

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

Alfeu Gonçalves dos Santos

**Aplicativo para classificação de macrofungos utilizando redes neurais**

Florianópolis

2022

Alfeu Gonçalves dos Santos

**Aplicativo para classificação de macrofungos utilizando redes neurais**

Trabalho de Conclusão de Curso submetido ao Departamento de Informática e Estatística da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador(a): Prof. Aldo von Wangenheim

Florianópolis

2022

Dedico este trabalho a minha mãe Terezinha Heiderscheidt dos Santos,  
minha irmã Camila Gonçalves dos Santos  
e minha namorada Iany Thaisy Braga da Costa.

## **AGRADECIMENTOS**

Agradeço especialmente a minha família, pela persistência em me educar, acompanhar e aconselhar nos momentos difíceis da minha vida. Agradeço a todos os profissionais da Universidade Federal de Santa Catarina, especialmente ao Prof. Aldo von Wangenheim que aceitou a tarefa de ser meu orientador e a Jean Carlo Rossa Hauck, Genivaldo Alves da Silva e Mateus Grellert da Silva que aceitaram fazer parte da minha banca de avaliação. Agradeço ao amigo Roger de Córdova Farias, responsável pelo desenvolvimento da API utilizada pelo meu aplicativo e que teve a paciência para corrigir problemas, efetuar melhorias e participar de longas reuniões comigo. Aos colegas do grupo de pesquisa MIND.Funga pela colaboração ao longo da realização deste trabalho. E por fim, a minha namorada Iany Thaisy Braga da Costa que me acompanhou nessa jornada com muito companheirismo, amizade e leveza, me dando todo o suporte necessário para conquistar os meus objetivos.

## RESUMO

O Brasil é um país rico em florestas e com vários biomas diferentes espalhados pelo território nacional. São diferentes climas e características que possibilitam o surgimento de diversas espécies de macrofungos, existindo assim, muitos profissionais e grupos de pesquisas mobilizados em catalogar e popularizar a micologia para um maior número de pessoas. Buscando criar uma ferramenta que possa ser utilizada por esses profissionais e visando colaborar no processo de catalogação e popularização do conhecimento em micologia, foi desenvolvido um aplicativo que visa a identificação rápida de algumas espécies, bem como o registro de fotos de macrofungos por parte dos usuários. Esse aplicativo também pode servir como ferramenta de ensino sendo utilizado em campo para auxiliar professores durante as suas aulas sobre macrofungos, além de possibilitar que voluntários auxiliem na ampliação das bases de dados de imagens dos cientistas. O presente trabalho foi realizado no contexto do projeto "MIND.Funga - Da pesquisa com macrofungos ameaçados de extinção das matas nebulares de Santa Catarina à inovação na identificação das espécies" do grupo de pesquisa MIND.Funga da UFSC. A aplicação se utilizou da rede neural treinada com o objetivo de reconhecer macrofungos através da base de imagens do grupo. A mesma foi testada por cientistas e pessoas convidadas por esse grupo, se mostrando bem sucedida na sugestão de nomes e obtenção de imagens para a base de dados desse grupo.

**Palavras-chave:** Desenvolvimento de Aplicativos; macrofungos; rede neural.

## **ABSTRACT**

Brazil is a country rich in forests and with several different biomes spread throughout the national territory. There are different climates and characteristics that enable the emergence of several species of macrofungi, thus, there are many professionals and research groups mobilized in cataloging and popularizing mycology to a larger number of people. Seeking to create a tool that can be used by these professionals and aiming to collaborate in the process of cataloging and popularizing the knowledge in mycology, an application was developed that aims at the quick identification of some species, as well as the registration of photos of macrofungi by the users. This application can also serve as a teaching tool, being used in the field to help teachers during their classes about macrofungi, besides enabling volunteers to help in the expansion of the image databases of scientists. The present work was carried out in the context of the project "MIND.Funga - From research with endangered macrofungi of the cloud forests of Santa Catarina to innovation in the identification of species" of the MIND.Funga research group of the UFSC. The application used a neural network trained with the objective of recognizing macrofungi through the group's image base. It was tested by scientists and people invited by this group, and proved to be successful in suggesting names and obtaining images for the group's database.

**Keywords:** Application Development; macrofungi; neural network.

## LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
MIND.F	Monitoring and Inventorying Neotropical Diversity of Fungi.
LAPIX	Image Processing and Computer Graphics Lab
UFSC	Universidade Federal de Santa Catarina
CNN	Convolutional neural network
JPEG	Joint Photographic Experts Group
JSON	JavaScript Object Notation
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
SCSS	Sassy Cascading Style Sheets
MVC	Model-View-Controller
HTTP	Hypertext Transfer Protocol
APK	Android Package Kit

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>9</b>
1.1 MOTIVAÇÃO	10
1.2 OBJETIVOS	10
1.2.1 Objetivo Geral	10
1.2.2 Objetivo Específicos	10
1.3 MÉTODOS DA PESQUISA	11
<b>2 FUNDAMENTAÇÃO TEÓRICA</b>	<b>12</b>
2.1 MICOLOGIA E MACROFUNGOS	12
2.2 AS TECNOLOGIAS DO GRUPO MIND. FUNGA E LAPIX	14
2.2.1 A Rede Neural Convolucional	14
2.2.2 API - Comunicação com o banco de dados e com a CNN	15
2.3 IONIC FRAMEWORK	20
2.4 ANÁLISE DE OUTROS APLICATIVOS	21
2.4.1 Definições da Pesquisa	22
2.4.2 Execução da Pesquisa	23
2.4.3 Análise dos aplicativos selecionados	23
2.4.3.1 PlantSnap	23
2.4.3.2 Fungus	24
2.4.3.3 Picture Mushroom	25
2.4.3.4 Mushroom Identifier	26
2.4.3.5 ShroomID	27
2.4.4 Comparativo e conclusões sobre e os aplicativos selecionados	28
<b>3 DESENVOLVIMENTO DO APLICATIVO</b>	<b>30</b>
3.1 ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE	30
3.1.1 Requisitos funcionais	30
3.1.2 Requisitos não funcionais	32
3.2 CASOS DE USO	33
3.3 DIAGRAMA DE VISÃO GERAL DE INTERAÇÃO	36
3.4 PROTÓTIPOS DE INTERFACE	37
3.5 TESTES NA API	40
3.6 PROGRAMAÇÃO DO APLICATIVO	41
<b>4 RESULTADOS ALCANÇADOS</b>	<b>48</b>
4.1 APLICATIVO DESENVOLVIDO	48
4.2 PESQUISA DE AVALIAÇÃO DO APLICATIVO	66
<b>5 CONCLUSÃO</b>	<b>71</b>
5.1 SUGESTÕES PARA TRABALHOS FUTUROS	71
<b>REFERÊNCIAS</b>	<b>73</b>



<b>APÊNDICE A – ARTIGO APLICATIVO PARA CLASSIFICAÇÃO DE MACROFUNGOS UTILIZANDO REDES NEURAS</b>	<b>76</b>
<b>APÊNDICE B – CÓDIGO FONTE</b>	<b>95</b>

## 1 INTRODUÇÃO

Tendo em vista a grande diversidade de macrofungos nas florestas brasileiras existe por parte da comunidade científica um esforço voltado ao estudo, à catalogação e ao registro fotográfico desses organismos. O MIND.Funga<sup>1</sup> é um exemplo de grupo de estudo com interesse em pesquisas de curto e longo prazo com macrofungos presentes em Florianópolis e região, sendo verificado pelo grupo a necessidade da criação de uma ferramenta que auxiliasse no processo de reconhecimento e armazenamento das informações coletadas pelo grupo.

O presente trabalho visa apresentar o processo de desenvolvimento da ferramenta solicitada pelo grupo, um aplicativo desenvolvido para o sistema Android que permite aos cientistas e convidados do grupo efetuarem o registro de imagens e a identificação de macrofungos através da sugestão de nomes de espécies. Foi utilizado o conhecimento prévio da equipe do MIND.Funga que forneceram uma curadoria especializada sobre as espécies com as informações e imagens coletadas por eles e também o LAPIX<sup>2</sup> com a rede neural treinada para identificação desses fungos e uma API com os endpoints que possibilitaram a comunicação entre o aplicativo desenvolvido e o backend criado anteriormente. Buscou-se entender os elementos criados pelos pesquisadores desses grupos, efetuando os testes dessas soluções e a integração com o aplicativo criado.

Os usuários do aplicativo podem efetuar o cadastro no sistema a edição básica de imagens coletadas em campo e a submissão das mesmas na rede neural treinada para identificação de macrofungos, sendo possível também, a verificação de informações sobre as espécies.

Foram verificadas também algumas soluções já disponíveis no mercado voltadas para identificação de macrofungos, como por exemplo, o *Fungus* um aplicativo desenvolvido por Dominik Steinhauser<sup>3</sup> feito para o objetivo específico do estudo da micologia, além de aplicativos de uso mais geral que reconhecem também plantas, flores, árvores e cactos como o PlantSnap<sup>4</sup>, buscando comparar e trazer elementos dessas soluções ao nosso aplicativo.

---

<sup>1</sup> <https://mindfunga.ufsc.br/>

<sup>2</sup> <https://lapix.ufsc.br/>

<sup>3</sup> <https://www.appbrain.com/dev/Dominik+Steinhauser/>

<sup>4</sup> <https://www.plantsnap.com/>

## 1.1 MOTIVAÇÃO

O desenvolvimento de um aplicativo para identificação de fungos feito para o grupo visa fornecer aos usuários uma experiência de uso pensada para as necessidades desses cientistas, buscando a captura e edição de imagens e a utilização dos algoritmos de reconhecimento visual criados com a curadoria dessa equipe com amplo conhecimento na área. Com a tecnologia sendo testada pelos pesquisadores, pode-se obter uma evolução do aplicativo buscando superar a precisão dos aplicativos já existentes. A ferramenta pode ser utilizada no ensino da micologia em instituições de ensino médio e para o uso de cientistas e pessoas interessadas nessa área de conhecimento, trazendo uma variada gama de informações e imagens das espécies disponíveis no banco de dados, sendo importante para coleta e armazenamento de novas imagens no banco de dados do grupo, possibilitando novas descobertas e futuras melhorias no treinamento da rede neural existente.

## 1.2 OBJETIVOS

### 1.2.1 Objetivo Geral

Desenvolver uma aplicação de front-end que possibilite a fotografia e identificação de macrofungos, permitindo a coleta e edição de fotos bem como a verificação de informações sobre esses organismos.

### 1.2.2 Objetivo Específicos

Para alcançar o objetivo geral foi necessário criar alguns objetivos específicos, sendo estes descritos abaixo:

- Avaliar as documentações criadas pelo MIND.Funga e LAPIX buscando entender o que já havia sido desenvolvido e visando a integração com o aplicativo;

- Efetuar testes na API e na rede neural desenvolvida, buscando por problemas nesses elementos;
- Sugerir melhorias na API para correção de problemas de integração com o aplicativo;
- Avaliar as soluções já existentes no mercado fazendo um levantamento de suas funcionalidades buscando comparar com o aplicativo do MIND.Funga;
- Desenvolver um aplicativo mobile para que os pesquisadores possam efetuar a coleta de informações através do celular ;

### 1.3 MÉTODOS DA PESQUISA

Para o elaboração do trabalho foram adotados alguns procedimentos buscando o seu desenvolvimento, sendo estes elencados abaixo:

- Reuniões e entrevistas com os componentes do grupo buscando entender os problemas enfrentados pelos mesmos;
- Levantamento dos requisitos com base nos relatos do grupo do MIND. Funga e LAPIX;
- Estudo da documentação das tecnologias desenvolvidas pelo grupo MIND.Funga e LAPIX;
- Elaboração e adaptação do protótipo das interfaces para o aplicativo;
- Análise comparativa dos aplicativos disponíveis, buscando avaliar suas funcionalidades e fazer comparações com o aplicativo do MIND. Funga;

- Testes de versões preliminares com os membros do grupo, buscando corrigir problemas e gerar melhorias;

## **2 FUNDAMENTAÇÃO TEÓRICA**

Para o entendimento deste trabalho é necessário conhecer alguns elementos que servirão como base para a elucidação das tecnologias e conceitos utilizados ao longo do desenvolvimento do aplicativo. Esses elementos podem ser divididos em alguns tópicos, sendo eles:

- Os conceitos gerais sobre macrofungos e micologia;
- Tecnologias desenvolvidas pelo grupo MIND. Funga e LAPIX que foram utilizadas pelo aplicativo;
- O framework de desenvolvimento IONIC, que forneceu as ferramentas necessárias para a criação do aplicativo;

### **2.1 MICOLOGIA E MACROFUNGOS**

De acordo com Oliveira (2014) a Micologia compreende um vasto campo de estudo que envolve os microrganismos conhecidos como fungos. O estudo desses organismos interessa aos mais variados setores da nossa sociedade com seus reflexos sendo vistos em produtos químicos e farmacêuticos, comestíveis, entre outros.

Nestas condições, os fungos trazem, muitos benefícios, tendo também, grande importância na natureza, através da ciclagem de nutrientes, como o carbono, como também sendo meio de conexão e comunicação entre os vários componentes dos ecossistemas (Webster & Weber 2007). Alguns envenenamentos também podem ser causados por fungos supostamente comestíveis, pela semelhança com estes, determinando quadros de micetismos (intoxicações pela ingestão de fungos).

Os fungos podem ser divididos entre microfungos e macrofungos dependendo da forma como se apresentam, mais especificamente suas estruturas reprodutivas. Os microfungos apresentam caracteres reprodutivos não visíveis a olho nu, sendo possível somente obter informações diagnósticas por aspectos

macro, com relação à cor, textura e organização. Já os macrofungos produzem estruturas macroscópicas, sendo visíveis a olho nu, ou seja, com tamanho superior a 1 mm (Marques, 2012).

Os macrofungos correspondem a um grupo relativamente pequeno com cerca de 15% das espécies de fungos conhecidas (Mueller et al. 2007).

Figura 1 – Exemplo de macrofungo / Trametes sp.



Fonte: Grupo MIND.Funga

A maioria dos macrofungos vivem nos mais diversos substratos da natureza sendo eles o solo seco, pântanos, troncos apodrecidos, entre outros, conforme explica Oliveira (2014).

Durante o desenvolvimento deste trabalho se deu o foco nos macrofungos, já que nessa categoria de fungos é possível a captura das fotos nas florestas através de um smartphone para posterior análise no aplicativo.

No âmbito do trabalho também é importante destacar os conceitos de Espécie e Espécime que são bastante utilizados pela API, segundo definição do Dicionário Ambiental (2014) espécie é uma palavra que aponta sempre para uma coletividade, vindo do latim species, “conjunto de traços que fazem reconhecer qualquer objeto”. Já os espécimes se referem a um indivíduo dentro dessa

coletividade, do latim specimen, “amostra, prova” é “qualquer indivíduo de uma espécie”. No aplicativo os espécimes são entendidos como amostras, ou fotos coletadas, essas imagens sempre fazem parte de uma determinada espécie.

## 2.2 AS TECNOLOGIAS DO GRUPO MIND. FUNGA E LAPIX

Para o desenvolvimento do aplicativo foram utilizadas algumas tecnologias que já haviam sido previamente criadas por membros do MIND.Funga em conjunto com o LAPIX que é um laboratório da UFSC dedicado em pesquisas e estudos em novas tecnologias para processamento, análise e visualização de imagens, entre outros. Para identificação dos macrofungos, já existia uma rede neural treinada para isso, sendo desenvolvida por Thiago Zimmermann Loureiro Chaves e uma API com os endpoints utilizados pelo aplicativo desenvolvida por Farias (2022).

### 2.2.1 A Rede Neural Convolutacional

As chamadas CNN ou redes neurais convolucionais, buscam fornecer uma união de redes neurais e a operação matemática de convolução utilizadas para filtros das imagens. Um filtro de convolução é uma matriz pequena que vai passando por cima de uma outra matriz bem maior, no caso a imagem, e vai multiplicando cada ponto da imagem e sua vizinhança. O resultado é uma nova imagem filtrada, onde uma determinada característica foi ressaltada ou alguma coisa que não queremos foi apagada. Se dá a essa técnica o nome de CNN ou redes de aprendizado profundo. As redes comuns simulam o funcionamento de partes do tecido cerebral de animais e possuem modelos computacionais de neurônios, esses neurônios são conectados entre si e respondem a estímulos, já as CNNs são redes que possuem diversas camadas de neurônios e, entre essas camadas, ao invés de aprenderem apenas com as conexões entre os neurônios aprendem também com os vários conjuntos de filtros de imagem, sendo uma característica importante dessas redes a capacidade de aprender com esses filtros de imagem (Bittencourt et al. 2022).

Segundo (Drechsler-Santos, 2022) o conjunto de dados do MIND.Funga que foi utilizado para o treinamento dessa CNN é composto por 14.707 imagens

selecionadas representando 423 espécies de fungos, as imagens brutas foram recebidas por e-mail ou incorporadas no conjunto de dados, as fontes incluíam micologistas parceiros, voluntários, cientistas cidadãos, grupos de micologia online e projetos de pesquisa anteriores focados na diversidade de macrofungos. Todas as imagens foram submetidas a um processo de controle de qualidade e tratamento, que envolveu algumas etapas críticas:

- Remoção de imagens de baixa qualidade, sem foco e imagens com objetos estranhos;
- Reenquadramento do espécime para o centro da imagem e padronização da proporção;
- As imagens tiveram seu fundo original substituído por fundo verde e branco (códigos HEX) para melhorar o treinamento da CNN e o reconhecimento do espécime. O grupo verificou que a substituição do plano de fundo não melhorou significativamente o desempenho da CNN, sendo assim, novas adições não foram modificadas (Drechsler-Santos, 2022).

### **2.2.2 API - Comunicação com o banco de dados e com a CNN**

A utilização da CNN por parte do aplicativo se dá através dos endpoints da API desenvolvida por Farias (2022), que efetua a comunicação entre o aplicativo e a parte de processamento e identificação das imagens, além de retornar dados armazenados no banco de dados da aplicação, tais como, dados das espécies, dados de usuários e informações de espécimes .

De acordo com Oliveira (apud Sousa, 2021) a API do inglês Application Programming Interface se apresenta como uma interface desenvolvida por código e que obedece a um conjunto de padrões e rotinas. Possibilitando a integração de diferentes tipos de aplicações e sistemas, portanto, uma API permite que qualquer cliente que tenha acesso a documentação e aos endpoints da mesma possa consumir os seus processos, tornando desnecessária a construção de outra aplicação com a mesma finalidade. Sendo assim, uma API possui um ou diversos endpoints, que fornecem um serviço para ser consumido por outras aplicações clientes.



Para utilizar os recursos dessa API o aplicativo precisa submeter requisições HTTP para a mesma enviando ou recebendo dados para o funcionamento correto do aplicativo. A API que foi utilizada pelo aplicativo está dividida em 5 grandes grupos, esses grupos possuem os endpoints que são utilizados pelo aplicativo, sendo eles exemplificados abaixo:

- Curatorship (Curadoria) - Nesse grupo estão os endpoints relacionados a curadoria dos espécimes de macrofungos, existem endpoints que trazem todas as submissões de espécimes efetuadas por todos os usuários do sistema sendo assim o usuário curador pode verificar as fotos submetidas no sistema sendo possível também definir manualmente uma espécie a um espécime submetida;

Tabela 1 – Endpoints utilizado para curadoria

Tipo	Endpoint	Objetivo
GET	/api/curatorship/specimen_submissions/	Retorna todos os envios de espécimes de todos usuários. Utilizado pelo usuário do tipo curador para verificar o envio de outros usuários
POST	/api/curatorship/specimen_submissions/{id}/define_species/	Utilizado pelo usuário do tipo curador para definir de qual espécie é determinado envio (espécime)

Fonte: elaborado pelo autor

- Species (Espécies) - Contém todos os endpoints necessários para edição, exclusão, e busca das espécies cadastradas no banco de dados da aplicação;

Tabela 2 – Endpoints utilizados para exibição de espécies

Tipo	Endpoint	Objetivo
GET	/api/species/	Retorna todas as espécies cadastradas no sistema, pode ser utilizada por usuários de qualquer tipo
GET	/api/species/{id}/	Ao ser passado o id de uma determinada espécie a API retorna as informações da mesma

Fonte: elaborado pelo autor

- Specimen (Espécimes) - É possível trazer as imagens dos espécimes através dos endpoints desse grupo, passando um id de Espécime o sistema retorna uma imagem com o formato JPEG;

Tabela 3 – Endpoints utilizados para exibição de imagem dos espécimes

Tipo	Endpoint	Objetivo
GET	/api/specimen/img/{id}.jpg/	Ao passar o id de um espécime a API retorna a imagem do mesmo

Fonte: elaborado pelo autor

- Specimen Submissions (Envios de Espécimes) - Através dos endpoints desse grupo é possível efetuar o envio dos Espécimes para a análise da rede neural, nesse grupo está o endpoint onde se pode enviar uma imagem para a API que retorna um JSON contendo as espécies sugeridas e suas probabilidades. Também é possível trazer os envios de espécimes do usuário atual;

Tabela 4 – Endpoints para envio e retorno dos espécimes

Tipo	Endpoint	Objetivo
POST	/api/specimen_submissions/reverse_search/	É enviado uma imagem do fungo em formato Base64, ou seja, a imagem é convertida em texto e enviada para o endpoint, a API recebe essa imagem e retorna um JSON com as probabilidades das espécies para aquela imagem
GET	/api/specimen_submissions/	Retorna todos os espécimes ou envios do usuário atual

Fonte: elaborado pelo autor

- Users ( Usuários) - Aqui está disponível toda a parte de acesso dos usuários, é possível realizar o cadastro de novos usuários, efetuar operações de login, logout e recuperação de senha.

