



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
ICTS Campus Sorocaba

Flávio Henrique Fernandes Santino

**APLICATIVO DE LEITURA DE *TAG DIGITAL* PARA
IDENTIFICAÇÃO DE PLANTAS MEDICINAIS**

Sorocaba

2022

Flávio Henrique Fernandes Santino

**APLICATIVO DE LEITURA DE *TAG DIGITAL* PARA
IDENTIFICAÇÃO DE PLANTAS MEDICINAIS**

Trabalho de conclusão de curso apresentado ao Instituto de Ciência e Tecnologia de Sorocaba da Universidade Estadual Paulista “Júlio de Mesquita Filho” como parte dos requisitos exigidos para a obtenção do título de Bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Dr. Márcio Alexandre Marques

Coorientadora: Prof^ª. Ms. Patricia Moriguchi

Sorocaba

2022

S235a

Santino, Flávio Henrique Fernandes

Aplicativo para leitura de tag digital para identificação de plantas medicinais / Flávio Henrique Fernandes

Santino. -- Sorocaba, 2022

69 p. : il., tabs., fotos

Trabalho de conclusão de curso (Bacharelado Engenharia de Controle e Automação) - Universidade Estadual Paulista (Unesp), Instituto de Ciência e Tecnologia, Sorocaba

Orientador: Marcio Alexandre Marques

Coorientadora: Patricia Moriguchi

1. Engenharia de Software. 2. Aplicativos móveis. 3. QR codes. 4. Banco de dados. 5. Plantas medicinais. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Ciência e Tecnologia, Sorocaba. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada

FORMULÁRIO DE AVALIAÇÃO DO TRABALHO DE GRADUAÇÃO

Nome do Aluno:	Flávio Henrique Fernandes Santino
Título do Trabalho:	Aplicativo de Leitura de Tag Digital para Identificação de Plantas Medicinais
Banca	
Orientador (O)	Prof. Dr. Márcio Alexandre Marques
Avaliador (A1):	Prof. Dr. Everson Martins
Avaliador (A2):	Profa. Me. Patrícia Moriguchi (UNIP/Sorocaba)

AVALIAÇÃO

1. Avaliação do Orientador quanto ao Desenvolvimento do Trabalho (Orientador) (N1) - Peso 25% da Nota Final (Notas de 0 a 10).

Item	Nota
Avaliação do(s) Relatório(s) Parcial(is) (60%)	10,0
Gerenciamento do TG (20%)	10,0
Grau de iniciativa do estudante (20%)	10,0
Média (N1)	10,0

2. Avaliação da Apresentação do Trabalho de Graduação (N2) – PESO 15% da Nota Final (Notas de 0 a 10).

Item	O	A1	A2	Média
Apresentação oral (40%)	10,0	10,0	10,0	10,0
Organização do material (40%)	10,0	10,0	10,0	10,0
Postura do Aluno (20%)	10,0	10,0	10,0	10,0
Média				10,0

3. Avaliação do Relatório Final e Artigo (N3) – PESO 60% da Nota Final (Notas de 0 a 10).

Item	O	A1	A2	Média
Relatório Final: estrutura do relatório, pesquisa bibliográfica, metodologia, qualidade técnico-científica, redação, etc. (60%)	10,0	10,0	10,0	10,0
Artigo: definição do problema, pesquisa bibliográfica, originalidade da proposta, redação. (40%)	10,0	10,0	10,0	10,0
Média				10,0

4. Parecer: (X) Aprovado sem alterações () Aprovado com alterações () Reprovado

5. Alterações Exigidas:

6. Nota Final **10,0**

Sorocaba, 08 de dezembro de 2022





Orientador

Avaliador 1

Avaliador 2

AGRADECIMENTOS

Primeiramente, gostaria de agradecer à Unesp por propiciar as condições necessárias para meu aprendizado acadêmico e desenvolvimento pessoal.

A minha família pelo amor, incentivo apoio durante toda minha jornada.

Ao corpo docente da Unesp Sorocaba, pelas oportunidades de aprendizado e incentivos durante minha graduação.

Ao meu professor e orientador, Dr. Marcio Alexandre Marques pelo suporte na definição do tema, desenvolvimento e correções deste trabalho.

À professora e coorientadora deste trabalho, Ms. Patricia Moriguchi pelo apoio no fornecimento dos dados para o desenvolvimento do trabalho e testes na aplicação.

