646s	22.954s	17.543s	60.297s	40.63s
Tarefa 9	27.95s	0s	34.262s	15.452s	113.788s	21.606s	5.779s	137.884s	52.639s	56.216s	0s	41.195s
Tarefa 10	95.821s	0s	87.335s	74.877s	191.949s	221.895s	23.898s	70.519s	16.417s	61.216s	10.863s	56.697s
Tarefa 11	-	-	-	67.091s	180.859s	222.841s	18.305s	-	15.715s	-	9.091s	64.12s
Tarefa 12	-	-	-	-	-	-	-	-	-	-	-	-
Tempo total	241,6s	1.073,4s	275,8s	314,2s	776,5s	804,5s	111s	340s	177,6s	271,2s	150,1s	294,2s

Fonte: Elaborado pelo autor

As tarefas 2 e 3, referentes à adição de um objeto do tipo Animal e de um objeto costumizado, não tiveram os tempos medidos com sucesso. Entretanto, todos os usuários relataram que conseguiram realizar ambas as tarefas sem nenhum problema. O mesmo ocorre com a tarefa 12, que consiste em se deslogar do sistema. Os usuários relataram que acharam a opção de *logout*, mas se depararam com um cenário de carregamento infinito, onde nunca chegavam na tela de Login. Isso ocorreu devido à uma implementação errada, onde uma função que necessita do token de autenticação do usuário estava sendo chamada logo após a função de logout, o que foi corrigido para a versão final.

A tarefa 8, que consiste em apagar um objeto da sua lista pessoal de variáveis de ambiente foi a que teve mais divergências de tempo. O tempo máximo entre aqueles que conseguiram realizar a tarefa foi de 966.052 segundos, o que são pouco mais de 16 minutos, um tempo excessivamente alto. Outro usuário conseguiu realizar a tarefa com 207 segundos, pouco mais de 3 minutos. Esses valores indicam um provável problema de usabilidade, já que muitos usuários não conseguiram realizar a tarefa e, entre os que conseguiram, houve uma grande divergência entre os tempos.

6 Considerações Finais

6.1 Conclusão

A integração de componentes digitais na vida pessoal e profissional das pessoas ocorreu na última década como consequência natural da evolução computacional. As áreas da educação também sentem esse impacto, produzindo e oferecendo ferramentas cada vez mais eficientes.

Estudos de campo são realizados desde as áreas iniciais de aprendizado, permitindo que se vivencie um pouco aquilo que se estuda. A ciência cidadã, por sua vez, ajuda a entender contextos únicos e elaborar soluções a curto, médio e longo prazo, coletando dados de maneira individual ou coletiva. A prática vem contando com a ajuda de ferramentas computacionais nos últimos anos, tornando a mesma mais rápida e efetiva, além de permitir a colaboração de um maior número de usuários.

A coleta e interpretação de dados é importante para se conhecer a realidade dos ambientes nos quais estamos inseridos. A interação com os elementos promove melhor aprendizado e memorização. Estudos de campo e ciência cidadã são práticas que promovem a imersão de seus participantes, permitindo que interajam com os elementos que compõe os lugares que eles penetram. Isso ajuda a construir conhecimento e criar metas e objetivos, moldando e alterando o espaço de estudo.

Desse modo, o objetivo deste trabalho foi desenvolver um aplicativo que auxiliasse os usuários nessas atividades, ou seja, na coleta e armazenamento de dados de interesse. Tudo isso, utilizando elementos visuais intuitivos e que proprocionam uma boa experiência e usabilidade, comprovadas por métodos de testagem, automatizados e heurísticos, que garantem a acurácia e qualidade da aplicação.

O aplicativo atualmente se encontra disponível na *Play Store* para ser baixado e utilizado.

6.2 Trabalhos Futuros

Conforme análise e comparação do aplicativo proposto com trabalhos correlatos, a principal limitação do aplicativo atualmente é o fato dele não fornecer uma opção de coleta compartilhada. A seguir, são listadas as atividades a serem realizadas a fim de melhorar o trabalho.

1. Implementar um algoritmo de inteligência artificial que consiga reconhecer, no conjunto de dados coletados, possíveis classes adequadas com o que o usuário está coletando e sugira como categoria.
2. Desenvolver uma versão administrativa, semelhante ao aplicativo, permitindo a administração dos dados.
3. Fornecer aos usuários relatórios sobre os dados coletados em períodos mensais.

Referências

AMORETTI, M. S. M. Protótipos e estereótipos: aprendizagem de conceitos mapas conceituais: experiência em educação a distância. *Informática na educação: teoria e prática*, v. 4, n. 2, p. 49–55, 2001.

BONNEY, R.; SHIRK, J. L.; PHILLIPS, T. B.; WIGGINS, A.; BALLARD, H. L.; MILLER-RUSHING, A. J.; PARRISH, J. K. Next steps for citizen science. *Science*, American Association for the Advancement of Science, v. 343, n. 6178, p. 1436–1437, 2014.

BOURHIS, P. et al. Json: Data model, query languages and schema specification. *PODS 2017 - Proceedings of the Thirty-Sixth ACM SIGMODSIGACT-SIGART Symposium on Principles of Database Systems*, 2017.

CHARLAND, A.; LEROUX, B. Mobile application development: Web vs. native. *Communications of the ACM*, p. 9–53, 2011.

FIGUEIREDO, E. Requisitos funcionais e requisitos não funcionais. *Icex, Dcc/Ufmg*, 2011.