Tabela 5 – Endpoints utilizados para gerenciamento de usuários

Tipo	Endpoint	Objetivo
GET	/api/users/	Retorna todos os usuários cadastrados no sistema. Utilizado pelo usuário do tipo administrador para a gerência de usuários
PATCH	/api/users/{id}/update_groups/	Ao enviar o id de um usuário e um novo grupo para o mesmo a API atualiza o grupo daquele usuário
POST	/api/users/login/	Ao enviar o usuário e senha para esse endpoint a API retorna um token para o aplicativo, esse token

		serve para ser enviado no cabeçalho de outras requisições, permitindo o acesso a alguns endpoints.
GET	/api/users/logout/	Ao enviar um token para esse endpoint ele invalida aquele token, dessa forma o usuário precisa logar novamente no sistema para obter um novo token
GET	/api/users/me/	Ao enviar um token para esse endpoint a API retorna informações sobre o usuário, tais como nome, sobrenome, e-mail e tipo de usuário. Utilizado para trazer informações sobre o usuário atual.
POST	/api/users/password_reset/	Ao enviar um usuário a API faz o envio de um e-mail para o cadastro de uma nova senha
POST	/api/users/signup/	Nesse endpoint é efetuado o cadastro de novos usuários, são enviados para o endpoint email, password, nome, e outras informações do usuário e a API faz o cadastro daquele usuário no banco de dados.

Fonte: elaborado pelo autor

## 2.3 IONIC FRAMEWORK

O IONIC foi o framework escolhido para o desenvolvimento do aplicativo pela familiaridade com ele e por possuir uma boa aceitação da comunidade de desenvolvedores, tendo uma ampla documentação disponível e muitos vídeos explicativos que facilitam e ajudam a curva de aprendizagem do desenvolvedor.

Esse framework foi criado em 2013, é gratuito e muito utilizado no desenvolvimento de aplicações para Android, IOS e Web. A escolha desse Framework se passa pela característica de ser um ambiente que permite o desenvolvimento de aplicações híbridas desenvolvidas com tecnologia web, como o HTML5, SCSS e o JavaScript (AngularJS), que se utilizam de uma funcionalidade desses sistemas chamada WebView para apresentar o código web como uma aplicação responsiva para qualquer plataforma, sendo assim, o aplicativo desenvolvido neste trabalho pode ser facilmente portado para outros sistemas operacionais futuramente (IONIC,2022).

O front-end da aplicação é desenvolvido com HTML e SCSS e as partes lógicas com AngularJS. Sendo assim, é possível seguir o padrão de desenvolvimento MVC (SANCHEZ,2018).

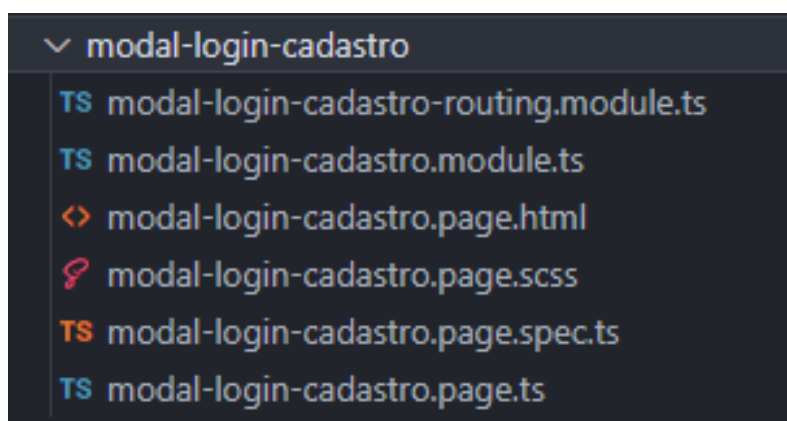
Esse padrão divide o desenvolvimento de software em 3 partes: modelo, visão e controle:

- Modelo: É a camada responsável por carregar os dados. Buscando enviar ou receber informações das bases de dados. O banco de dados do aplicativo como já visto anteriormente está armazenado remotamente sendo acessado através da API.
- Visão: É a parte do código que nos permite apresentar os dados que o modelo fornece, neste caso todo o código HTML que foi desenvolvido e que nos permite mostrar a saída dos dados processados. O IONIC disponibiliza uma gama de componentes tais como plugins, botões, modais, caixas de texto, entre outros. Sendo possível modificar suas propriedades através de suas propriedades SCSS.

- Controle: A camada que serve como um link entre a visão e o modelo. Possibilita o envio de comandos para o modelo e permite atualizar seu status para que a exibição possa alterar sua apresentação.

Na figura 2 é possível verificar a organização do código de uma das classes do aplicativo, com os HTML e SCSS para visão e o TS para controle e modelo.

Figura 2 – Organização básica do código de uma classe em um projeto IONIC.



Fonte: elaborado pelo autor

## 2.4 ANÁLISE DE OUTROS APLICATIVOS

Com o objetivo de avaliar a qualidade, facilidade de uso e eficácia na categorização de espécies, analisando as suas funcionalidades e a interface gráfica de outros aplicativos semelhantes para a finalidade de classificação de fungos, foi realizada uma análise de outros aplicativos concorrentes com base nos protocolos elaborados por Kitchenham (2004). É mostrado abaixo o objetivo da revisão, escopo da pesquisa, estratégias de busca, critérios de inclusão e exclusão para seleção desses aplicativos e a técnica de extração de dados e sumarização de resultados.

- Questão: Quais os principais aplicativos para a identificação e classificação de fungos disponíveis para uso?

- População: Aplicativos para a identificação e classificação de fungos ou aplicativos para classificação de vegetais (onde às vezes aparecem fungos), disponíveis em bibliotecas digitais.
- Intervenção: Análise de aplicativos para identificação de fungos.
- Resultado: Comparação de aplicativos para identificação de fungos.
- Contexto: Biblioteca digital Google Play.

### 2.4.1 Definições da Pesquisa

Os detalhes da definição de pesquisa, como: locais de pesquisa, termos, critérios de inclusão e exclusão, são apresentados no quadro 1.

Quadro 1 – Definições da pesquisa

Bibliotecas pesquisadas	Termos pesquisados
- Google Play;	- "Mushroom Identification Apps"; - "Aplicativos para identificar fungos".
Critérios de inclusão	
- Aplicativos com mais de 100 mil downloads; - Aplicativos com mais de uma versão; - Aplicativos cuja última atualização tenha sido feita entre Março de 2020 até Março de 2022; - Aplicativos para Android; - Aplicativos que consigam efetuar classificação de espécies de fungos através de fotos.	
Critérios de exclusão	
- Aplicativos com menos de 100 mil downloads; - Aplicativos que mostram informações sobre fungos mas que não possuem nenhuma inteligência artificial para identificação dos mesmos; - Aplicativos com menos de 3 estrelas de avaliação; - Aplicativos que não tiram fotos de fungos;	

Fonte: elaborado pelo autor

## 2.4.2 Execução da Pesquisa

Inicialmente a seleção resultou em 60 aplicativos e uma análise por avaliação, tipo do aplicativo e data de atualização resultou em 5 aplicativos selecionados. A execução da busca pode ser vista no quadro 2.

Quadro 2 – Execução da pesquisa

Google Play			
Termos Pesquisados	"Mushroom Identification Apps" ou "Aplicativos para identificar fungos".		
Redefinição da Pesquisa	Downloads - Maior que 100 mil downloads; Avaliação - Aplicativos com mais de 3 estrelas de avaliação; Atualização - Março de 2020 até Março de 2022.		
Quantidade de Resultados	60	Selecionados	5

Fonte: elaborado pelo autor

## 2.4.3 Análise dos aplicativos selecionados

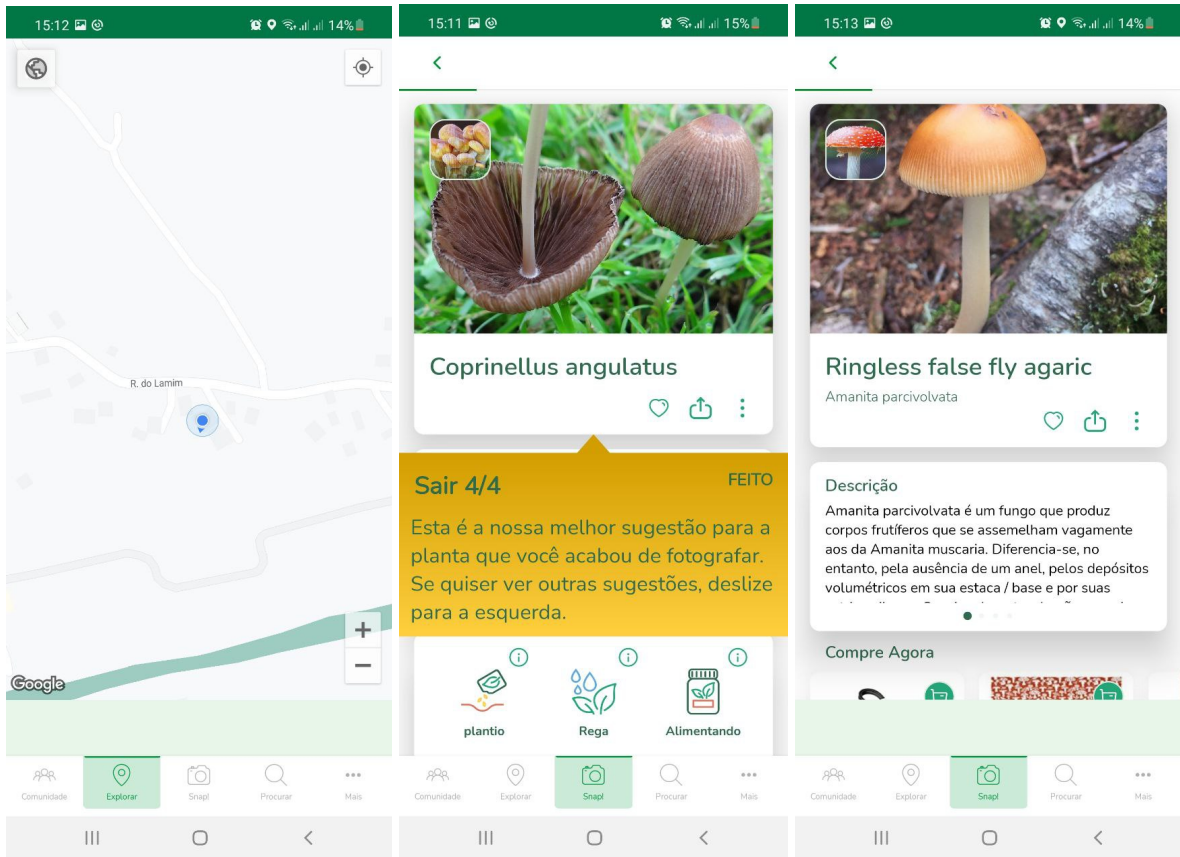
### 2.4.3.1 *PlantSnap*

Com o PlantSnap é possível identificar instantaneamente mais de 600.000 tipos de plantas: flores, árvores, suculentas, cogumelos e cactos. Além disso, o aplicativo mostra uma grande quantidade de informações sobre as espécies e em caso de plantas dicas de cultivo e cuidados. É possível também verificar parques e regiões próximas cadastradas no aplicativo, apesar de também possuir algumas funcionalidades pagas, o que limita o seu uso.

Durante os testes como o PlantSnap foi possível verificar a identificação de duas espécies de fungos, o que mostra que o aplicativo realmente tem alguma inteligência para identificar algumas dessas espécies, apesar de não ser algo voltado somente para isso.



Figura 3 - Imagens de uso do PlantSnap.



Fonte: elaborado pelo autor

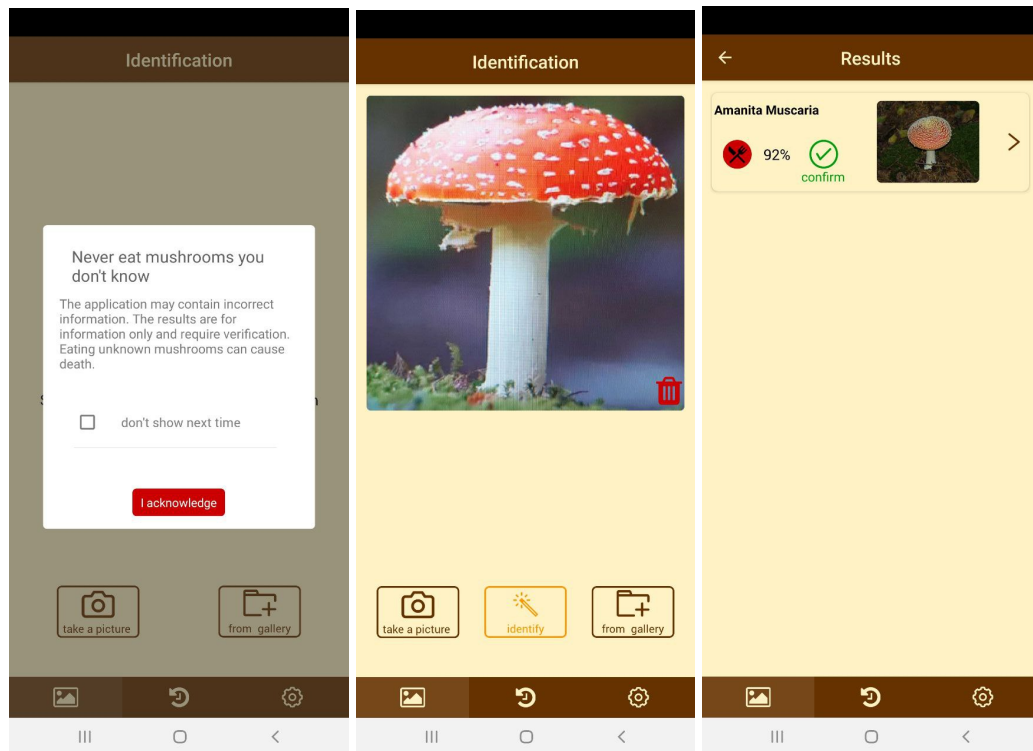
#### 2.4.3.2 Fungus

O aplicativo Fungus parece ser um aplicativo mais simples para uso mais amador, desenvolvido por Dominik Steinhauser o aplicativo usa algoritmos de inteligência artificial e redes neurais onde o cálculo e identificação de espécies ocorre no servidor. Portanto, é necessário ter uma conexão com a Internet para reconhecer os cogumelos. O banco de dados contém atualmente 12.352 cogumelos de 1736 espécies, atingindo precisão superior a 80%.

Para se obter melhores resultados, os cogumelos precisam ser fotografados várias vezes de diferentes perspectivas, para que todas as partes do cogumelo de diferentes ângulos possam ser vistas nas fotos. O aplicativo mantém um histórico de pesquisa e exibe as fotos dos cogumelos reconhecidos.

Possui uma interface simples e intuitiva, e traz algumas informações da Wikipédia sobre as espécies identificadas, não parece haver uma curadoria ampla do banco de dados de espécies parecendo ser um aplicativo desenvolvido por uma única pessoa, com o uso de um banco de dados genérico e público.

Figura 4 - Imagens de uso do Fungus.



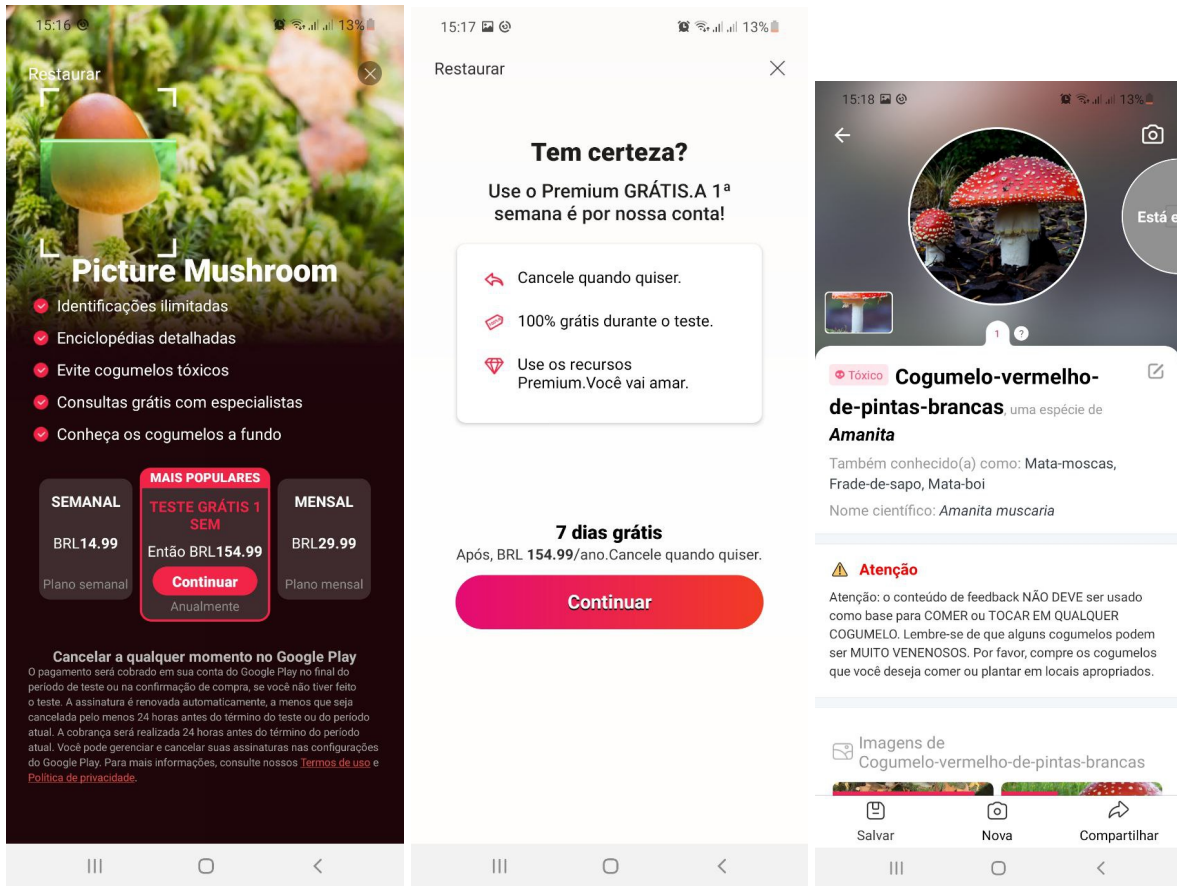
Fonte: elaborado pelo autor

### 2.4.3.3 *Picture Mushroom*

Picture Mushroom é um aplicativo voltado para identificação de fungos, possui somente a possibilidade de 7 dias de uso gratuitamente, após esse período o usuário precisa assinar algum dos planos oferecidos, apresenta bastante propagandas para que o usuário pague pelo aplicativo, o que dificulta o uso do mesmo e polui a tela com anúncios, conseguiu reconhecer um Amanita muscaria que foi testado e apresentou informações gerais sobre o fungo. Interessante a maneira que o aplicativo informa para o usuário em qual perspectiva a foto deve

ser tirada, apresentando um desenho do fungo como marca d'água na posição que o aplicativo deseja que a foto seja tirada.

Figura 5 - Imagens de uso do Picture Mushroom.