Aos servidores da Universidade, que de alguma forma apoiaram durante o desenvolvimento das atividades do curso.

E a todos que fizeram parte da minha formação e me apoiaram, direta ou indiretamente, o meu agradecimento.

RESUMO

Com a evolução da tecnologia e conseqüentemente com a democratização do acesso à internet por meio de dispositivos móveis, com ênfase no segmento de *smartphones*, é crescente o número de ferramentas desenvolvidas para atender às necessidades do usuário. Segundo dados referentes ao ano de 2020, o Brasil possui um crescimento de 55% nas instalações de aplicativos nos últimos dois anos, estando significativamente acima da média de outros países. Associando o potencial cenário para o desenvolvimento de soluções *mobile* em conjunto com a necessidade de identificação de plantas medicinais armazenadas no horto de plantas medicinais da Universidade Paulista (UNIP/Sorocaba), e utilizadas como objeto de estudo para o curso de graduação em Farmácia, planejou-se o desenvolvimento de uma solução para o problema. O aplicativo funciona através da identificação do código de identificação da planta, individual para cada espécie, retornando ao usuário as informações principais das espécies catalogadas. Foi utilizado o ambiente de desenvolvimento de *software* (IDE) Android Studio para a implementação do aplicativo com as funções definidas, realização de testes e disponibilização da ferramenta. Como auxílio nesse processo, foram utilizadas ferramentas de gerenciamento de banco de dados em nuvem - *Firebase*, responsável pelo armazenamento dos dados que podem ser consultados pelo aplicativo. Após a implementação e realização de testes por vários usuários, os resultados indicaram facilidade no uso do aplicativo e, as funcionalidades de escaneamento do *QR code* da planta, consulta ao banco de dados e visualização de dados *offline* funcionaram como esperado, tendo sido utilizado como teste um conjunto de dados de 18 plantas medicinais.

Palavras-chave: Aplicativo, *QR Code*, Banco de Dados, Android Studio

ABSTRACT

With the evolution of technology and consequently the democratization of internet access through mobile devices, especially on the smartphone segment, the number of mobile tools (applications) developed to meet user needs is growing. According to data for the year 2020, Brazil has seen a 55% growth in app installations in the last two years, significantly above the average of other countries. Associating the potential scenario for the development of mobile solutions together with the need to identify medicinal plants stored in the medicinal plant garden of Paulista University (UNIP/Sorocaba), used as an object of study for the undergraduate course in Pharmacy, the development of a solution to the problem was planned. The application works by identifying the identification code (QR Code) of the plant, individual for each species, returning to the user the main information of the cataloged species. The Android Studio software development environment (IDE) was used to implement the application with the defined functions, perform tests and make the tool available. As an aid in this process, cloud database management tools were used - Firebase, responsible for storing the data that can be consulted by the application. After implementation and testing by several users, the results indicated ease of use of the application and the features of scanning the QR code of the plant, querying the database and viewing offline data worked as expected, having been used as a test a dataset of 18 medicinal plants.

Key words: *Application, QR Code, Database Android Studio*

LISTA DE FIGURAS

Figura 1 - Diagrama representando a forma de utilização do aplicativo.....	14
Figura 2 - Exemplo de lousa digital em sala de aula.....	16
Figura 3 - Fluxograma simplificado do processo de testes de uma aplicação.....	20
Figura 4 - Exemplo de código do tipo <i>QR Code</i>	21
Figura 5 - Exemplo do processo de leitura de um <i>QR Code</i>	21
Figura 6 - Representação esquemática da troca de informações entre dispositivos e banco de dados em nuvem	22
Figura 7 - Interface do Android Studio	25
Figura 8 - Painel de Analytics do Firebase.....	26
Figura 9 - Exemplo de <i>QR Code</i> contendo o texto “Projeto de PFC da Unesp Sorocaba”.....	27
Figura 10 - Exemplo de código de barras contendo o texto “Unesp-2022”.....	28
Figura 11 – Exemplo de visualização de um arquivo no formato PDF a ser apresentado na tela do <i>smartphone</i>	29
Figura 12 – Diagrama esquemático que representa diferentes formas de armazenamento de informações e múltiplos pontos de acesso a um servidor em nuvem.....	31
Figura 13 – Captura de tela da seção de armazenamento do Firebase, contendo os arquivos armazenados no servidor.	32
Figura 14 – Interface da plataforma de criação de códigos QR.	33
Figura 15 – Desenho esquemático exemplificando a forma como os dados de um código QR são interpretados.	34
Figura 16 – Diagrama esquemático representando o fluxo de informações ao realizar uma consulta no aplicativo	36
Figura 17 - Diagrama esquemático representando o funcionamento do aplicativo	38
Figura 18 - Captura de tela do Firebase Console	39
Figura 19 - Tela inicial da aplicação no Android Studio.....	40
Figura 20 - Desenvolvimento da tela "Histórico"	41