GUPTA, A.; THAPAR, J.; SINGH, A.; SINGH, P.; SRINIVASAN, V.; VARDHAN, V. Simplifying and improving mobile based data collection. In: *Proceedings of the Sixth International Conference on Information and Communications Technologies and Development: Notes-Volume 2*. [S.l.: s.n.], 2013. p. 45–48.

HARTMANN, G. et al. Cross-platform mobile development. *Tribal, Lincoln House, The Paddocks, Tech.*, 2011.

ISLAM, M. R. et al. Mobile application and its global impact. *International Journal of Engineering Technology IJET-IJENS*, v. 10, n. 06, 2010.

KIM, S.; MANKOFF, J.; PAULOS, E. Sensr: evaluating a flexible framework for authoring mobile data-collection tools for citizen science. In: *Proceedings of the 2013 conference on Computer supported cooperative work*. [S.l.: s.n.], 2013. p. 1453–1462.

KRAVCIK, M. et al. Mobile collector for field trips. *Educational Technology Society*, p. 25–33, 2004.

KRESS, W. J.; GARCIA-ROBLEDO, C.; SOARES, J. V.; JACOBS, D.; WILSON, K.; LOPEZ, I. C.; BELHUMEUR, P. N. Citizen science and climate change: mapping the range expansions of native and exotic plants with the mobile app leafsnap. *BioScience*, Oxford University Press, v. 68, n. 5, p. 348–358, 2018.

LEWIS, J. R. The system usability scale: past, present, and future. *International Journal of Human–Computer Interaction*, Taylor & Francis, v. 34, n. 7, p. 577–590, 2018.

MARÇAL, E.; ANDRADE, R. M. de C.; VIANA, W. Mobile learning em aulas de campo: um estudo de caso em geologia. *RIED. Revista Iberoamericana de Educación a Distancia*, Asociación Iberoamericana de Educación Superior a Distancia, v. 20, n. 2, p. 315–336, 2017.

MARÇAL, E.; VIANA, W.; ANDRADE, R. M.; RODRIGUES, D. A mobile learning system to enhance field trips in geology. In: IEEE. *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*. [S.l.], 2014. p. 1–8.

MORESI, E. A. D.; BARBOSA, J. A.; FILHO, M. B.; ALVES, P. N.; SANTOS, J.; ASSIS, F.; BERNARDES, T.; LIMA, V. O emprego do aplicativo scihub em projetos de ciência cidadã. *Revista Iberoamericana De Sistemas, Cibernética E Informática*, v. 14, n. 2, p. 45–52, 2017.

PALMIERI, M. et al. Comparison of cross-platform mobile development tools. *16th International Conference on Intelligence in Next Generation Networks*, p. 179–186, 2012.

PREECE, J. Citizen science: New research challenges for human–computer interaction. *International Journal of Human-Computer Interaction*, Taylor & Francis, v. 32, n. 8, p. 585–612, 2016.

RAMANATHAN, N.; ALQUADDOOMI, F.; FALAKI, H.; GEORGE, D.; HSIEH, C.-K.; JENKINS, J.; KETCHAM, C.; LONGSTAFF, B.; OOMS, J.; SELSKY, J. et al. Ohmage: an open mobile system for activity and experience sampling. In: IEEE. *2012 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops*. [S.l.], 2012. p. 203–204.

SENICIATO, T.; CAVASSAN, O. Aulas de campo em ambientes naturais e aprendizagem em ciências – um estudo com alunos do ensino fundamental. 2004.

SILVA, E. P. A. da; SOTTO, E. C. S. Utilização do ionic framework no desenvolvimento de aplicações híbridas em arquitetura orientada a serviço. *Revista Interface Tecnológica*, v. 15, n. 1, p. 97–108, 2018.

SILVA, L. L. B. da et al. Desenvolvimento de aplicações para dispositivos móveis: Tipos e exemplo de aplicação na plataforma ios. *II Workshop de Iniciação Científica em Sistemas de Informação, Goiânia - GO*, 2015.

SILVA, M. M. da; SANTOS, M. T. P. Os paradigmas de desenvolvimento de aplicativos para aparelhos celulares. *T.I.S São Carlos*, v. 3, n. 2, p. 162–170, 2014.

SMUTNỲ, P. Mobile development tools and cross-platform solutions. In: IEEE. *Proceedings of the 13th International Carpathian Control Conference (ICCC)*. [S.l.], 2012. p. 653–656.

SRIVASTAVA, N.; KUMAR, U.; SINGH, P. Software and performance testing tools. *Journal of Informatics Electrical and Electronics Engineering*, v. 2, n. 01, p. 1–12, 2021.

TOMLINSON, M.; SOLOMON, W.; SINGH, Y.; DOHERTY, T.; CHOPRA, M.; IJUMBA, P.; TSAI, A. C.; JACKSON, D. The use of mobile phones as a data collection tool: a report from a household survey in south africa. *BMC medical informatics and decision making*, BioMed Central, v. 9, n. 1, p. 1–8, 2009.

TREVISAN, I.; SILVA-FORSBERG, M. C. Aulas de campo no ensino de ciências e biologia: Aproximações com a abordagem ciência, tecnologia e sociedade (cts). *Scientia Amazonia*, v. 3, n. 1, p. 138–148, 2014.

TUMELERO, N. *Pesquisa de campo: conceitos, finalidade e etapas de como fazer*. Naína Tumelero, 2018. Disponível em: <https://blog.mettzer.com/pesquisa-de-campo/>.