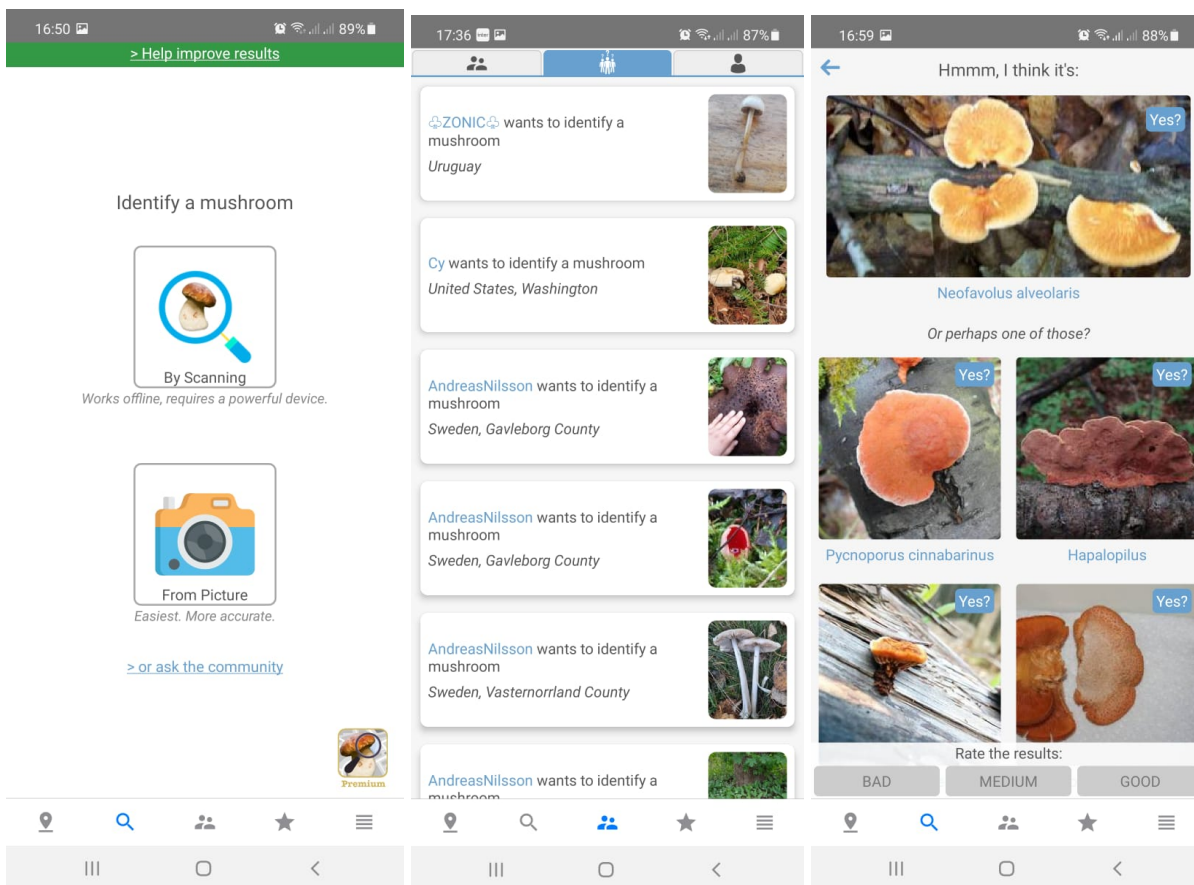


Fonte: elaborado pelo autor

#### 2.4.3.4 Mushroom Identifier

Esse aplicativo possui capacidade de reconhecer pelo menos 900 espécies de fungos. Inclui também ferramentas de mapa para busca de espécies de fungos próximas ao local onde você se encontra e também possui um fórum onde os usuários podem entrar em contato para tirar dúvidas e postar fotos de fungos. Possui uma versão premium e uma versão free que se utiliza de ads para gerar receita.

Figura 6 - Imagens de uso do Mushroom Identifier.

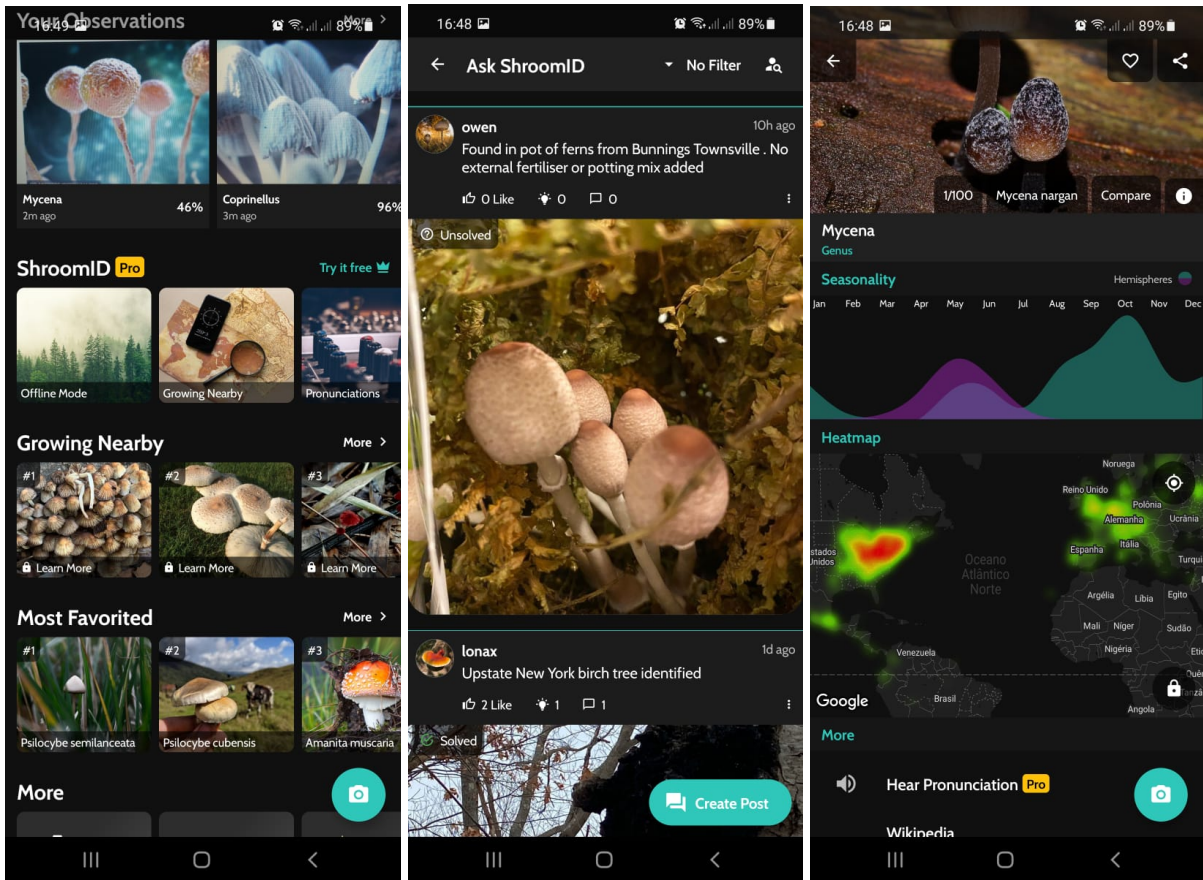


Fonte: elaborado pelo autor

#### 2.4.3.5 ShroomID

Esse é um aplicativo bem completo para identificação de fungos e possui algumas características de interface e design bem interessantes e não encontradas em outros aplicativos, o mapa de calor das espécies com as principais localidades que elas foram encontradas além da sazonalidade das espécies são mostradas de formas rápidas após a identificação dos fungos. Possui também uma boa comunidade de pessoas que utilizam o aplicativo e se comunicam através de um fórum, ali são postadas dúvidas e respostas para algumas fotos tiradas. O grande problema desta aplicação também é a questão de possuir várias funcionalidades pagas, a identificação de fungos de maneira offline por exemplo, só é possível na versão Premium que custa 27.99 dólares.

Figura 7 - Imagens de uso do ShroomID.



Fonte: elaborado pelo autor

#### 2.4.4 Comparativo e conclusões sobre e os aplicativos selecionados

Após a análise dos principais aplicativos para identificação de fungos, podemos concluir que eles oferecem uma ampla variedade de recursos e características úteis para os usuários. Todos eles permitem que você tire uma foto do fungo e o compare com imagens em sua base de dados para tentar identificá-lo, possuindo uma ampla variedade de informações sobre o hábitat e comestibilidade dos fungos, bem como glossários de termos relacionados a fungos.

Embora cada aplicativo tenha suas próprias vantagens e desvantagens, todos eles podem ser úteis para a identificação de fungos. No entanto, é importante lembrar que esses aplicativos são apenas ferramentas de auxílio e não devem ser usados como substitutos de um especialista em fungos qualificado, todos possuem sugestões de identificação que podem conter erros.

Apesar da grande quantidade desses aplicativos existem ainda poucas opções gratuitas, sendo a maioria dos aplicativos analisados pagos ou com várias propagandas que ajudam as empresas a viabilizarem financeiramente esses projetos. Durante o uso desses aplicativos isso foi um fator que se apresentou de maneira negativa pois muitas vezes as funcionalidades ficam escondidas ou mais difíceis de acessar devido a grande quantidade de propagandas que se apresentam.

É possível concluir também que existem aplicativos que apresentam uma maior quantidade de espécies em seu banco de imagens quando comparadas às 423 espécies levantadas pelo grupo MIND. Funga, como é o caso do Fungus e do Mushroom Identifier.

Na tabela 6 é possível verificar uma tabela comparativa com as principais características dos aplicativos analisados:

Tabela 6 - Comparativo entre os aplicativos selecionados.

<b>Nome</b>	<b>Versão</b>	<b>Última Atualização</b>	<b>Nº de Downloads</b>	<b>Quantidade de Espécies</b>	<b>É gratuito?</b>
PlantSnap	5.01.02	28/02/ 2022	10,000,000+	-	Não
Fungus	1.0.4	23/11/2020	100,000+	1736	Sim
Picture Mushroom	2.8.10	11/03/2022	1,000,000+	-	Não
Mushroom Identifier	2.66	08/11/2021	1,000,000+	900	Não
ShroomID	2.2.3	12/03/2022	100,000+	-	Não

Fonte: elaborado pelo autor

### 3 DESENVOLVIMENTO DO APLICATIVO

São apresentados neste capítulo os processos de desenvolvimento utilizados para a criação do aplicativo, entre eles, criação de protótipos e interfaces, testes, entrevistas, levantamento de requisitos, comparação com concorrentes e a programação do mesmo. Foram utilizadas para isso algumas das sugestões apresentadas por SILVA (2019).

#### 3.1 ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE

A especificação dos requisitos é um dos primeiros passos para o desenvolvimento de uma aplicação. Para executá-la foram feitas reuniões bissemanais com o grupo de pesquisa MIND.Funga, sendo possível verificar as necessidades do grupo e identificar as atividades a serem desenvolvidas para o aplicativo. Abaixo estão os requisitos levantados:

##### 3.1.1 Requisitos funcionais

Os requisitos funcionais do aplicativo são os seguintes:

- RF01 – Efetuar Login: O aplicativo deve apresentar a opção de menu “Login” para a identificação de usuários no sistema;
- RF02 – Identificar Fungo: O aplicativo deve apresentar a opção de menu “Que fungo é esse?” para que os usuários possam tirar fotos e identificar os fungos;
- RF03 – Mostrar mapa de registros: O aplicativo deve apresentar a opção de menu “Mapa de Registros” para que os usuários possam ver os fungos identificados em mapa;

- RF04 – Ver Espécies: O aplicativo deve apresentar a opção de menu “Espécies” para que os usuários possam ver informações sobre as espécies cadastradas no sistema, ele poderá ordenar as espécies de maneira alfabética e por categoria .
- RF05 – Editar Espécie: O usuário poderá editar uma espécie já cadastrada, clicando sobre o nome da espécie, caso ele seja curador.
- RF06 – Cadastrar Espécie: Usuários do tipo curador podem cadastrar uma nova espécie, para tanto devem inserir as seguintes informações: Taxonomia, Autor, Família, Ordem, Filo, Etimologia, Curiosidade, Nota sobre a lista vermelha IUCN, Fotos do fungo e Observações gerais.
- RF07 – Proceder Curadoria: O aplicativo deve apresentar a opção de menu “Curadoria” para que os usuários do tipo Curador possam colaborar com correções e edições das informações cadastradas, ao salvar e confirmar as edições a espécie será inserida na lista de espécies públicas. O usuário poderá fazer a revisão e gestão de fotos das espécies cadastradas.
- RF08 – Gerenciar usuários: Usuários do tipo Administrador poderão fazer o gerenciamento de usuários através da opção de menu "Gerenciar Usuários" . Deverá ser possível listar todos os usuários do sistema e efetuar a edição das informações desses usuários



- RF09 – Mostrar informações sobre o MIND.Funga: O aplicativo deve apresentar a opção de menu “Sobre o MIND.Funga” para que os usuários possam ver a versão do aplicativo e obter informações sobre a iniciativa MIND.Funga, além de informações sobre as redes sociais do projeto.
- RF10 – Mostrar dicas de fotografia: O aplicativo deve apresentar a opção de menu “Dicas de fotografia” para que os usuários possam ver as melhores maneiras de efetuar as fotografias dos fungos.
- RF11 – Ver Álbum: Mostra os espécimes enviados pelo usuário com informações sobre data, substrato, observações e localização do espécime enviado.

### **3.1.2 Requisitos não funcionais**

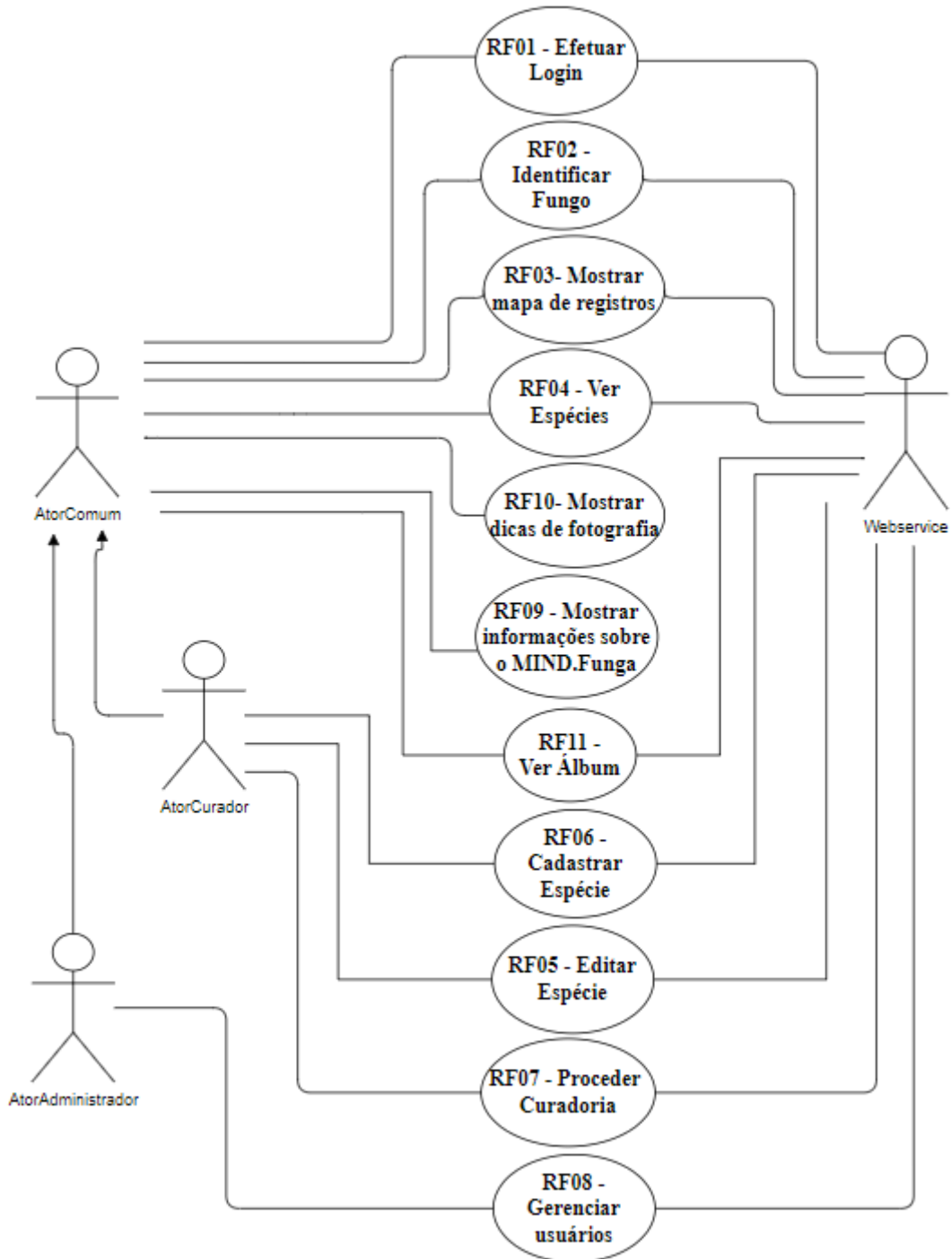
Os requisitos não funcionais do aplicativo são os seguintes:

- RNF 01 – Identidade visual do App MIND.Funga: A interface gráfica deve seguir o padrão desenvolvido por Nathalie Ferreira;
- RNF 02 – Linguagem da programação: O programa será implementado em uma linguagem que permita a execução em dispositivos que possuam sistema operacional Android.

### 3.2 CASOS DE USO

Buscando apresentar de maneira gráfica as principais funcionalidades do aplicativo e as suas interações com os atores. Assim temos que os casos de uso são:

Figura 8 - Diagrama de casos de uso do Aplicativo



Fonte: elaborado pelo autor

Para o caso de uso *RF01 – Efetuar Login* devem ser consideradas as seguintes restrições:

- A operação envia o login e senha do usuário para um endpoint da API, recebendo uma confirmação caso o usuário seja válido;
- Existem 3 tipos de usuários: Comum, Curador, Administrador ;
- Caso o usuário seja do tipo Comum ele irá ver somente as opções: Login/Logout, Que fungo é esse?, Meu Álbum, Espécies, Dicas de fotografia e Sobre o MIND.Funga;
- Caso o usuário seja do tipo Curador ele irá ver todas as opções do usuário Comum mais a opção de Curadoria, adição e edição de espécies;
- Caso o usuário seja do tipo Administrador ele irá ver todas as opções do usuário Comum mais a opção de Gerenciar usuários;
- Caso o operador não possua um login ele poderá efetuar o cadastro preenchendo informações como Nome, sobrenome, e-mail, estado, e uma checkbox para armazenar se o usuário é especialista e deseja colaborar com a curadoria de imagens do aplicativo, senha e confirmação de senha

- Caso o operador não estiver online ele poderá armazenar as fotos em seu dispositivo e quando ele ficar online poderá efetuar o login e subir as fotos para o servidor
- Caso o operador não souber a sua senha, ele deverá ser encaminhado para uma tela de redefinição de senha onde irá inserir o seu e-mail, para que um e-mail de redefinição de senha seja enviado para ele

Para o caso de uso *RF02 – Identificar Fungo* devem ser consideradas as seguintes restrições:

- Ao clicar no menu deverá aparecer uma modal com Dicas importantes para que o usuário possa tirar as fotos da melhor maneira;
- O usuário pode tirar uma foto e a mesma irá para o seu Álbum, ou fazer o upload da imagem.
- Ao clicar em tirar foto o aplicativo deve solicitar acesso ao GPS e a câmera do dispositivo.
- O usuário poderá classificar em qual ângulo a foto foi tirada, o usuário pode incluir informações complementares como tipo de Substrato, Localização( que será preenchida automaticamente mas poderá ser editada), data da foto, e observações gerais .
- O usuário pode inserir mais de uma imagem da mesma espécie em diversos ângulos diferentes para facilitar a identificação do fungo .



### 3.4 PROTÓTIPOS DE INTERFACE

Para o desenvolvimento do aplicativo foi necessário a elaboração de alguns protótipos buscando desenvolver uma base visual para o aplicativo final, abaixo estão disponíveis algumas idéias de interfaces que foram elaboradas para esse trabalho.

Na figura 10 pode-se verificar o protótipo desenvolvido para a tela inicial do aplicativo, é possível observar a identidade visual pensada.

Figura 10 - Protótipo da tela inicial do sistema



Fonte: MIND.Funga

Na figura 11 é possível verificar o protótipo do menu que irá aparecer para o usuário comum, os perfis de usuário são descritos no requisito funcional 01.

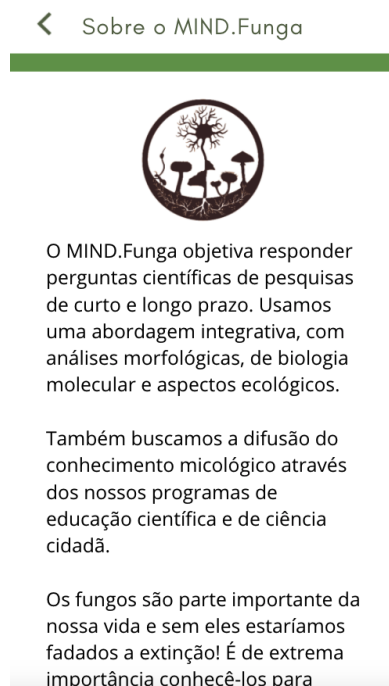
Figura 11 - Protótipo do menu do usuário comum / RF01



Fonte: MIND.Funga

Na figura 12 está o protótipo desenvolvido para o requisito funcional 09 a tela que mostra mais informações sobre o grupo MIND. Funga.

Figura 12 - Protótipo da tela contendo informações sobre o MIND. Funga / RF09



Fonte: MIND.Funga

Na figura 13 é possível observar um protótipo para o requisito funcional 01 a tela de login do aplicativo.

Figura 13 - Protótipo da tela de login no aplicativo / RF01



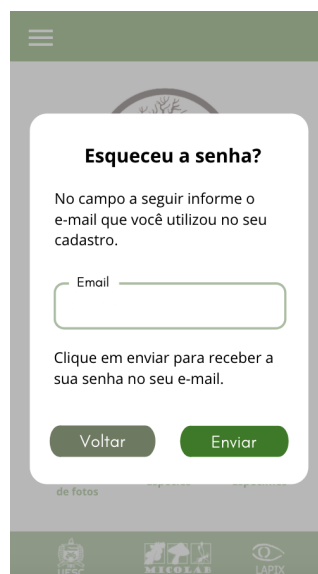
O protótipo da tela de login do aplicativo MIND.Funga apresenta o seguinte layout:

- Barra superior verde com ícone de menu (três linhas horizontais).
- Logo circular com uma árvore e cogumelos, com o texto "MIND.Funga" abaixo.
- Campos de entrada para "Email" e "Senha".
- Link "Esqueceu sua senha?".
- Botão verde "Acessar".
- Link "CADASTRE-SE".
- Barra inferior com ícones de parceiros: UFSC, MEGOLAB e LAPIX.