Figura 21 - Tela "Sobre" do aplicativo.....	42
Figura 22 - Fluxograma contendo as instruções executadas para o escaneamento do código QR e a exibição de informações na tela	44
Figura 23 - Diagrama esquemático representando a composição do endereço para <i>download</i> do arquivo.....	45
Figura 24 - Sequência de instruções (simplificada) durante o processo de visualização das últimas consultas realizadas.....	48
Figura 25 - Exemplo da criação dos arquivos em PDF para ser carregado no banco de dados.	50
Figura 26 - Informações das plantas armazenadas no banco de dados.	51
Figura 27 - Captura de tela do processo de instalação do aplicativo.....	52
Figura 28 - Telas da aplicação após a instalação.....	53
Figura 29 - Capturas de tela do processo de escaneamento da <i>tag</i>	53
Figura 30 – Captura de tela da consulta realizada utilizando a aplicação	54
Figura 31 - Mensagem de erro.....	55

LISTA DE ABREVIATURAS E SIGLAS

IDE - *Integrated development environment*

API - *Application Programming Interface*

QR - *Quick Response*

PDF - *Portable Document Format*

URL - *Uniform Resource Locators*

XML - *Extensible markup language*

GB - *Gigabyte*

SUMÁRIO

1. INTRODUÇÃO	12
2. REVISÃO BIBLIOGRÁFICA	15
2.1 O uso de tecnologia na área de ensino	15
2.2 Sistema Operacional Android	16
2.3 Desenvolvimento de <i>softwares</i>	17
2.3.1 Desenvolvimento de <i>softwares</i> para aplicativos móveis	17
2.3.2 Desenvolvimento de aplicativos para Android.....	17
2.3.3 Definições e especificações	18
2.3.4 Aplicativos nativos	19
2.3.5 Testes de <i>software</i>	19
2.4 Aplicativos de escaneamento de <i>QR Code</i>	20
2.5 Firebase	21
2.6 Kit de desenvolvimento de aplicações para Android	22
2.7 Formato de armazenamento de arquivo - PDF	23
3. METODOLOGIA	24
3.1 Ferramentas utilizadas.....	24
3.1.1 Android – Sistema operacional.....	24
3.1.2 Android Studio	25
3.1.3 Firebase.....	25

3.1.4	<i>QR Code</i>	27
3.1.5	Google <i>ML kit</i>	28
4.	DESENVOLVIMENTO	29
4.1	Definição das ferramentas utilizadas	29
4.1.1	Formato de armazenamento.....	29
4.1.2	Plataforma de armazenamento de informações em nuvem	30
4.1.3	<i>Tag</i> de identificação.....	33
4.1.4	Algoritmo de interpretação do <i>QR Code</i>	34
4.1.5	Arquitetura do <i>software</i>	35
4.1.6	Interface do usuário	36
4.2	Implementação do Aplicativo	37
4.2.1	Configuração do Firebase	38
4.2.2	Desenvolvimento do <i>layout</i> da tela inicial	39
4.2.3	Desenvolvimento da tela “Histórico”	41
4.2.4	Desenvolvimento da tela “Sobre”.....	42
4.2.5	Desenvolvimento do código em Java	42
4.2.6	Definição de parâmetros gerais e permissões.....	49
5.	RESULTADOS	50
5.1	Inserção das informações no banco de dados	50
5.2	Instalação no <i>smartphone</i> e utilização	51
5.3	Tratamento de erros	54

5.4	Teste do Aplicativo	55
5.5	Vídeo da Utilização do Aplicativo	56
6.	CONCLUSÕES	57
	REFERÊNCIAS	58
	GLOSSÁRIO	61
	APÊNDICE A	63
	APÊNDICE B.....	64
	APÊNDICE C	65

1. INTRODUÇÃO

O avanço rápido de soluções e inovações geradas através da tecnologia está cada vez mais presente no cotidiano da maioria da população, marcado pelo aparecimento de ferramentas projetadas para enfrentar problemas diários como, por exemplo, soluções de saúde, automação residencial, intermediação de compras e vendas e acesso a ferramentas financeiras como bancos e corretoras. Além disso, nota-se cada vez mais o desenvolvimento de soluções focadas em aplicações específicas para determinadas áreas, incluindo soluções em escala empresarial, industrial e acadêmica (CHAN, 2021).

A amplificação do acesso às novas tecnologias é, em sua maior parte, favorecida pelo desenvolvimento de soluções *mobile*, cuja escala de aplicação aumenta exponencialmente nos últimos anos, principalmente em decorrência do maior poder de processamento e conectividade desses dispositivos, associada à facilidade de atingir várias camadas da população (LAVADO, 2021), independentemente da localização geográfica e com um custo relativamente baixo quando comparado a outros meios tradicionais. Segundo dados referentes ao ano de 2020, o Brasil possui um crescimento de 55% nas instalações de aplicativos nos últimos dois anos, estando significativamente acima da média de outros países (APPSFLYER, 2021)

Com uma demanda cada vez maior pela criação de soluções e desenvolvimento em plataformas como computadores ou *smartphones*, nota-se um crescente aumento no número de ferramentas disponíveis para o desenvolvimento de soluções como, por exemplo, IntelliJ IDEA e Android Studio. A solução IntelliJ IDEA é um ambiente de desenvolvimento integrado (IDE) que abrange o desenvolvimento de várias soluções, tanto para computadores como para *smartphones* (JETBRAINS, 2022). O Android Studio é também um IDE, porém com foco em desenvolvimento exclusivamente para aplicativos *mobile* para o sistema operacional Android (ANDROID SUTDIO, 2022).

Ambas as soluções possuem ferramentas que facilitam o desenvolvimento como, por exemplo, simulação de dispositivos Android durante todo o processo, acesso a bibliotecas de código e a ferramentas já prontas para determinados processos rotineiros durante a execução dos aplicativos, sendo uma ferramenta completa para uso profissional e intuitiva para usuários que ainda não possuem tanto conhecimento em programação.

Com uma ampla gama de soluções disponíveis para o desenvolvimento de aplicações para dispositivos móveis, associadas a necessidade de democratização do acesso à informação por

meio da tecnologia, de forma a garantir a transmissão de informações claras, concisas e verificadas a fim de permitir que os usuários, independentemente de níveis sociais ou educacionais, possam ser empoderados com informações, dentre elas, as capazes de trazer melhorias quanto a saúde e bem-estar e, além disso, informações de saúde, fortemente presentes nesse contexto, apresenta-se como um forte veículo de garantia de direitos de cidadania ao usuário, tornando-se um potencial promotor do conhecimento e garantindo mais autonomia aos cidadãos (LEITE, et al, 2014).

Incrementar ao aspecto social, informações sobre plantas medicinais comumente utilizadas para fins educacionais, incluindo a identificação de plantas para o curso de farmácia da Universidade Paulista (UNIP), câmpus de Sorocaba, mostram-se como uma boa oportunidade para a disseminação de informações relevantes à sociedade através da tecnologia. O desenvolvimento do aplicativo para Android para a identificação de *tags* em plantas medicinais tem como motivo a modernização do atual sistema de identificação de plantas medicinais, presentes no horto de plantas medicinais da Universidade, com potencial para expansão para outros projetos de cunho social.

A nova solução proposta favorecerá uma identificação mais fácil e precisa da grande quantidade de plantas presentes no ambiente, trazendo mais clareza ao usuário quanto às informações relevantes das plantas, tornando o processo mais ágil e preciso quando comparado ao método atual de identificação. Além disso, o aplicativo contará com um “Histórico” com todas as informações das plantas já pesquisadas e que podem ser facilmente consultadas, além desse recurso funcionar de forma *offline*.

O processo de identificação, exemplificado na Figura 1, ocorrerá através do escaneamento de um código do tipo QR contido em uma etiqueta (*tag*) fixada na planta. Após a leitura, as informações lidas pela câmera do *smartphone* possibilitarão a identificação da planta e, através da consulta em um banco de dados em nuvem, será exibido para o usuário as informações relevantes para a identificação e estudo da planta, bem como suas indicações e aplicações.