Fonte: MIND.Funga

Na figura 14 é possível observar um protótipo para a tela de recuperação de senha, também pensado no requisito funcional 01.

Figura 14 - Protótipo de recuperação de senha / RF01



O protótipo da tela de recuperação de senha do aplicativo MIND.Funga apresenta o seguinte layout:

- Barra superior verde com ícone de menu (três linhas horizontais).
- Logo circular parcialmente visível no topo.
- Título "Esqueceu a senha?".
- Texto explicativo: "No campo a seguir informe o e-mail que você utilizou no seu cadastro."
- Campo de entrada para "Email".
- Texto explicativo: "Clique em enviar para receber a sua senha no seu e-mail."
- Botões "Voltar" e "Enviar".
- Barra inferior com ícones de parceiros: UFSC, MEGOLAB e LAPIX.

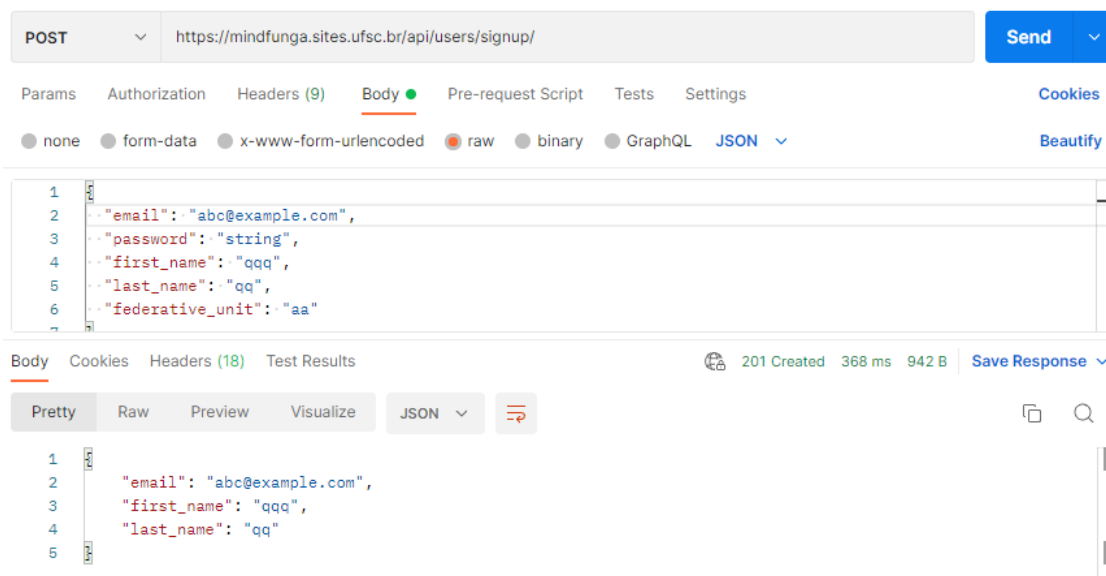
Fonte: MIND.Funga



### 3.5 TESTES NA API

Antes de começar a parte de programação do aplicativo foi necessário uma etapa de testes na API existente, tendo em vista que essa API é um trabalho ainda em desenvolvimento de um membro do grupo MIND.Funga e portanto poderia apresentar alguns erros. A ideia desses testes foi sugerir melhorias na API e auxiliar no processo de resolução de problemas existentes. Foram efetuadas requisições de testes em todos os endpoints que seriam utilizados pelo aplicativo buscando verificar o seu correto funcionamento. Para isso foi utilizado um software chamado Postman, que permite executar requisições HTTP e verificar a resposta da API. É possível verificar na figura 15 o exemplo de um teste efetuado no endpoint “signup”, responsável pelo cadastro de novos usuários, repare que no Body(corpo) dessa requisição são passados os dados desse novo usuário e a API nesse caso retornou a resposta 201, ou seja, o usuário foi criado com sucesso. A API também retorna na resposta um json contendo email, nome e sobrenome do usuário criado.

Figura 15 - Testes de envios de pacotes para a API utilizando o POSTMAN



Fonte: elaborado pelo autor

Para todos os endpoints utilizados pelo aplicativo e apresentados na seção 2.2.2 deste trabalho, foram feitas simulações de envios de pacotes bem como a verificação das respostas retornadas pela API.

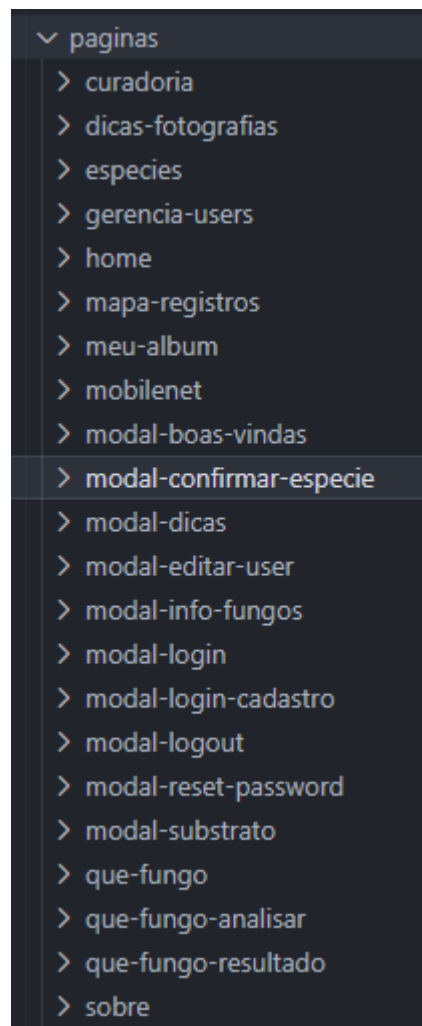
Antes do início do processo de programação do aplicativo foram informados problemas em alguns endpoints visando a correção de bugs, isso diminuiu a necessidade de correção durante o desenvolvimento do aplicativo.

### 3.6 PROGRAMAÇÃO DO APLICATIVO

Para a programação do aplicativo se utilizou o framework IONIC citado anteriormente. Foi necessário o conhecimento em HTML, SCSS e programação em Angular para efetuar a elaboração das telas e lógicas presentes no aplicativo.

Cada tela que aparece no aplicativo representa uma página dentro do projeto do aplicativo, para criar essas páginas se executa o comando “ionic g page paginas/nome\_da\_pagina” na raiz do projeto através do terminal do sistema operacional utilizado para a programação. Ao executar esse comando uma nova página é criada dentro da pasta src/app/paginas, na figura 16 é possível ver todas as páginas que foram criadas neste trabalho:

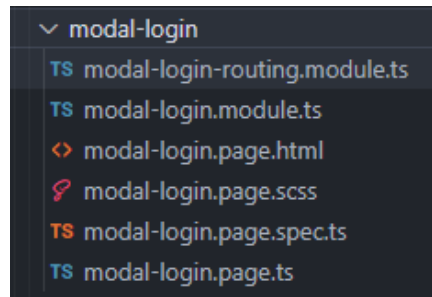
Figura 16 - Listas de páginas criada para o aplicativo



Fonte: elaborado pelo autor

Cada página possui uma estrutura padrão gerada pelo framework quando a mesma é inserida no projeto. Na figura 17 possível verificar a estrutura criada, nesse exemplo é utilizada a tela de login, RF01 do sistema para a demonstração:

Figura 17 - Estrutura de uma página criada no IONIC



Fonte: elaborado pelo autor

Os arquivos mais importantes nessa estrutura e que contém a lógica programada para a exibição da interface do aplicativo que é vista pelo usuário são os seguintes:

- modal-login.page.scss - Nesse arquivo é onde são configuradas as propriedades visuais dos elementos que são vistos nas páginas, é possível efetuar configurações da cor da borda do elemento, cor de fundo, cor do texto, espaçamento do componente em relação a tela, entre outros.

Figura 18 - Arquivo modal-login.page.scss

```
.roundedInput {
  --border-color: var(--ion-color-medium-shade);
  --border-radius: 3px;
  --border-width: 1px;
  --box-shadow: 2px 2px gray;
  --highlight-height: 0;
  --background: #f8f9fa;
  --color: black;
  padding: 5px;
}

.textEsqueceu {
  color: #498521;
  padding-left: 10px;
  padding-top: 0px;
}

.erro_msg {
  color: #FF0000;
  padding-left: 10px;
  padding-top: 5px;
}

.textCadastro {
  color: #498521;
}

div{
  margin: 5%;
}
```

Fonte: elaborado pelo autor

- modal-login.page.html - Apresenta toda a descrição dos elementos que aparecem para o usuário, como botões, textos, e outras informações que aparecem na tela. Na figura 19 é possível verificar o código HTML da tela de login.

Figura 19 - Arquivo modal-login.page.html

```

<ion-header>
</ion-header>

<ion-content>
  <div id="container">
    <p align="center"></p><br>
    <!-- <p align="center"><ion-button color="danger" (click)="dismiss()">fechar</ion-button-->

    <form (ngSubmit)="loginSubmit()">
      <ion-item class="roundedInput">
        <ion-label position="stacked">E-mail</ion-label>
        <ion-input type="text" [(ngModel)]="todo.email" name="E-mail"></ion-input>
      </ion-item>
      <ion-item class="roundedInput">
        <ion-label position="stacked">Senha</ion-label>
        <ion-input type="password" [(ngModel)]="todo.password" name="Senha"></ion-input>
      </ion-item>
      <ion-text class="erro_msg" *ngIf="this.loginService.erro_msg != null" >{{this.loginService.erro_msg}}</ion-text><br><br>
      <ion-text class="textEsqueceu" (click)="esqueceuSenha()">Esqueceu sua senha?</ion-text><br>
      <p align="center"><ion-button type="submit" block>Acessar</ion-button></p>
      <p align="center"><ion-text class="textCadastro" (click)="cadastrarUsuario()">Cadastre-se</ion-text></p>
    </form>
  </div>
</ion-content>

```

Fonte: elaborado pelo autor

É possível ver que existem elementos que foram inseridos nesse arquivo que possibilitam que o usuário digite um e-mail e uma senha, esses itens são representados com a tag ion-input. A tag ion-label é responsável pela exibição de um texto que identifica aquele campo de entrada de dados. A tag ion-item determina que aqueles elementos em conjunto fazem parte da classe "roundedInput" configurada no arquivo .scss, sendo assim, eles irão ter as propriedades gráficas definidas nesta classe.

Figura 20 - Elementos de e-mail e senha da tela de login

```

<ion-item class="roundedInput">
  <ion-label position="stacked">E-mail</ion-label>
  <ion-input type="text" [(ngModel)]="todo.email" name="E-mail"></ion-input>
</ion-item>
<ion-item class="roundedInput">
  <ion-label position="stacked">Senha</ion-label>
  <ion-input type="password" [(ngModel)]="todo.password" name="Senha"></ion-input>
</ion-item>

```

Fonte: elaborado pelo autor

Os elementos de input de usuário e senha vistos na figura 20 estão em um formulário, esses dados serão tratados pela função `loginSubmit()`.

Esses dados são enviados quando o usuário clicar no botão `Acessar` representado pela tag `ion-button`, que é do tipo `Submit`.

Figura 21 - Elemento botão do tipo submit que está na tela de login

```
<ion-button type="submit" block>Acessar</ion-button></p>
```

Fonte: elaborado pelo autor

Textos clicáveis para que o usuário possa se cadastrar no aplicativo e reiniciar a sua senha são apresentados com a tag `ion-text`, é possível perceber que existe eventos de click que possibilitam o redirecionamento para funções `cadastroUsuário()` e `esqueceuSenha()` quando o usuário efetua o click nesses textos.

Figura 22 - Elemento do tipo texto Esqueceu sua Senha

```
<ion-text class="textEsqueceu" (click)="esqueceuSenha()">Esqueceu sua senha?</ion-text>
```

Fonte: elaborado pelo autor

- `modal-login.page.ts` - É o arquivo que possui as funções e variáveis que são utilizadas pela página, aqui estão presentes as funções que fazem a comunicação com os endpoints da API, também é possível criar funções que mudam variáveis utilizadas no HTML, dessa maneira é possível apresentar resultados na tela, ou mesmo ler textos digitados pelo usuário.

Na figura 23 é possível verificar a função `loginSubmit()` presente nesse arquivo. Ela chama outra função chamada `login()` que efetua a chamada HTTP para o endpoint de login da API e em caso de uma resposta bem sucedida o sistema encaminha os dados da resposta para a função `verificarToken()` passando o token gerado como parâmetro, caso a resposta da API seja algum erro, a função `verificarToken` é chamada com parametro `null`.

Figura 23 - Função loginSubmit()

```
public loginSubmit(){
  this.loginService.login(this.todo).then(data => this.verificarToken(data)).catch(error=> this.verificarToken(null));
}
```

Fonte: elaborado pelo autor

Na função verificarToken() ocorre a mudança de variáveis e elementos que são mostradas na tela de login, por exemplo, em caso de erro o sistema mostrará uma mensagem de erro no login, já em caso de login bem sucedido existe o salvamento do token gerado, e a configuração de algumas variáveis que são utilizadas pelo aplicativo.

Figura 24 - Função verificarToken()

```
public verificarToken(data){
  if(data==null){
    this.loginService.logou=false;
    this.loginService.erro_msg= "Erro de login, tente novamente.";
    this.loginService.token = null;
  }
  else{
    this.loginService.logou=true;
    this.loginService.erro_msg=null;
    this.loginService.token=data['token'];
    this.pegarInformacoesUsuario();
    this.modalCtrl.dismiss();
  }
}
```

Fonte: elaborado pelo autor

Esse tipo de lógica de programação utilizada como exemplo para a demonstração do desenvolvimento do aplicativo, é utilizada por outras páginas do sistema, podendo então resumir de maneira geral a divisão entre os elementos da seguinte maneira:

- Arquivos HTML exibem os elementos na tela
- Arquivos SCSS configuram propriedades gráficas dos elementos apresentados no HTML

- Arquivos TS efetuam a mudança de estado e valores de elementos da tela e fazem a comunicação com a API.

Na figura 25 é possível verificar como a tela utilizada como exemplo é apresentada para o usuário, é possível ver os elementos apresentados no código HTML bem como a parte visual devidamente configurada pelo arquivo SCSS.

Figura 25 - Tela de login programada / RF01



MIND.Funga

E-mail

Senha

Esqueceu sua senha?

ACESSAR

Cadastre-se

Fonte: elaborado pelo autor



## 4 RESULTADOS ALCANÇADOS

Ao longo do desenvolvimento deste trabalho foi possível efetuar o desenvolvimento do aplicativo solicitado pelo grupo MIND.Funga e a realização de testes com os membros dessa equipe, foram feitas revisões no aplicativo desenvolvido buscando se adequar aos feedbacks fornecidos pelos componentes do grupo. É possível observar a adequação aos elementos de interface pensados pelo grupo bem como a adequação aos casos de usos pensados.

### 4.1 APLICATIVO DESENVOLVIDO

Abaixo são apresentadas algumas das telas e fluxos desenvolvidos para esse trabalho. Para os testes do aplicativo foi gerado um arquivo .apk que não foi colocado em lojas de aplicativos, mas foi utilizado para a instalação em dispositivos Android por convidados do MIND.Funga.

Na tela inicial do aplicativo mostrada na figura 26, é possível ver o botão de fazer login sendo apresentado no centro da tela, nos protótipos essa tela apresentava o botão para analisar fungos no centro, porém analisando a API desenvolvida foi verificado que o usuário deveria fazer login para gerar um token antes de usar o endpoint que faz a identificação do fungo, portanto, foi colocado um botão para que o usuário do aplicativo possa fazer o login antes de enviar imagem para o sistema.

Figura 26 - Tela inicial do aplicativo



Fonte: elaborado pelo autor

Ao clicar no botão de login, o usuário é encaminhado para uma tela onde ele pode digitar o seu e-mail e a senha para ser validado pela API. A tela de login pode ser vista na Figura 27.

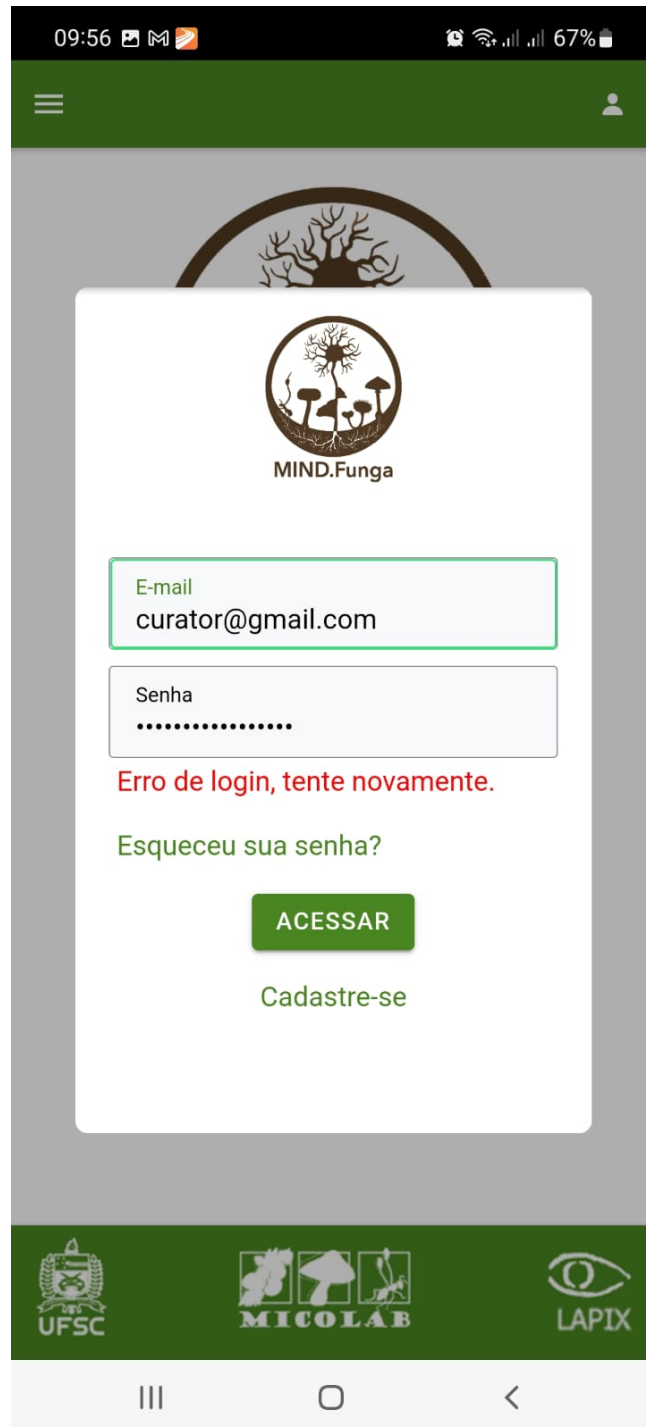
Figura 27 - Tela de login do aplicativo / RF01



Fonte: elaborado pelo autor

Em caso de erro no login o sistema mostra uma mensagem de erro, informando ao usuário que algum problema aconteceu durante o login. Conforme pode ser visto na figura 28.

Figura 28 - Mensagem de erro ao logar / RF01



Fonte: elaborado pelo autor

Em caso de sucesso o sistema mostra uma mensagem de boas vindas ao usuário, conforme pode ser visto na figura 29.

Figura 29 - Tela de boas vindas no sistema / RF01

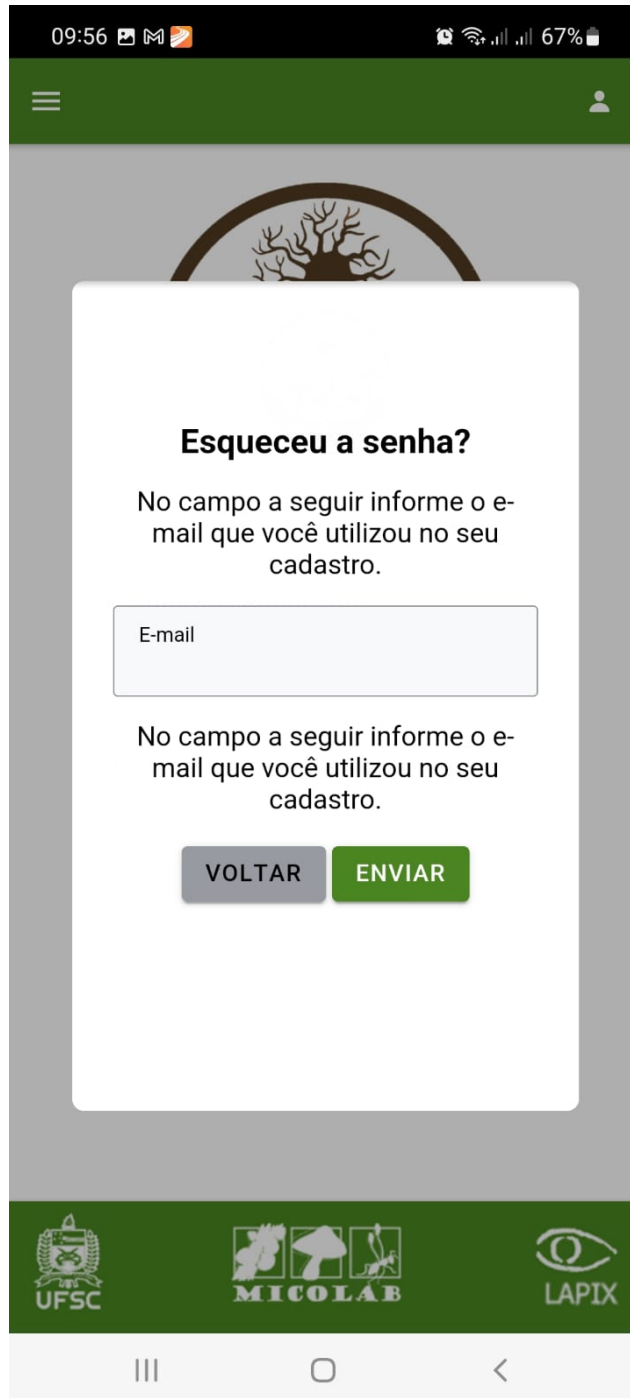


Fonte: elaborado pelo autor

Em caso de esquecimento da senha o usuário também pode solicitar a criação de uma nova senha através do botão Esqueceu sua senha na tela de login,

essa tela apresenta a possibilidade do usuário digitar o e-mail cadastrado no sistema, conforme pode ser visto na figura 30.