Figura 1 - Diagrama representando a forma de utilização do aplicativo



Fonte: Autoria própria

2. REVISÃO BIBLIOGRÁFICA

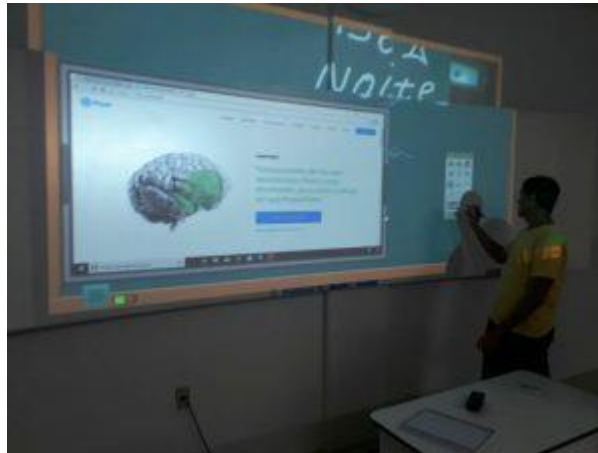
2.1 O uso de tecnologia na área de ensino

O uso de ferramentas tecnológicas associadas a educação é conhecido por uma relação morosa, especialmente quando o assunto está relacionado ao uso dessas ferramentas em sala de aula. O uso de dispositivos eletrônicos como ferramenta de aprendizado tem se mostrado crescente nos últimos anos, inclusive em sala de aula, favorecendo o uso dessas tecnologias para o aprendizado. Muitos estudos têm destacado os benefícios do uso de ferramentas tecnológicas para a ampliação da aprendizagem, permitindo a integração de diferentes tipos de mídias e tecnologias (GRESCZYSCZYN; FILHO; MONTEIRO, 2016).

Dentre as inúmeras tecnologias utilizadas para fornecer auxílio ao ensino, de forma a tornar o processo de aprendizado mais dinâmico e atrativo, tem-se a tecnologia denominada *Learning Management System (LMS)*, sendo chamada de Ambiente Virtual de Aprendizagem (AVA) no Brasil. A tecnologia se baseia em um ambiente virtual de aprendizagem, disponível via internet, capaz de fornecer informações sobre livros, exercícios personalizadas, vídeo aulas e exercício de aprendizado *gamificado* (BARROS, 2019).

Considerando a necessidade de facilitação do aprendizado, a lousa digital é um dos recursos tecnológicos que vem sendo utilizado recentemente. Esse recurso possibilita ao professor o enriquecimento das aulas, possibilitando que os materiais sejam atualizados constantemente, permitindo a exibição de conteúdo audiovisual como vídeos explicativos ou filmes. No contexto educacional, a lousa digital ilustrada na Figura 2, apresenta-se como a possibilidade de integração dos principais recursos multimídia em um dispositivo, tornando o cotidiano da aprendizagem mais dinâmico e intuitivo (NAKASHIMA; BARROS; AMARAL, 2009)

Figura 2 - Exemplo de lousa digital em sala de aula



Fonte: Autoria desconhecida

2.2 Sistema Operacional Android

O sistema operacional para dispositivos móveis Android tem, como base, outro sistema operacional denominado Linux, que por sua vez, é o resultado da evolução de outros sistemas operacionais anteriores, sendo conhecido pela sua grande compatibilidade com diversas plataformas e suas bibliotecas de código aberto, fazendo com que possa ser altamente personalizável (TANENBAUM; BOS, 2016).

O sistema Android se destaca, de forma pouco comum, como um sistema operacional baseado em código aberto em conjunto com aplicações de terceiros com código fechado. Considerando esse cenário, uma meta importante do sistema é fornecer um ambiente rico para aplicações de terceiros, garantindo a estabilidade e a compatibilidade do sistema em diversas aplicações para vários dispositivos de fabricantes diferentes. Além do mais, o sistema possui como requisitos o suporte a interações comuns de dispositivos móveis como, por exemplo, envio e recebimento de mensagens SMS, conexão com operadoras de telefonia móvel e garante a eficiência desses processos para otimizar o gerenciamento da bateria (TANENBAUM; BOS, 2016).