Figura 30 - Tela de reset de senha / RF01



09:56 [ícones] 67%

☰ [ícone de usuário]

**Esqueceu a senha?**

No campo a seguir informe o e-mail que você utilizou no seu cadastro.

E-mail

No campo a seguir informe o e-mail que você utilizou no seu cadastro.

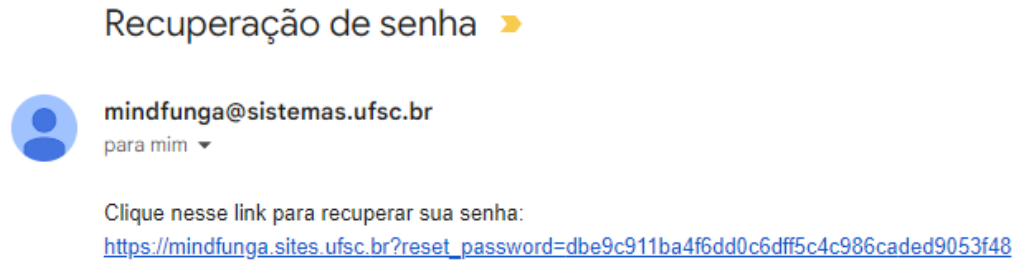
VOLTAR ENVIAR

UFSC MICOLAB LAPIX

Fonte: elaborado pelo autor

Ao enviar a solicitação de reset de senha com um e-mail válido o sistema encaminha uma mensagem para o usuário permitindo que o mesmo efetue o cadastro de uma nova senha. Conforme visto na figura 31.

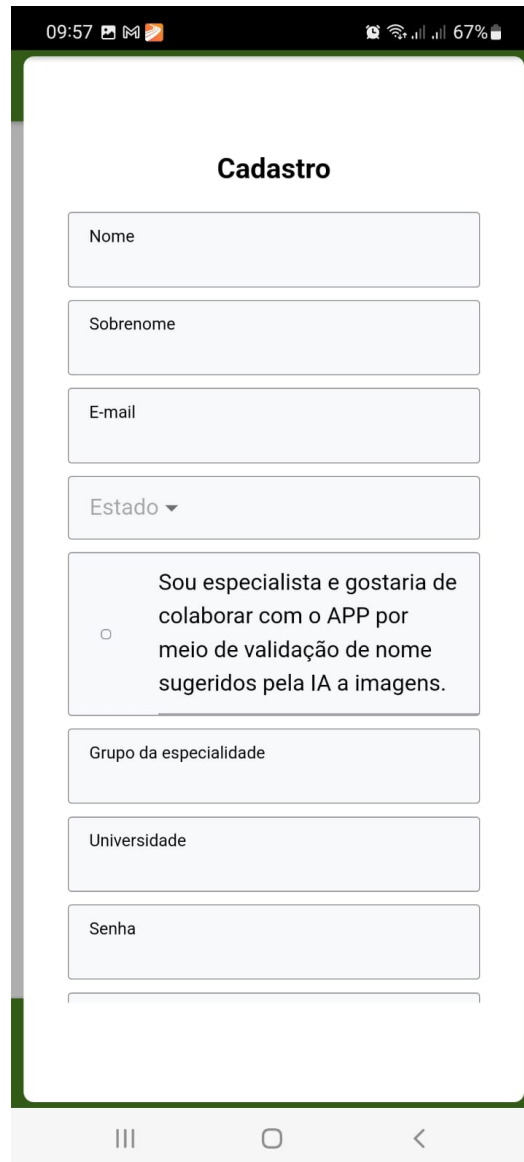
Figura 31 - E-mail para recuperação de senha / RF01



Fonte: elaborado pelo autor

Também é possível efetuar um cadastro de um novo usuário no sistema, caso o mesmo não possua um login, para isso ele precisa preencher algumas informações como, nome, sobrenome, e-mail entre outras. Na figura 32 é possível ver a tela de cadastro de um novo usuário. Em caso de sucesso no cadastro, o sistema envia um e-mail para o usuário confirmando que o registro está completo e que ele já pode efetuar o login no aplicativo.

Figura 32 - Tela de cadastro de usuário / RF01



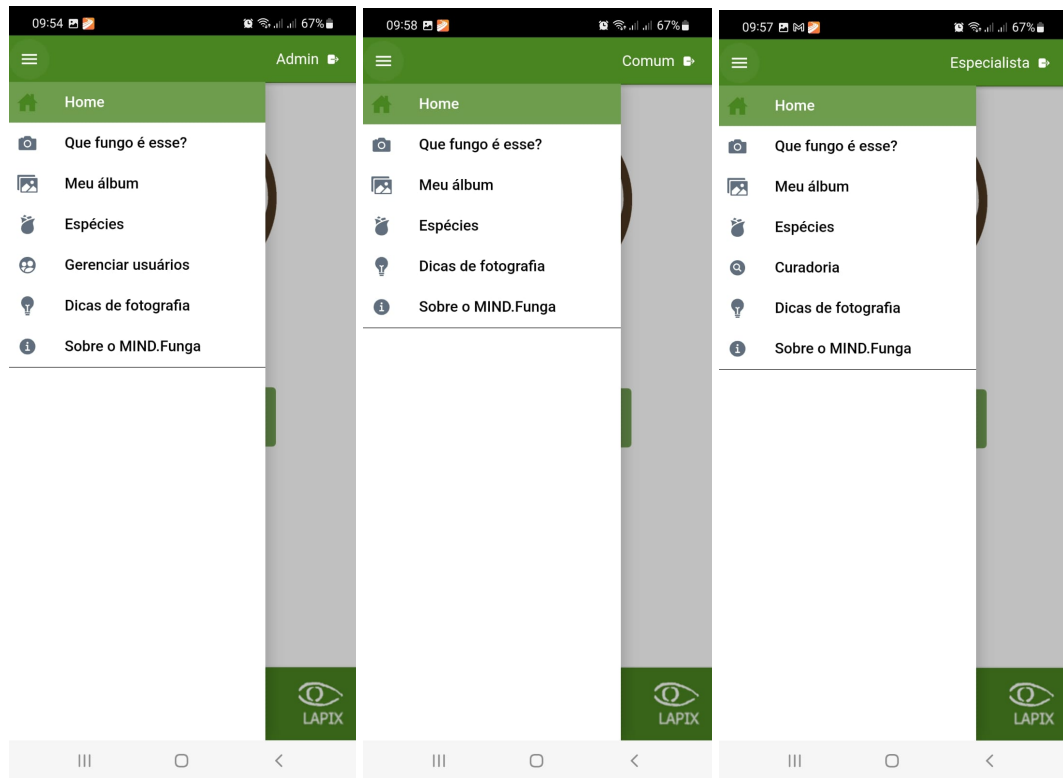
The image shows a mobile application registration screen titled "Cadastro". The screen is framed by a green border. At the top, the status bar shows the time 09:57, signal strength, Wi-Fi, and battery level at 67%. The registration form consists of several input fields: "Nome", "Sobrenome", "E-mail", and "Estado" (with a dropdown arrow). Below these is a checkbox labeled "Sou especialista e gostaria de colaborar com o APP por meio de validação de nome sugeridos pela IA a imagens." followed by "Grupo da especialidade", "Universidade", and "Senha". There is also an empty input field at the bottom of the form. The bottom of the screen shows the standard Android navigation bar with three icons: a home button, a square button, and a back arrow.

Fonte: elaborado pelo autor

Dependendo do tipo de usuário logado, o sistema apresenta um menu lateral com opções diferentes, cada perfil de usuário pode efetuar ou não determinadas operações, o usuário curador ou especialista pode efetuar a curadoria dos espécimes submetidos pelos usuários, o usuário do tipo administrador pode efetuar o gerenciamento de usuário e para o usuário do tipo comum não é apresentado essas opções. Na figura 33 é apresentado os menus para os diferentes tipos de usuários.



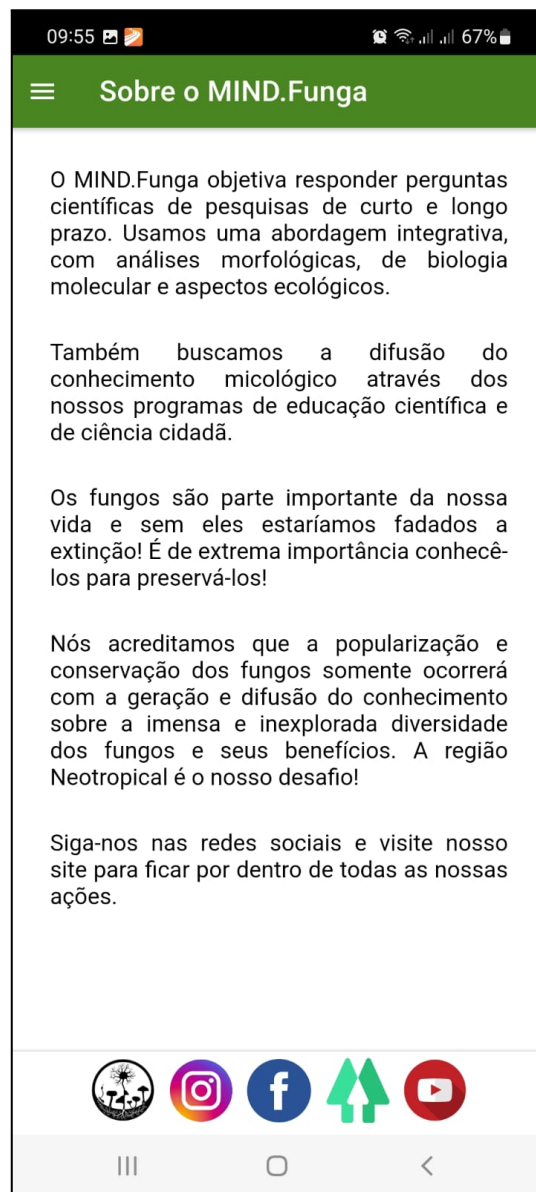
Figura 33 - Menus laterais do aplicativo dado o tipo de usuário / RF01



Fonte: elaborado pelo autor

O usuário do aplicativo pode verificar mais informações sobre o grupo MIND.Funga, é possível clicar nos ícones das redes sociais do grupo para verificar as páginas do grupo. Para isso ele deve acessar o menu com a opção Sobre o Mind.Funga, essa tela é mostrada na figura 34.

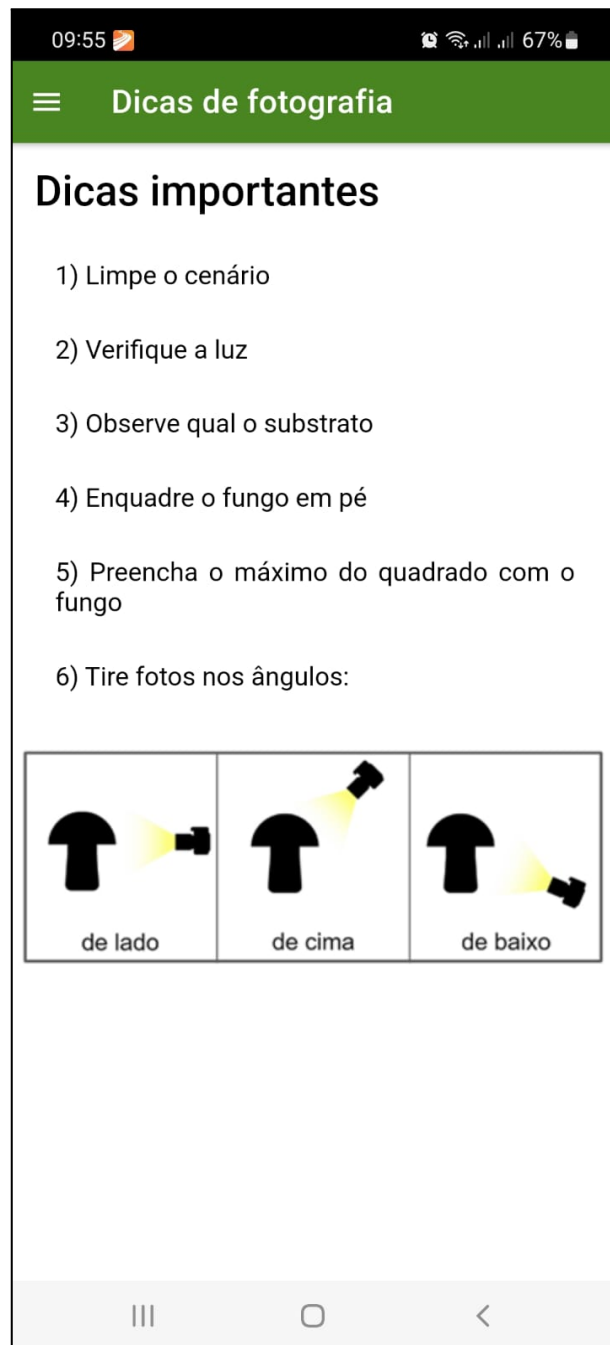
Figura 34 - Tela com mais informações sobre o grupo MIND.Funga / RF09



Fonte: elaborado pelo autor

Ao clicar em Dicas de Fotografia o usuário é encaminhado para uma tela que mostra algumas informações que devem ser seguidas na hora de fotografar os fungos, ao seguir essas dicas o usuário terá melhores resultados na identificação do fungo. Na figura 35 é possível ver a tela que mostra as dicas de fotografia.

Figura 35 - Tela com dicas para a fotografia do fungo / RF10



Fonte: elaborado pelo autor

Todos os usuários logados no aplicativo podem clicar no menu Meu álbum que apresenta todos os envios feitos por esse usuário, é possível efetuar a busca

desses envios por data, tipo de substrato, localização, entre outros. A figura 36 apresenta essa tela.

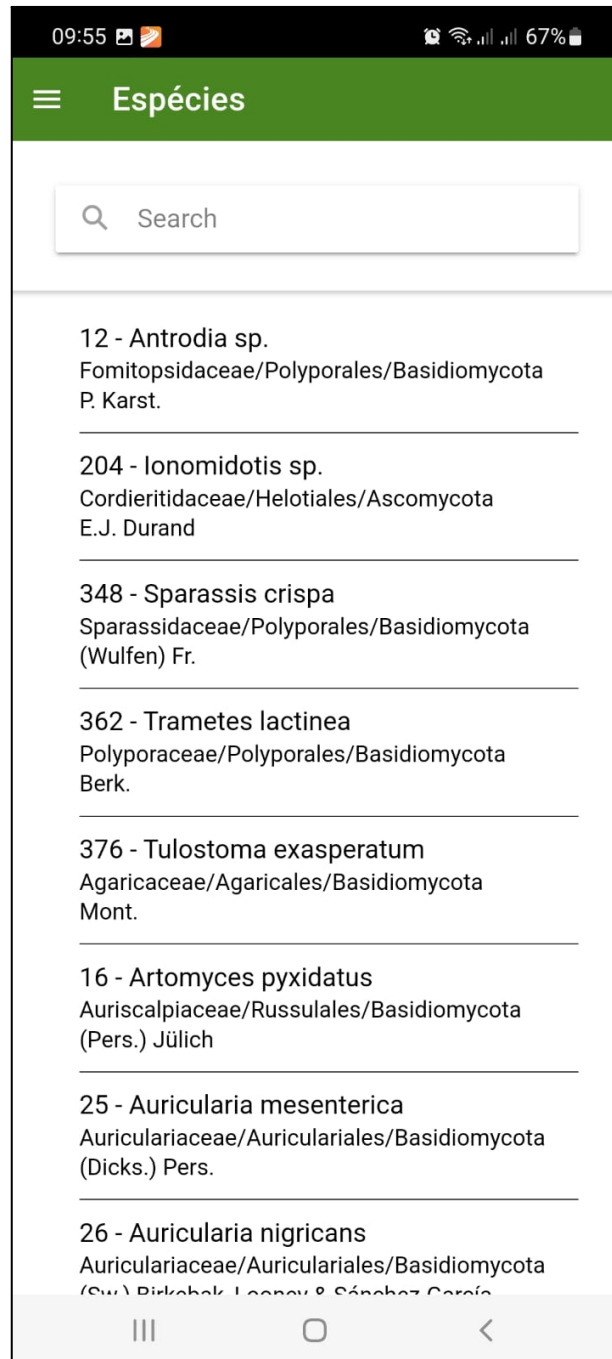
Figura 36 - Tela que contém os envios de espécimes do usuário / RF11



Fonte: elaborado pelo autor

Todos os usuários logados no aplicativo podem clicar no menu Espécies e verificar as espécies cadastradas no aplicativo, é possível pesquisar as espécies por nome, id, família, entre outros. A figura 37 mostra a tela de espécies.

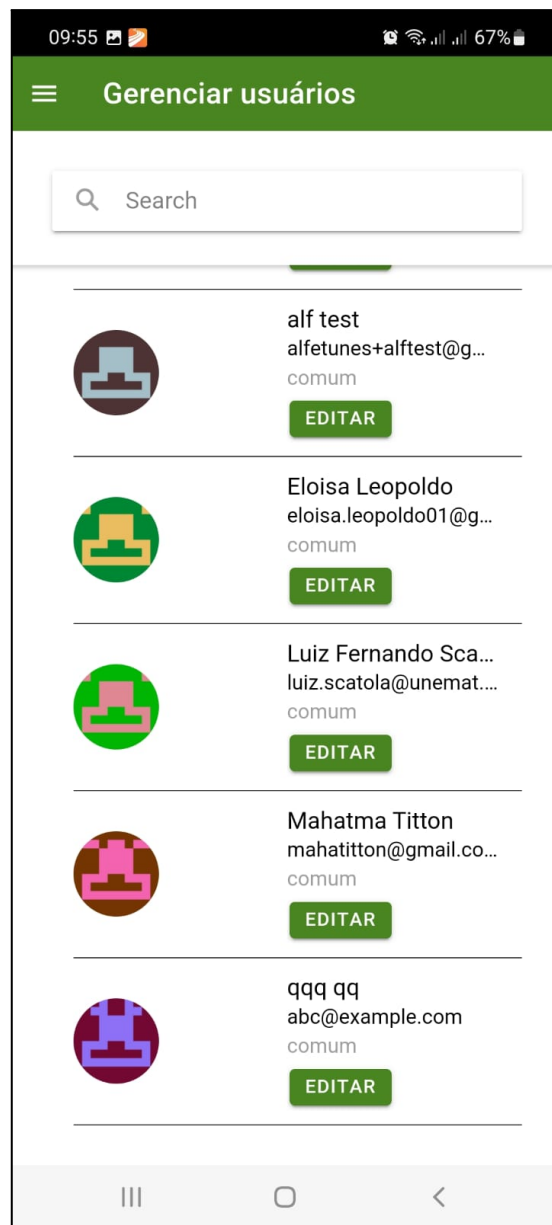
Figura 37 - Tela de espécies cadastradas no aplicativo / RF04



Fonte: elaborado pelo autor

O usuário do tipo administrador tem acesso a um menu em que ele pode efetuar o gerenciamento de usuários cadastrados no aplicativo, é possível editar usuários adicionando um novo grupo de permissão ao usuário. A tela de gerenciamento de usuários é apresentada na figura 38.

Figura 38 - Gerenciamento de usuários / RF08



Fonte: elaborado pelo autor

Usuários do tipo curador, podem verificar os envios de outros usuários, na tela de curadoria é possível verificar as informações submetidas, a data do envio, a

foto do espécime, a localização da captura, entre outros. Na figura 39 está a tela de curadoria do sistema.

Figura 39 - Tela de curadoria do sistema / RF07



Fonte: elaborado pelo autor

Todos os usuários podem efetuar o envio de fotos através do aplicativo, é possível efetuar o corte destas imagens antes de enviar para a análise da rede neural conforme pode ser vista na figura 40.

Figura 40 - Tela de envio e edição de imagens / RF02



Fonte: elaborado pelo autor

Após o corte da imagem, o usuário digita algumas informações sobre aquela captura, a data e a localização são preenchidas automaticamente, o usuário digita algumas observações que ele achar relevante, o substrato onde o fungo foi encontrado e o ângulo da foto, conforme visto na figura 41.



Figura 41 - Tela de adição de informações sobre o fungo / RF02



12:40 57%

☰ Que fungo é esse?



VOLTAR ANALISAR

Selecione o ângulo da foto:

Frente ▾

Por favor, preencha as informações complementares:

Substrato  
Solo

Observações  
amanita muscaria

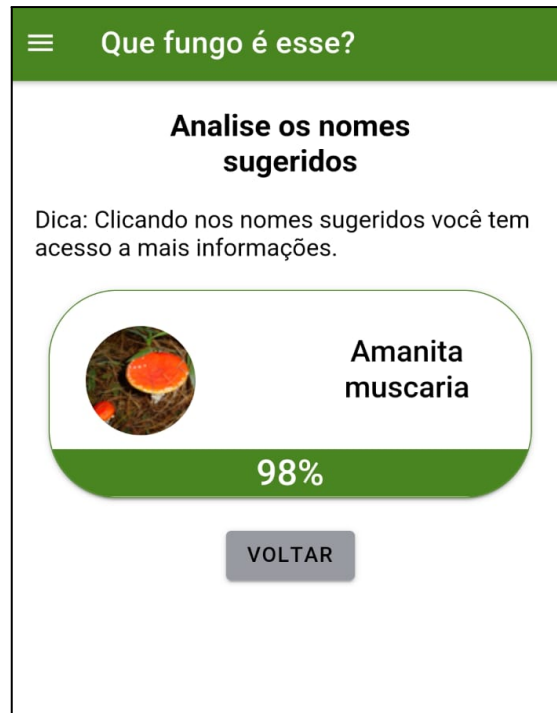
Data  
01/12/2022

Localização  
Brasil-Santa Catarina-Florianópolis-88054-6

Fonte: elaborado pelo autor

Após isso o aplicativo envia a foto para a API recebendo um JSON com as sugestões de nome para aquele espécime e a probabilidade de pertencer a uma determinada espécie. Essas informações são exibidas na tela apresentada na figura 42.

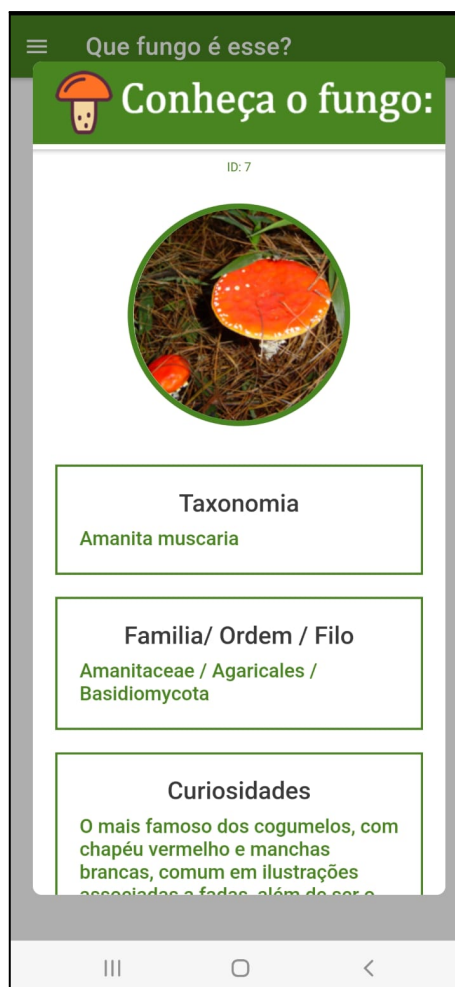
Figura 42 - Tela de sugestão de nomes de espécies / RF02



Fonte: Elaborado pelo autor

É possível perceber o que acontece quando o usuário do aplicativo clica na espécie sugerida, é mostrada mais informações sobre a espécie, conforme pode ser visto na figura 43.

Figura 43 - Tela com mais informações sobre a espécie / RF02



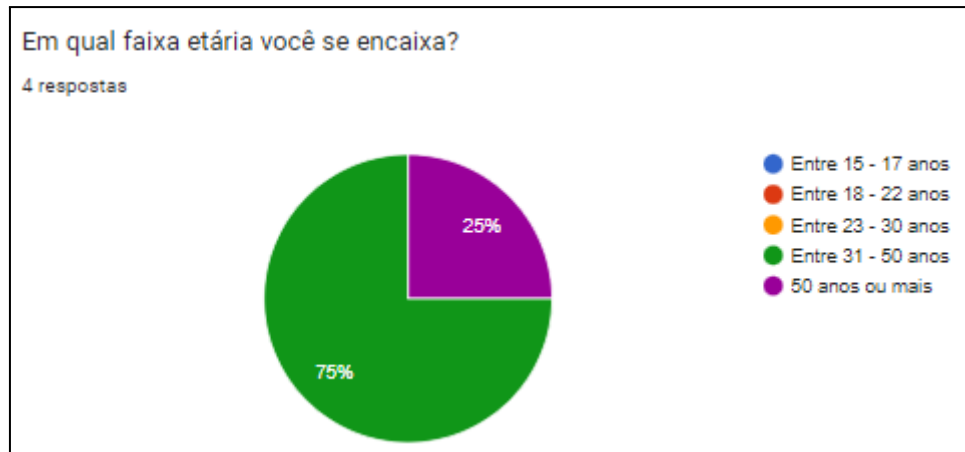
Fonte: Elaborado pelo autor

## 4.2 PESQUISA DE AVALIAÇÃO DO APLICATIVO

Após a finalização do desenvolvimento foi efetuada uma pesquisa para avaliação do aplicativo com os usuários do sistema, estes convidados pelo grupo MIND.Funga, sendo alunos e cientistas da área de micologia. Abaixo são apresentados alguns dos resultados obtidos com essa pesquisa.

A maioria dos usuários do aplicativo têm entre 31 - 50 anos.

Figura 44 - Faixa etária de usuários do sistema



Fonte: Elaborado pelo autor

Para ajudar os usuários a instalar o aplicativo foi gerado um vídeo mostrando o processo de instalação do mesmo. Avaliando as respostas dos usuários todos sentiram uma facilidade nesse processo, o que demonstra o sucesso na estratégia de distribuição e suporte para instalação do aplicativo.

Figura 45 - Avaliação da facilidade de instalação do aplicativo



Fonte: Elaborado pelo autor

Todos os usuários acharam fácil o processo de cadastro de usuário no aplicativo, ou seja, a aplicação se mostrou eficiente nesse item.

Figura 46 - Avaliação da facilidade de cadastro no aplicativo



Fonte: Elaborado pelo autor

Nenhum usuário passou por problemas técnicos durante o uso do aplicativo, o que demonstra uma estabilidade na aplicação de um modo geral.

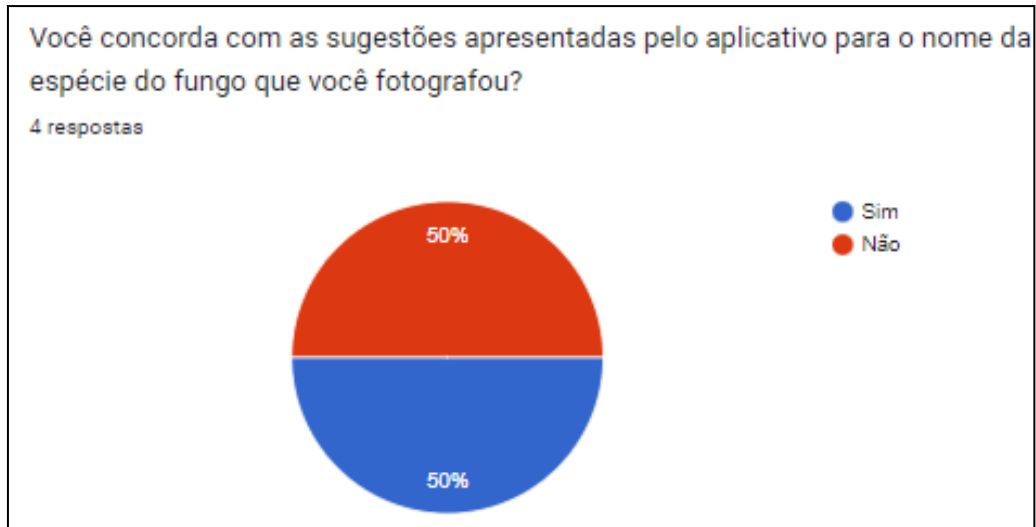
Figura 47 - Avaliação da estabilidade do aplicativo



Fonte: Elaborado pelo autor

Alguns usuários reportaram problemas nas sugestões de nome da espécie sugerida para os fungos, o que demonstra que a rede neural utilizada precisa ser melhorada para que as sugestões sejam mais eficientes.

Figura 48 - Avaliação das sugestões de espécie do aplicativo



Fonte: Elaborado pelo autor

As respostas demonstradas na figura 49 vem a confirmar que alguns usuários não ficaram satisfeitos com as sugestões de nomes de fungos que foram dadas pela rede neural.

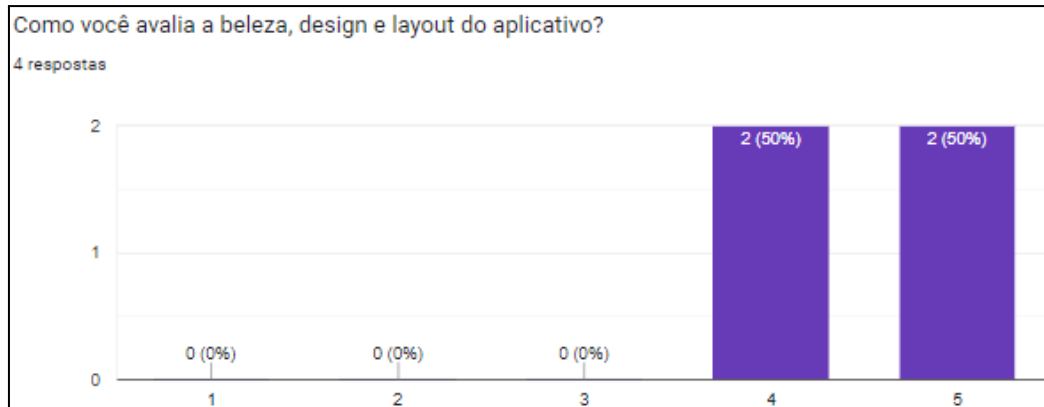
Figura 49 - Avaliação das sugestões dadas pelo aplicativo



Fonte: Elaborado pelo autor

Avaliando a beleza e design do aplicativo alguns usuários não deram a nota máxima para o aplicativo, porém as respostas foram satisfatórias tendo em vista que é um aplicativo que tende a ter melhorias futuras nesse aspecto.

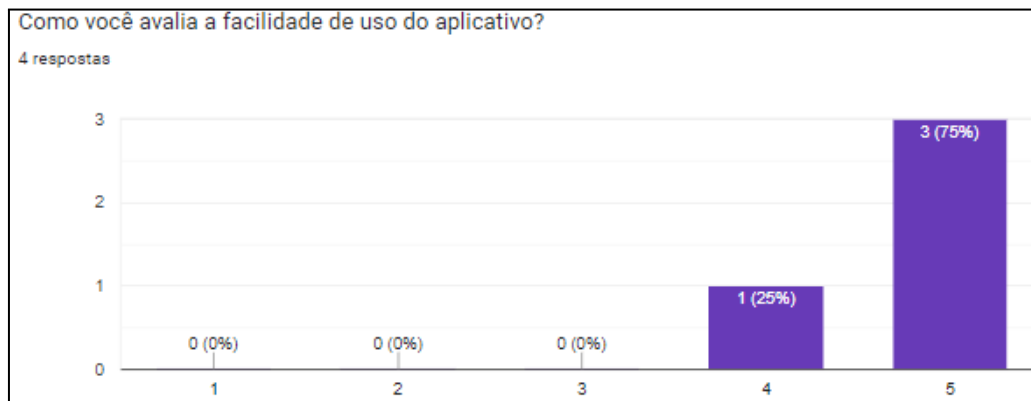
Figura 50 - Avaliação do design do aplicativo



Fonte: Elaborado pelo autor

A maioria dos usuários avaliou o aplicativo como sendo fácil de utilizar.

Figura 51 - Avaliação da facilidade de uso do aplicativo



Fonte: Elaborado pelo autor

Com base nas respostas dos usuários avaliados é possível concluir que no mérito do desenvolvimento desse trabalho, não levando em consideração a rede neural pré-existente o aplicativo apresentou bons resultados, se apresentando como fácil de instalar, utilizar, com um design agradável e sem problemas graves de performance.

## 5 CONCLUSÃO

Este trabalho buscou apresentar o processo efetuado para a elaboração do aplicativo de identificação de macrofungos desenvolvido para o grupo de pesquisa MIND.Funga. Buscou-se efetuar uma prova de conceito que é necessária para o grupo verificar que essa ferramenta pode se apresentar como uma grande auxiliadora na colheita de mais informações sobre fungos bem como auxiliar no processo de ensino e popularização do estudo da micologia.

Foi feita uma pesquisa de aplicativos semelhantes para verificar suas principais características e efetuar comparações com essa primeira versão desenvolvida, sendo possível perceber a diferença principalmente na quantidade de publicidade que os mesmos apresentam, no layout pensado por essas soluções e na quantidade de espécies e informações cadastradas.

O uso do Framework Ionic se mostrou acertado para a elaboração do aplicativo e foi possível gerar um arquivo .apk que foi instalado em dispositivos Android tendo sucesso na comunicação com a API utilizada para o backend da aplicação.

O aplicativo desenvolvido foi utilizado por usuários do grupo MIND.Funga e convidados tendo sucesso na captura e edição de imagens de fungos, houveram algumas reclamações referentes a exatidão das sugestões das espécies o que exige um melhor treinamento da rede neural, porém, o aplicativo se apresentou para os usuários como sendo fácil de utilizar, instalar e possuindo uma boa interface, servindo como um protótipo que poderá ser utilizado pelo grupo em pesquisas futuras

É esperado que o aplicativo seja utilizado pelo grupo e que auxilie na busca de novas espécies de fungos e que seja atualizado com novos desenvolvimentos se tornando uma grande ferramenta que será responsável por expandir e gerenciar todo o banco de dados de espécies de fungos que se faz presente no território estudado pelo grupo.

### 5.1 SUGESTÕES PARA TRABALHOS FUTUROS



Com base nos primeiros feedbacks foram avaliadas alguns pontos que podem ser melhorados em futuras versões do mesmo e também alguns casos de usos solicitados que não foram desenvolvidos nessa primeira versão por priorização de outros requisitos. Esses elementos podem ser desenvolvidos em trabalhos futuros. Portanto algumas sugestões de trabalhos são:

- Criação de uma rede neural que possa ser utilizada de maneira offline, seria uma rede que iria rodar no próprio aplicativo não sendo necessária a comunicação com a API para a identificação dos fungos;
- Melhorias na interface do aplicativo buscando um design mais moderno e mudança de cor;
- Desenvolver edição e adição de nova espécies conforme RF05 e RF06;
- Apresentar o número de envios de cada usuário, buscando saber quantas contribuições cada usuário deu para a aplicação;
- Criar a tela de Mapa de Registros com os registros dos espécimes enviadas sendo apresentadas em um mapa conforme solicitado no RF03;
- Possibilitar mais filtros e campos de pesquisas nas telas desenvolvidas;
- Colocar o Protocolo de captura de imagens de macrofungos (<https://repositorio.ufsc.br/handle/123456789/230592>) no RF10
- Melhorar a Rede Neural e a API para se adequarem a novos requisitos dos membros do MIND.Funga;
- Publicar o aplicativo existente na Google Play Store.

## REFERÊNCIAS

SILVA, Ricardo Pereira e. **Engenharia de Requisitos Produção de Especificação de Requisitos**. Florianópolis, 2019. 57 slides. Disponível em: <[https://www.inf.ufsc.br/~ricardo.silva/INE5417e5608/apostila/capitulo00\\_analise\\_de\\_requisitos\\_2016.pdf](https://www.inf.ufsc.br/~ricardo.silva/INE5417e5608/apostila/capitulo00_analise_de_requisitos_2016.pdf)>. Acesso em: 19 de fevereiro de 2021.

KITCHENHAM, Barbara. **Procedures for Performing Systematic Reviews**. [S.l.], 2004.

OLIVEIRA, JC. **Tópicos em Micologia Médica**. 4º ed. Rio de Janeiro: ControlLab, 2014. Disponível em: [https://controllab.com/pdf/topicos\\_micologia\\_4ed.pdf](https://controllab.com/pdf/topicos_micologia_4ed.pdf). Acesso em: 13 out.2022.

MARQUES, Marta Bento Silva. **Diversidade e Ecologia dos Macrofungos do Jardim Botânico da Universidade de Coimbra**. 2012. Disponível em: <https://core.ac.uk/download/pdf/143400399.pdf>. Acesso em: 13 out.2022.

DICIONÁRIO AMBIENTAL. **O que é uma Espécie e um Espécime**. 2014. Disponível em: <https://oeco.org.br/dicionario-ambiental/28502-o-que-e-uma-especie-e-um-especime>. Acesso em: 13 out.2022.

BITTENCOURT, Felipe; KARSTEDT, Fernanda; PULGARIN, Melissa; et al. **Protocolo de captura de imagens de macrofungos**. 2022. Disponível em: <https://repositorio.ufsc.br/handle/123456789/230592>. Acesso em: 15 out.2022.

DRECHSLER, Elisandro. **An Infrastructure for Cataloguing and Classifying Macrofungi using Convolutional Neural Networks**. 2022. Em fase de elaboração.

FARIAS, Roger de Córdova. **Desenvolvimento de API para classificação de macrofungos**. 2022. Em fase de elaboração.

IONIC. **Ionic Framework**. 2022. Disponível em: <https://ionicframework.com/>. Acesso em: 17 out.2022.

SOUSA, Vinicius Rodrigues. **Desenvolvimento e monitoramento de uma API de E-Commerce**. 2021. Disponível em: <https://www.sorocaba.unesp.br/Home/Graduacao/EngenhariadeControleeAutomacao/galdenoro1906/tcc---copia.pdf>. Acesso em: 18 out.2022.

SANCHEZ, ANA. **Ionic is composed**. 2018. Disponível em: [https://medium.com/@asanchez\\_77574/ionic-is-composed-the-mvc-model-view-controller-or-model-view-controller-e47566a1dd2e](https://medium.com/@asanchez_77574/ionic-is-composed-the-mvc-model-view-controller-or-model-view-controller-e47566a1dd2e). Acesso em: 20 out.2022.

WEBSTER, J. WEBER, R. **Introduction to fungi**. Cambridge: New York. [S.l.], 2007

MUELLER, G.M; SCHMIT, J.P; LEACOCK, P.R; et al. **Global diversity and distribution of macrofungi.** 2007. Disponível em: <https://doi.org/10.1007/s10531-006-9108-8>. Acesso em: 17 dez.2022.

# Apêndices

## APÊNDICE A – ARTIGO APLICATIVO PARA CLASSIFICAÇÃO DE MACROFUNGOS UTILIZANDO REDES NEURAIAS

### Aplicativo para classificação de macrofungos utilizando redes neurais

Alfeu G. dos Santos

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

alfeu.goncalves@grad.ufsc.br

**Abstract.** *This paper describes an application developed in the context of a research project on macrofungi called MIND.Funga conducted at the Federal University of Santa Catarina (UFSC). The goal of the app is to help professionals and volunteers to quickly identify species of macrofungi and record photos to expand existing databases. In addition, the app can also be used as a teaching tool for teachers, helping them in their classes on macrofungi. The application has been developed with a neural network using an image base from the research group and successfully tested by scientists and people invited by the group.*

**Resumo.** *Este artigo descreve um aplicativo desenvolvido no contexto de um projeto de pesquisa sobre macrofungos chamado MIND.Funga realizado na Universidade Federal de Santa Catarina (UFSC). O objetivo do aplicativo é ajudar profissionais e voluntários a identificar rapidamente espécies de macrofungos e registrar fotos para ampliar as bases de dados existentes. Além disso, o aplicativo também pode ser usado como ferramenta de ensino para professores, auxiliando-os em suas aulas sobre macrofungos. O aplicativo foi desenvolvido com uso de uma rede neural usando uma base de imagens do grupo de pesquisa e testado com sucesso por cientistas e pessoas convidadas pelo grupo.*

#### 1. Introdução

Tendo em vista a grande diversidade de macrofungos nas florestas brasileiras existe por parte da comunidade científica um esforço voltado ao estudo, à catalogação e ao registro fotográfico desses organismos. Em conjunto com o grupo de pesquisa MIND.Funga (Monitoramento e Inventário da Diversidade Neotropical de Fungos) e ao LAPIX (Laboratório de Processamento de Imagem e Computação Gráfica) da Universidade Federal de Santa Catarina (UFSC) foi desenvolvido um aplicativo para ajudar cientistas e outros profissionais a identificar e registrar espécies de macrofungos nas florestas do Brasil. O aplicativo foi criado para o sistema Android e utiliza uma rede neural treinada com base em imagens e informações fornecidas pelos grupos e desenvolvida por Thiago Zimmermann Loureiro Chaves, além de uma API para comunicação com o backend criada por Farias(2022). Os usuários do aplicativo podem cadastrar-se, editar imagens e enviá-las para a rede neural para identificação de espécies. O objetivo da ferramenta é auxiliar no processo de reconhecimento e armazenamento de informações coletadas pelo grupo de pesquisa.

O desenvolvimento de um aplicativo para identificação de fungos feito para o grupo visa fornecer aos usuários uma experiência de uso pensada para as necessidades

desses cientistas, buscando a captura e edição de imagens e a utilização dos algoritmos de reconhecimento visual criados com a curadoria dessa equipe com amplo conhecimento na área. Com a tecnologia sendo testada pelos pesquisadores, pode-se obter uma evolução do aplicativo buscando superar a precisão dos aplicativos já existentes. A ferramenta pode ser utilizada no ensino da micologia em instituições de ensino médio e para o uso de cientistas e pessoas interessadas nessa área de conhecimento, trazendo uma variada gama de informações e imagens das espécies disponíveis no banco de dados, sendo importante para coleta e armazenamento de novas imagens no banco de dados do grupo, possibilitando novas descobertas e futuras melhorias no treinamento da rede neural existente.

## **2. Fundamentação Teórica**

Para o entendimento deste artigo é necessário conhecer alguns elementos que servirão como base para a elucidação das tecnologias e conceitos utilizados ao longo do desenvolvimento do aplicativo. Esses elementos podem ser divididos em alguns tópicos, sendo eles:

- Os conceitos gerais sobre macrofungos e micologia;
- Tecnologias desenvolvidas pelo grupo MIND. Funga e LAPIX que foram utilizadas pelo aplicativo;
- O framework de desenvolvimento IONIC, que forneceu as ferramentas necessárias para a criação do aplicativo;

### **2.1. Micologia e Macrofungos**

De acordo com Oliveira (2014) a micologia é o estudo dos fungos, organismos microscópicos que têm uma ampla variedade de aplicações na sociedade, incluindo a produção de produtos químicos e farmacêuticos e alimentos. No entanto, alguns fungos também podem causar envenenamentos quando são confundidos com espécies comestíveis.

Os fungos também têm um papel importante na natureza, ajudando a ciclar nutrientes e estabelecer conexões e comunicação entre diferentes componentes dos ecossistemas (Webster & Weber 2007). E podem ser divididos em microfungos e macrofungos de acordo com o tamanho de suas estruturas reprodutivas. Os microfungos têm estruturas reprodutivas não visíveis a olho nu e só podem ser diagnosticados por seus aspectos macro, enquanto os macrofungos produzem estruturas visíveis a olho nu e representam cerca de 15% das espécies de fungos conhecidas (Mueller et al. 2007).

Durante o desenvolvimento do aplicativo se deu o foco nos macrofungos, já que nessa categoria de fungos é possível a captura de fotos através de um smartphone para posterior análise no aplicativo.

No âmbito do aplicativo também é importante destacar os conceitos de Espécie e Espécime que são bastante utilizados, segundo definição do Dicionário Ambiental (2014) espécie é uma palavra que aponta sempre para uma coletividade, vindo do latim

species, “conjunto de traços que fazem reconhecer qualquer objeto”. Já os espécimes se referem a um indivíduo dentro dessa coletividade, do latim specimen, “amostra, prova” é “qualquer indivíduo de uma espécie”. No aplicativo os espécimes são entendidos como amostras, ou fotos coletadas, essas imagens sempre fazem parte de uma determinada espécie.

## **2.2. As tecnologias do grupo MIND. Funga e Lapix**

Para o desenvolvimento do aplicativo foram utilizadas algumas tecnologias que já haviam sido previamente criadas por membros do grupo MIND.Funga em conjunto com o LAPIX. Já existia uma rede neural treinada para isso sendo desenvolvida por Thiago Zimmermann Loureiro Chaves e uma API com os endpoints utilizados pelo aplicativo desenvolvida por Farias (2022)

Segundo (Drechsler-Santos, 2022) o conjunto de dados do MIND.Funga que foi utilizado para o treinamento da rede neural é composto por 14.707 imagens selecionadas representando 423 espécies de fungos, as imagens brutas foram recebidas por e-mail ou incorporadas no conjunto de dados, as fontes incluíam micologistas parceiros, voluntários, cientistas cidadãos, grupos de micologia online e projetos de pesquisa anteriores focados na diversidade de macrofungos. Todas as imagens foram submetidas a um processo de controle de qualidade e tratamento, que envolveu as seguintes etapas:

- Remoção de imagens de baixa qualidade, sem foco e imagens com objetos estranhos;
- Reenquadramento do espécime para o centro da imagem e padronização da proporção;
- Remoção do fundo original substituindo por fundo verde e branco (códigos HEX) para melhorar o treinamento da CNN e o reconhecimento do espécime. O grupo verificou que a substituição do plano de fundo não melhorou significativamente o desempenho da CNN, sendo assim, novas adições não foram modificadas.

Já os endpoints da API do inglês Application Programming Interface foram utilizados para permitir que o aplicativo fizesse uso dos recursos do backend, do banco de dados e da rede neural criada. O aplicativo precisa submeter requisições HTTP para a API enviando ou recebendo dados para o seu funcionamento. A API que foi utilizada pelo aplicativo está dividida em 5 grandes grupos, esses grupos possuem os endpoints que são utilizados pelo aplicativo, sendo eles mostrados abaixo:

- Curatorship (Curadoria): Há endpoints que mostram todas as submissões de espécimes feitas por todos os usuários do sistema, permitindo que o usuário curador visualize as fotos enviadas. Também é possível definir manualmente uma espécie para um espécime submetido;
- Species (Espécies): Contém todos os endpoints necessários para edição, exclusão, e busca das espécies cadastradas no banco de dados da aplicação;
- Specimen (Espécimes): É possível trazer as imagens dos espécimes através dos endpoints desse grupo, passando um id de Espécime o sistema retorna uma imagem com o formato JPEG;

- Specimen Submissions (Envios de Espécimes): Através dos endpoints desse grupo é possível efetuar o envio dos Espécimes para a análise da rede neural, nesse grupo está o endpoint onde se pode enviar uma imagem para a API que retorna um JSON contendo as espécies sugeridas e suas probabilidades. Também é possível trazer os envios de espécimes do usuário atual;
- Users (Usuários): Aqui está disponível toda a parte de acesso dos usuários, é possível realizar o cadastro de novos usuários, efetuar operações de login, logout e recuperação de senha.

### 2.3. Ionic Framework

O IONIC é um framework de desenvolvimento de aplicativos móveis gratuito e amplamente utilizado que foi criado em 2013. Ele é aceito pela comunidade de desenvolvedores e possui uma ampla documentação e vídeos explicativos para facilitar a aprendizagem. O IONIC permite o desenvolvimento de aplicativos híbridos usando tecnologias web como HTML5, SCSS e AngularJS. O aplicativo desenvolvido com o IONIC pode ser facilmente portado para outros sistemas operacionais. O front-end do aplicativo é desenvolvido com HTML e SCSS e a lógica é desenvolvida com AngularJS (IONIC,2022).

O padrão de desenvolvimento MVC (Model-View-Controller) é utilizado, dividindo o desenvolvimento de software em três partes: modelo, visão e controle. O modelo é a camada do software responsável por carregar os dados, enviando ou recebendo informações das bases de dados. A visão é a parte do código que apresenta os dados fornecidos pelo modelo, no caso do IONIC, através do código HTML e componentes como plugins, botões e modais. O controle é a camada que liga a visão e o modelo, permitindo o envio de comandos para o modelo e atualizando seu status para que a exibição possa ser alterada (SANCHEZ,2018). A figura 1 mostra a organização do código de uma das classes do aplicativo.

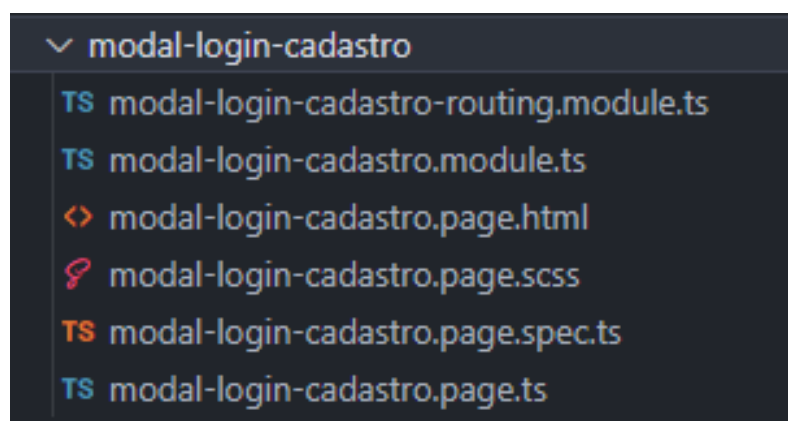


Figura 1. Organização básica do código de uma classe em um projeto IONIC



## 2.4. Análise de outros aplicativos

Com o objetivo de avaliar a qualidade, facilidade de uso e eficácia na categorização de espécies, analisando as suas funcionalidades e a interface gráfica de outros aplicativos semelhantes para a finalidade de classificação de fungos, foi realizada uma análise de outros aplicativos concorrentes com base nos protocolos elaborados por Kitchenham (2004). É mostrado abaixo o objetivo da revisão, escopo da pesquisa, estratégias de busca, critérios de inclusão e exclusão para seleção desses aplicativos e a técnica de extração de dados e sumarização de resultados.

- Questão: Quais os principais aplicativos para a identificação e classificação de fungos disponíveis para uso?
- População: Aplicativos para a identificação e classificação de fungos ou aplicativos para classificação de vegetais (onde às vezes aparecem fungos), disponíveis em bibliotecas digitais.
- Intervenção: Análise de aplicativos para identificação de fungos.
- Resultado: Comparação de aplicativos para identificação de fungos.
- Contexto: Biblioteca digital: Google Play.

### 2.4.1 Definições da Pesquisa

Os detalhes da definição de pesquisa, como: locais de pesquisa, termos, critérios de inclusão e exclusão, são apresentados no quadro 1.

Bibliotecas pesquisadas	Termos pesquisados
- Google Play;	- "Mushroom Identification Apps"; - "Aplicativos para identificar fungos".
Critérios de inclusão	
- Aplicativos com mais de 100 mil downloads; - Aplicativos com mais de uma versão; - Aplicativos cuja última atualização tenha sido feita entre Março de 2020 até Março de 2022; - Aplicativos para Android; - Aplicativos que consigam efetuar classificação de espécies de fungos através de fotos.	
Critérios de exclusão	

- Aplicativos com menos de 100 mil downloads;
- Aplicativos que mostram informações sobre fungos mas que não possuem nenhuma inteligência artificial para identificação dos mesmos;
- Aplicativos com menos de 3 estrelas de avaliação;
- Aplicativos que não tiram fotos de fungos;

**Quadro 1. Definições da pesquisa**

### 2.4.2 Execução da Pesquisa

Inicialmente a seleção resultou em 60 aplicativos e uma análise por avaliação, tipo do aplicativo e data de atualização resultou em 5 aplicativos selecionados. A execução da busca é mostrada no quadro abaixo.

Google Play			
Termos Pesquisados	- "Mushroom Identification Apps" OU "Aplicativos para identificar fungos".		
Redefinição da Pesquisa	Downloads - Maior que 100 mil downloads; Avaliação - Aplicativos com mais de 3 estrelas de avaliação; Atualização - Março de 2020 até Março de 2022.		
Quantidade de Resultados	60	Selecionados	5

**Quadro 2. Execução da pesquisa**

### 2.4.3 Comparativo e conclusões sobre e os aplicativos selecionados

Após a análise dos principais aplicativos para identificação de fungos, podemos concluir que eles oferecem uma ampla variedade de recursos e características úteis para os usuários. Todos eles permitem que você tire uma foto do fungo e o compare com imagens em sua base de dados para tentar identificá-lo, possuindo uma ampla variedade de informações sobre o hábitat e comestibilidade dos fungos, bem como glossários de termos relacionados a fungos.

Embora cada aplicativo tenha suas próprias vantagens e desvantagens, todos eles podem ser úteis para a identificação de fungos. No entanto, é importante lembrar que esses aplicativos são apenas ferramentas de auxílio e não devem ser usados como substitutos de um especialista em fungos qualificado, todos possuem sugestões de identificação que podem conter erros.

Apesar da grande quantidade desses aplicativos existem ainda poucas opções gratuitas, sendo a maioria dos aplicativos analisados pagos ou com várias propagandas que ajudam as empresas a viabilizarem financeiramente esses projetos. Durante o uso

desses aplicativos esse foi um fator que se apresentou de maneira negativa pois muitas vezes as funcionalidades ficam escondidas ou mais difíceis de acessar devido a grande quantidade de propagandas que se apresentam.

É possível concluir também que existem aplicativos que apresentam uma maior quantidade de espécies em seu banco de imagens quando comparadas às 423 espécies levantadas pelo grupo MIND.Funga, como é o caso do Fungus e do Mushroom Identifier.

Na tabela 1 é possível verificar uma comparação entre os aplicativos analisados:

Nome	Versão	Última Atualização	Nº de Downloads	Quantidade de Espécies	É gratuito?
PlantSnap	5.01.02	28/02/ 2022	10,000,000+	-	Não
Fungus	1.0.4	23/11/2020	100,000+	1736	Sim
Picture Mushroom	2.8.10	11/03/2022	1,000,000+	-	Não
Mushroom Identifier	2.66	08/11/2021	1,000,000+	900	Não
ShroomID	2.2.3	12/03/2022	100,000+	-	Não

**Tabela 1. Comparativo entre os aplicativos selecionados.**

### 3. Desenvolvimento do Aplicativo

#### 3.1. Especificações de requisitos do software

A especificação dos requisitos é um passo importante no processo de desenvolvimento de uma aplicação. Para isso, foram realizadas reuniões bissemanais com o grupo de pesquisa MIND.Funga para verificar as necessidades do grupo e identificar as atividades que deveriam ser desenvolvidas no aplicativo. Alguns dos requisitos levantados incluem:

Requisitos funcionais:

- RF01 – Efetuar Login: O aplicativo deve apresentar a opção de menu “Login” para a identificação de usuários no sistema;
- RF02 – Identificar Fungo: O aplicativo deve apresentar a opção de menu “Que fungo é esse?” para que os usuários possam tirar fotos e identificar os fungos;
- RF03 – Mostrar mapa de registros: O aplicativo deve apresentar a opção de menu “Mapa de Registros” para que os usuários possam ver os fungos identificados em mapa;

- RF04 – Ver Espécies: O aplicativo deve apresentar a opção de menu “Espécies” para que os usuários possam ver informações sobre as espécies cadastradas no sistema, ele poderá ordenar as espécies de maneira alfabética e por categoria .
- RF05 – Editar Espécie: O usuário poderá editar uma espécie já cadastrada, clicando sobre o nome da espécie, caso ele seja curador.
- RF06 – Cadastrar Espécie: Usuários do tipo curador podem cadastrar uma nova espécie, para tanto devem inserir as seguintes informações: Taxonomia, Autor, Família, Ordem, Filo, Etimologia, Curiosidade, Nota sobre a lista vermelha IUCN, Fotos do fungo e Observações gerais.
- RF07 – Proceder Curadoria: O aplicativo deve apresentar a opção de menu “Curadoria” para que os usuários do tipo Curador possam colaborar com correções e edições das informações cadastradas, ao salvar e confirmar as edições a espécie será inserida na lista de espécies públicas. O usuário poderá fazer a revisão e gestão de fotos das espécies cadastradas.
- RF08 – Gerenciar usuários: Usuários do tipo Administrador poderão fazer o gerenciamento de usuários através da opção de menu "Gerenciar Usuários" . Deverá ser possível listar todos os usuários do sistema e efetuar a edição das informações desses usuários.
- RF09 – Mostrar informações sobre o MIND.Funga: O aplicativo deve apresentar a opção de menu “Sobre o MIND.Funga” para que os usuários possam ver a versão do aplicativo e obter informações sobre a iniciativa MIND.Funga, além de informações sobre as redes sociais do projeto.
- RF10 – Mostrar dicas de fotografia: O aplicativo deve apresentar a opção de menu “Dicas de fotografia” para que os usuários possam ver as melhores maneiras de efetuar as fotografias dos fungos.
- RF11 – Ver Álbum: Mostra os espécimes enviados pelo usuário com informações sobre data, substrato, observações e localização do espécime enviado.

#### Requisitos não funcionais:

- RNF 01 – Identidade visual do App MIND.Funga: A interface gráfica deve seguir o padrão desenvolvido por Nathalie Ferreira;
- RNF 02 – Linguagem da programação: O programa será implementado em uma linguagem que permita a execução em dispositivos que possuam sistema operacional Android.

### 3.2. Casos de uso

Buscando apresentar de maneira gráfica as principais funcionalidades do aplicativo e as suas interações com os atores. Assim temos que os casos de uso são:

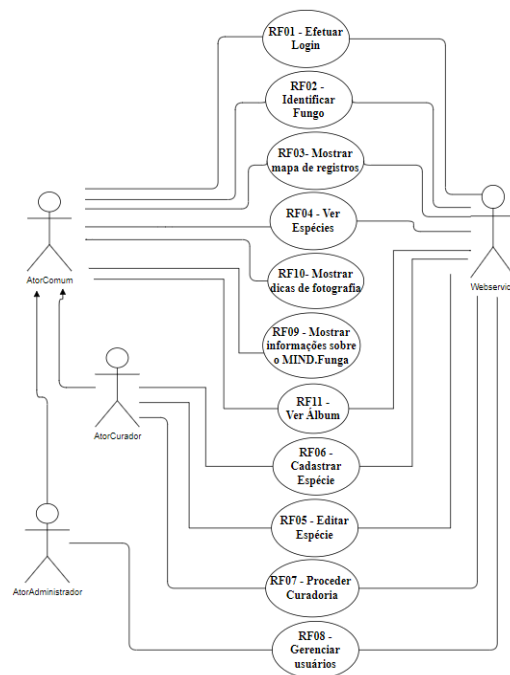


Figura 2. Diagrama de casos de uso do Aplicativo

### 3.3. Diagrama de visão geral de interação

Na Figura 3 é possível observar o Diagrama de visão geral de interação desenvolvido:

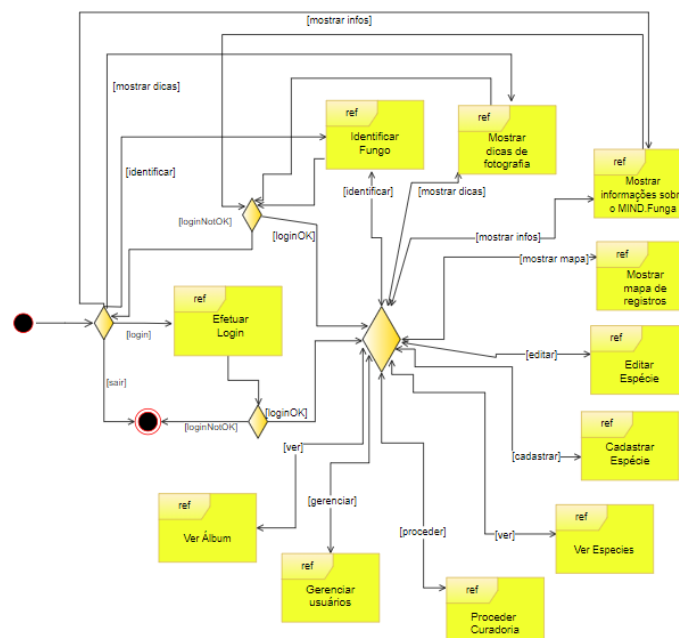
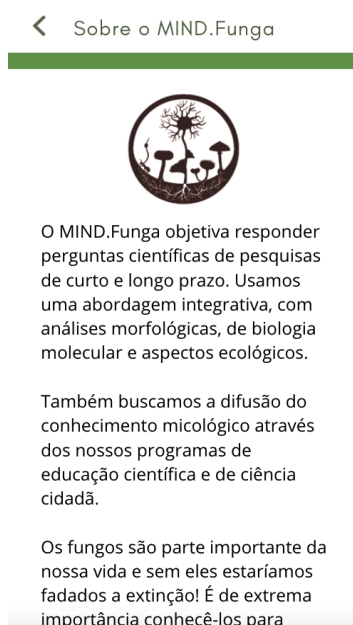


Figura 3. Diagrama de visão geral de interação

### 3.4. Protótipos de interface

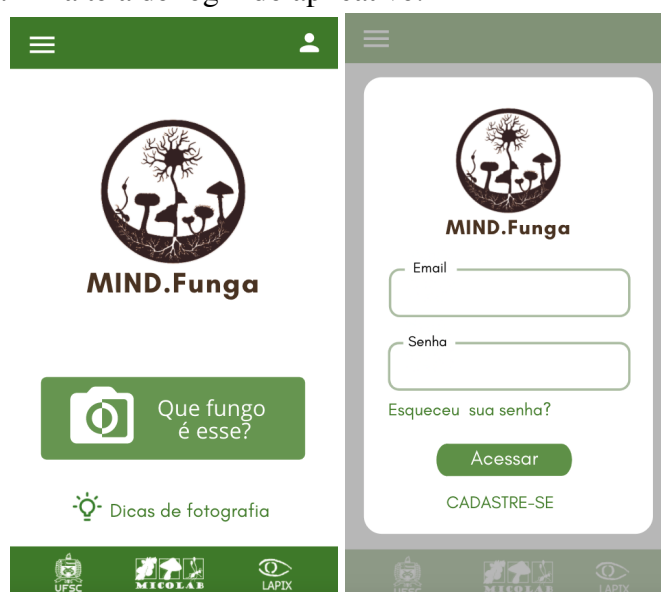
Para o desenvolvimento do aplicativo foi necessário a elaboração de alguns protótipos buscando desenvolver uma base visual para o aplicativo final, abaixo estão disponíveis algumas idéias de interfaces que foram elaboradas para esse trabalho.

Na figura 4 está o protótipo desenvolvido para o requisito funcional 09 a tela que mostra mais informações sobre o grupo MIND. Funga.



**Figura 4. Protótipo da tela contendo informações sobre o MIND. Funga / RF09**

Na figura 5 é possível observar um protótipo para a tela inicial do aplicativo e o requisito funcional 01 a tela de login do aplicativo.



**Figura 5. Tela inicial (esquerda) e Protótipo da tela de login no aplicativo / RF01**

### 3.5. Testes na API

Antes de iniciar o desenvolvimento do aplicativo, foi realizada uma etapa de testes na API existente para identificar e corrigir possíveis erros. Para isso, foram feitas requisições de teste em todos os endpoints que seriam usados pelo aplicativo com alguns softwares, entre eles o Postman. A figura 6 mostra um exemplo de um teste realizado no endpoint "signup", responsável pelo cadastro de novos usuários. Na requisição, os dados do novo usuário são passados no corpo da requisição e a API retorna a resposta 201, indicando que o usuário foi criado com sucesso. A API também retorna um json com o email, nome e sobrenome do usuário criado.

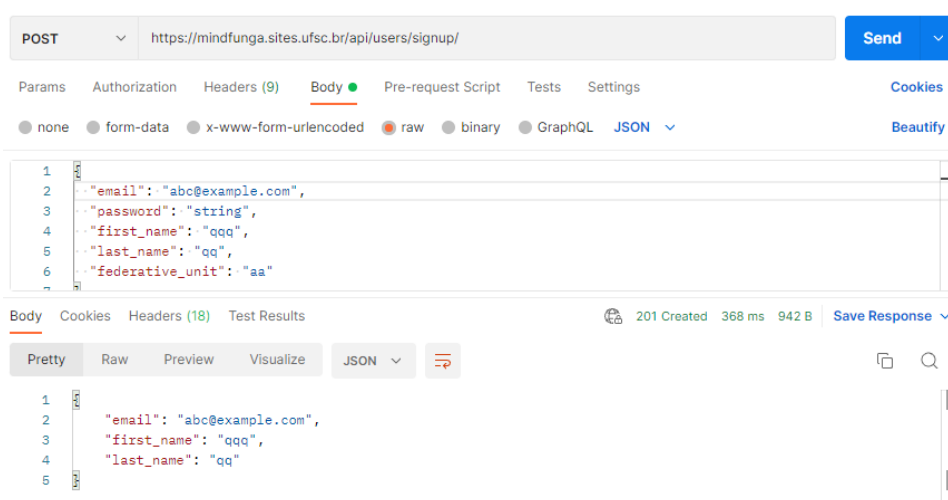


Figura 6. Testes de envios de pacotes para a API utilizando o POSTMAN

### 3.6. Programação do Aplicativo

Cada tela que aparece no aplicativo representa uma página dentro do projeto do aplicativo, para criar essas páginas se executa o comando “ionic g page paginas/nome\_da\_pagina” na raiz do projeto através do terminal do sistema operacional utilizado para a programação. Ao executar esse comando uma nova página é criada dentro da pasta src/app/paginas, foram criadas várias páginas para esse trabalho visando atender a maior quantidade de requisitos possíveis.

Cada uma dessas páginas possui uma estrutura padrão gerada pelo framework quando a mesma é inserida no projeto. Os arquivos mais importantes nessa estrutura e que contém a lógica programada para a exibição da interface do aplicativo são os seguintes:

- .scss: Nesse arquivo é onde são configuradas as propriedades visuais dos elementos que são vistos nas páginas, é possível efetuar configurações da cor da borda do elemento, cor de fundo, cor do texto, espaçamento do componente em relação a tela, entre outros.

- .html: Apresenta toda a descrição dos elementos que aparecem para o usuário, como botões, textos, e outras informações que aparecem na tela.
- .ts - É o arquivo que possui as funções e variáveis que são utilizadas pela página, aqui estão presentes as funções que fazem a comunicação com os endpoints da API, também é possível criar funções que mudam variáveis utilizadas no HTML, dessa maneira é possível apresentar resultados na tela, ou mesmo ler textos digitados pelo usuário.

## 4. Resultados Alcançados

### 4.1. Aplicativo Desenvolvido

Abaixo são apresentadas algumas das telas e fluxos desenvolvidos para esse trabalho. Para os testes do aplicativo foi gerado um arquivo .apk que não foi colocado em lojas de aplicativos, mas foi utilizado para a instalação em dispositivos Android por convidados do MIND.Funga.

Na tela inicial do aplicativo mostrada na figura 7, é possível ver o botão de fazer login sendo apresentado no centro da tela, nos protótipos essa tela apresentava o botão para analisar fungos no centro, porém analisando a API desenvolvida foi verificado que o usuário deveria fazer login para gerar um token antes de usar o endpoint que faz a identificação do fungo, portanto, foi colocado um botão para que o usuário do aplicativo possa fazer o login antes de enviar imagem para o sistema.



Figura 7. Tela inicial do aplicativo desenvolvido



Ao clicar no botão de login, o usuário é encaminhado para uma tela onde ele pode digitar o seu e-mail e a senha para ser validado pela API. A tela de login pode ser vista na Figura 8.

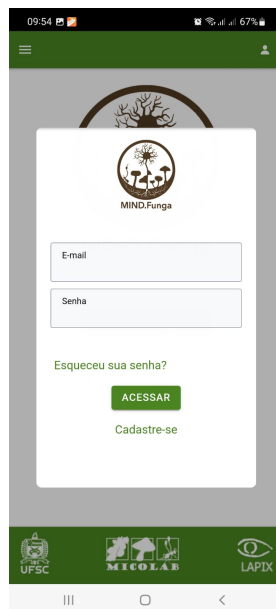


Figura 8. Tela de login do aplicativo / RF01

Dependendo do tipo de usuário logado, o sistema apresenta um menu lateral com opções diferentes, cada perfil de usuário pode efetuar ou não determinadas operações, o usuário curador ou especialista pode efetuar a curadoria dos espécimes submetidos pelos usuários, o usuário do tipo administrador pode efetuar o gerenciamento de usuário e para o usuário do tipo comum não é apresentada essas opções. Na figura 9 é apresentado os menus para os diferentes tipos de usuários.

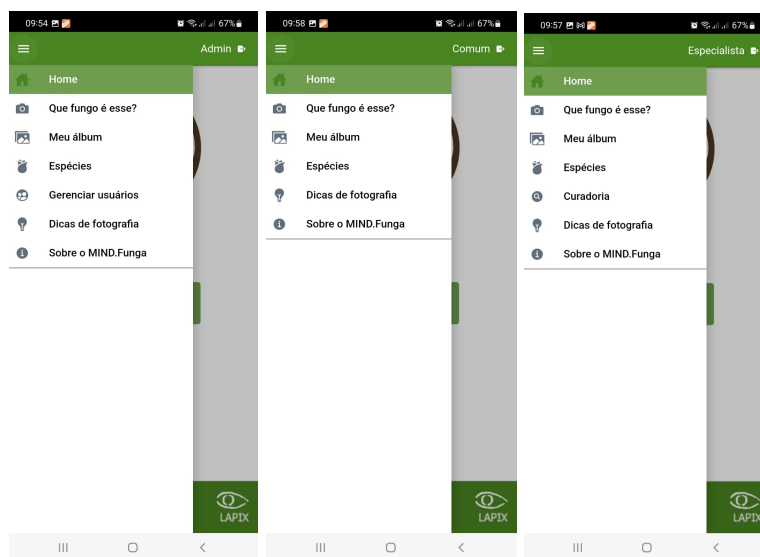


Figura 9. Menus laterais do aplicativo dado o tipo de usuário

Todos os usuários logados no aplicativo podem clicar no menu Meu álbum que apresenta todos os envios feitos por esse usuário, é possível efetuar a busca desses envios por data, tipo de substrato, localização, entre outros. A figura 10 apresenta essa tela.



**Figura 10. Tela que contém os envios de espécimes do usuário / RF11**

Todos os usuários podem efetuar o envio de fotos através do aplicativo, é possível efetuar o corte destas imagens antes de enviar para a análise da rede neural. Após o corte da imagem, o usuário digita algumas informações sobre aquela captura, a data e a localização são preenchidas automaticamente, o usuário digita algumas observações que ele achar relevante, o substrato onde o fungo foi encontrado e o ângulo da foto. O aplicativo envia a foto para a API recebendo um JSON com as sugestões de nome para aquele espécime e a probabilidade de pertencer a uma determinada espécie. Esse fluxo é mostrado na figura 11.

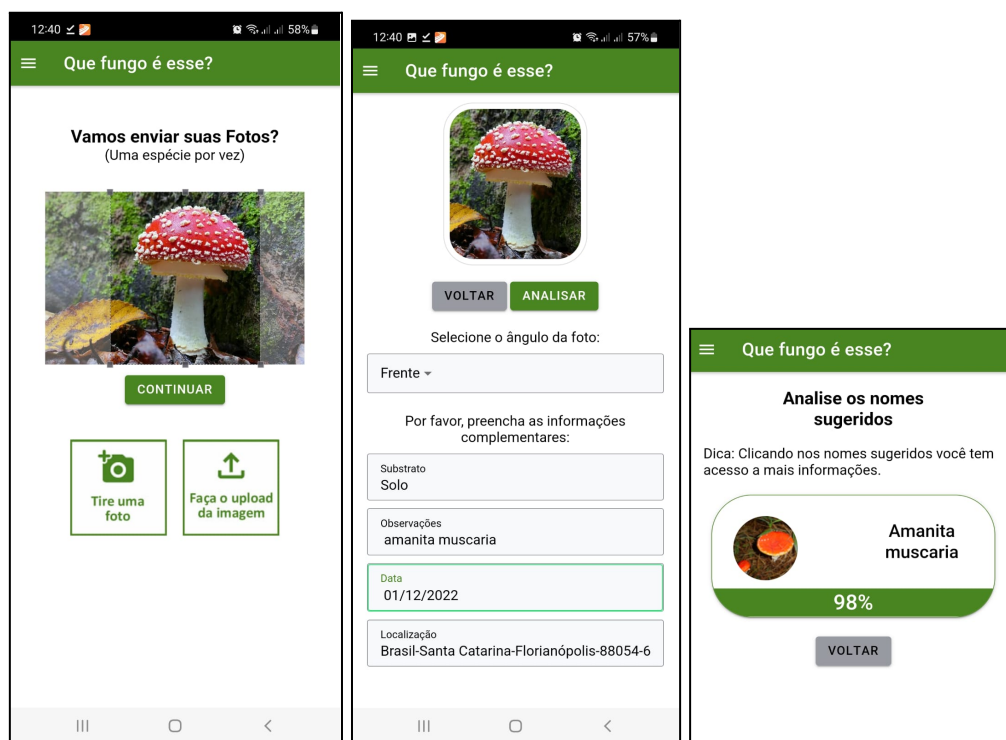


Figura 11. Fluxo de identificação do fungo / RF02

#### 4.2. Pesquisa de avaliação do Aplicativo

Após a finalização do desenvolvimento foi efetuada uma pesquisa para avaliação do aplicativo com os usuários do sistema, estes foram convidados pelo grupo MIND.Funga, sendo alunos e cientistas da área de micologia. Abaixo são apresentados alguns dos resultados obtidos com essa pesquisa.

Todos os usuários acharam fácil o processo de instalação do aplicativo. Na figura 12 é possível ver que todos acharam fácil a instalação do aplicativo

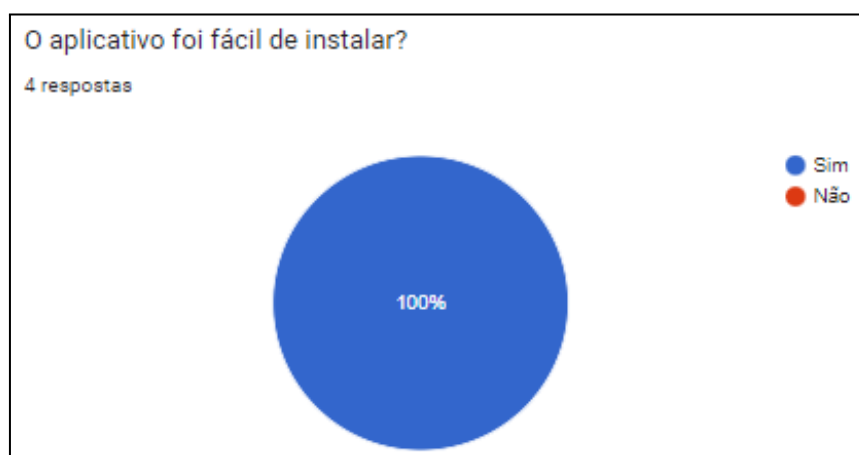


Figura 12. Avaliação da facilidade de instalação do aplicativo

Todos os usuários acharam fácil o processo de cadastro de usuário no aplicativo, ou seja, a aplicação se mostrou eficiente nesse item. Na figura 13 é possível ver que todos os usuários responderam que foi fácil se cadastrar no aplicativo.



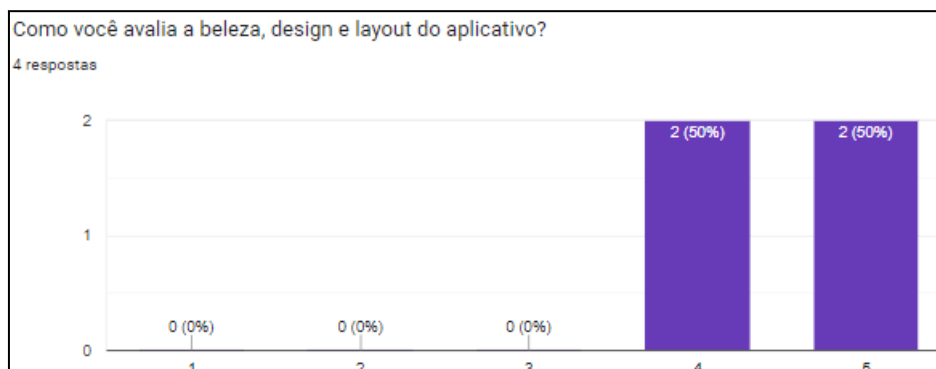
**Figura 13. Avaliação da facilidade de cadastro no aplicativo**

Nenhum usuário passou por problemas técnicos durante o uso do aplicativo, o que demonstra uma estabilidade na aplicação de um modo geral. Na figura 14 é possível ver os resultados dessa questão



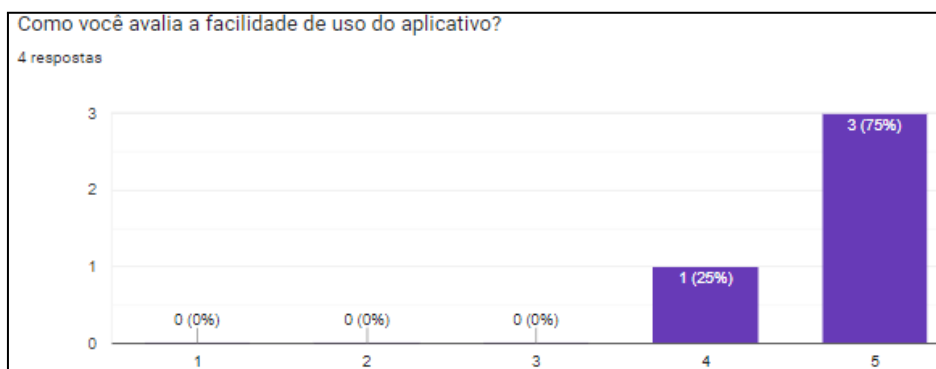
**Figura 14. Avaliação da estabilidade do aplicativo**

Avaliando a beleza e design do aplicativo alguns usuários não deram a nota máxima para o aplicativo, porém as respostas foram satisfatórias tendo em vista que é um aplicativo que tende a ter melhorias futuras nesse aspecto. Na figura 15 é possível ver as avaliações desse item.



**Figura 15. Avaliação do design do aplicativo**

A maioria dos usuários avaliou o aplicativo como sendo fácil de utilizar, conforme pode ser visto na figura 16.



**Figura 16. Avaliação da facilidade de uso do aplicativo**

Com base nas respostas dos usuários avaliados é possível concluir que foram alcançados bons resultados e o aplicativo se apresentou como fácil de instalar, utilizar, com um design agradável e sem problemas graves de performance.

## 5. Conclusão

Este trabalho apresentou o processo de desenvolvimento de um aplicativo para o grupo de pesquisa MIND.Funga que visa a identificação de macrofungos. O objetivo foi criar uma ferramenta que auxiliasse na coleta de informações sobre fungos e que também promovesse o estudo e popularização da micologia.

O aplicativo foi criado com o Framework Ionic e foi testado em dispositivos Android, tendo sucesso na comunicação com a API e o backend. O aplicativo foi utilizado por membros do grupo MIND.Funga e teve boa aceitação, sendo avaliado como fácil de usar e instalar.

O aplicativo pode ser utilizado pelo grupo em pesquisas futuras e pode ser atualizado com novos desenvolvimentos, tornando-se uma importante ferramenta para gerenciar e expandir o banco de dados de espécies de fungos no território estudado pelo grupo.

### **5.1. Sugestões para trabalhos futuros**

É possível sugerir algumas melhorias e adições para futuras versões do aplicativo.

As sugestões incluem:

- A criação de uma rede neural que possa ser utilizada offline;
- Melhorias na interface do aplicativo;
- Desenvolvimento da edição e adição de espécies;
- Apresentação de dados sobre o número de envios de cada usuário;
- Criação de uma tela de mapa de registros;
- Adição de outros filtros e campos de pesquisa nas telas desenvolvidas;
- Melhorias na rede neural e na API;
- Publicação do aplicativo na Google Play Store;
- Incluir protocolo de captura de imagens de macrofungos nas dicas de fotografia.

## **6. Referências**

- KITCHENHAM, Barbara. Procedures for Performing Systematic Reviews. [S.l.], 2004.
- OLIVEIRA, JC. Tópicos em Micologia Médica. 4º ed. Rio de Janeiro: ControlLab, 2014. Disponível em: [https://controllab.com/pdf/topicos\\_micologia\\_4ed.pdf](https://controllab.com/pdf/topicos_micologia_4ed.pdf). Acesso em: 13 out.2022.
- DICIONÁRIO AMBIENTAL. O que é uma Espécie e um Espécime. 2014. Disponível em: <https://oeco.org.br/dicionario-ambiental/28502-o-que-e-uma-especie-e-um-especime>. Acesso em: 13 out.2022.
- DRECHSLER, Elisandro. An Infrastructure for Cataloguing and Classifying Macrofungi using Convolutional Neural Networks. 2022. Em fase de elaboração.
- FARIAS, Roger de Córdova. Desenvolvimento de API para classificação de macrofungos. 2022. Em fase de elaboração.

IONIC. Ionic Framework. 2022. Disponível em: <https://ionicframework.com/>. Acesso em: 17 out.2022.

SANCHEZ, ANA. Ionic is composed. 2018. Disponível em: [https://medium.com/@asanchez\\_77574/ionic-is-composed-the-mvc-model-view-controller-or-model-view-controller-e47566a1dd2e](https://medium.com/@asanchez_77574/ionic-is-composed-the-mvc-model-view-controller-or-model-view-controller-e47566a1dd2e). Acesso em: 20 out.2022.

WEBSTER, J. WEBER, R. Introduction to fungi. Cambridge: New York. [S.l.], 2007

MUELLER, G.M; SCHMIT, J.P; LEACOCK, P.R; et al. Global diversity and distribution of macrofungi. 2007. Disponível em: <https://doi.org/10.1007/s10531-006-9108-8>. Acesso em: 17 dez.2022.

## APÊNDICE B – CÓDIGO FONTE

Os códigos fontes desenvolvidos neste trabalho foram compactados em um arquivo .zip, e estão disponíveis para download no link abaixo:

- <https://arquivos.ufsc.br/f/74aedec35882495f9466/>

Alguns passos que podem ser úteis para rodar o aplicativo na sua máquina:

- Instalar o IONIC e o NPM;
- Descompactar os arquivos;
- Dentro da pasta descompactada executar o comando: *npm install* ;
- Após a instalação dos pacotes, utilizar *ionic serve*, para rodar a aplicação em browser para debug, desenvolvimentos e visualização rápida da interface ;
- Se recomenda gerar um .apk para teste da aplicação em dispositivos android. Execute o comando: *ionic cordova build android para isso.*  
( em caso de sucesso no build o arquivo .apk será gerado em `tcc_alfeu\platforms\android\app\build\outputs\apk\debug\` )

Caso queira instalar o aplicativo desenvolvido, o mesmo se encontra no link abaixo:

- <https://arquivos.ufsc.br/f/2b141826262d4e7794f8/>