O modelo de desenvolvimento de aplicações do sistema Android se mostra diferente do modelo de desenvolvimento do sistema predecessor, o Linux, sendo a aplicação desenvolvida para Android definida por um arquivo “pacote” que contém vários elementos necessários para a instalação e execução do aplicativo como, por exemplo: o código fonte, elementos visuais e recursos gráficos, declarações sobre o que

está contido no sistema, critérios de relacionamento com o dispositivo e/ou outras aplicações dentre outros (TANENBAUM; BOS, 2016).

O arquivo contendo essas informações, necessário para a instalação da aplicação no dispositivo, é denominado *apk* (Android *package*).

2.3 Desenvolvimento de *softwares*

2.3.1 Desenvolvimento de *softwares* para aplicativos móveis

Cada vez mais os sistemas operacionais são obrigados a operarem como sistemas distribuídos, ou seja, devem ser executáveis em diversas plataformas e relacionar-se com diversos tipos de outros sistemas. Além da já conhecida execução de *softwares* em computadores pessoais, a presença do uso de aplicações em dispositivos móveis também se mostra necessária no contexto atual. Muitas vezes é necessário realizar a integração de *softwares* novos com sistemas mais antigos, denominados legados, escritos em diferentes linguagens de programação (SOMMERVILLE, 2021).

Para o desenvolvimento de aplicativos para dispositivos móveis é necessário o conhecimento específico das tecnologias utilizadas para a plataforma em que se deseja executar a aplicação. Além do mais, cada um dos aplicativos pode ser desenvolvido em um ambiente de desenvolvimento específico da plataforma, possuindo compatibilidade com uma ou mais linguagens de programação (SILVA; SANTOS, 2014).

2.3.2 Desenvolvimento de aplicativos para Android

Um arquivo de uma aplicação para o sistema operacional Android, denominado “*apk*”, é definido por um conjunto de itens necessários para a interpretação das funcionalidades da aplicação pelo sistema operacional, sendo os principais:

1 – Manifesto: Arquivo com definições gerais da aplicação, contendo detalhes sobre suas funcionalidades e como ela deve ser executada pelo sistema. Também é responsável pelo fornecimento do nome de *package* da aplicação, sendo este um nome único que se refere ao aplicativo;

2 – Recursos necessários para a aplicação: Nesta secção, se destacam os arquivos XML utilizados para definir os elementos visuais, incluindo as telas que serão exibidas para o usuário;

3 – O código em si, que define as instruções a serem realizadas pela aplicação;

4 – Uma assinatura digital, possibilitando a identificação com segurança do autor (TANENBAUM; BOS, 2016).

A definição da interação da aplicação com o usuário ocorre por meio de elementos chamados “Atividades”, responsáveis por responder aos comandos dos usuários e executar as tarefas programadas. Por exemplo, ao abrir uma aplicação, automaticamente uma atividade é executada, contendo as informações a serem exibidas na tela inicial e atividades de segundo plano pré-definidas pelo desenvolvedor.

De forma a facilitar o descobrimento e, conseqüentemente, a inicialização das atividades em uma aplicação o sistema Android possui um recurso denominado Intento, sendo este um mecanismo utilizado para, por exemplo, chamar uma nova atividade ao clicar em um determinado botão, bem como outras funcionalidades (TANENBAUM; BOS, 2016).

2.3.3 Definições e especificações

Os requisitos para o desenvolvimento de um *software* devem ser claramente definidos antes do desenvolvimento. Ele deve ser expresso em termos de fenômenos que ocorrem no mundo real. Alguns desses fenômenos são controlados pelo mundo e observados pela máquina – por meio de sensores – ou controlados pela máquina e observados pelo mundo – por meio de atuadores. A máquina é construída com o objetivo de alcançar o atendimento aos requisitos (SOMMERVILLE, 2021).

A especificação dos requisitos descreve restrições à fenômenos compartilhados que devem ser impostos pelo programa a ser desenvolvido.

Como parte atuante na definição das especificações, o engenheiro de *software* tem como responsabilidade projetar e implementar um programa que seja adequado, ou seja, que se comporte de acordo com as especificações previamente definidas (SOMMERVILLE, 2021).

Para entender de forma adequada os requisitos e realizar o projeto do *software*, o engenheiro precisa entender como a parte afetada – o ambiente de implementação – se comporta (ou espera-se que se comporte), porque isso pode afetar a forma como a máquina irá atender aos requisitos (SOMMERVILLE, 2021).

2.3.4 Aplicativos nativos

Aplicativos nativos desenvolvidos para celulares podem ser definidos como aplicações que foram desenvolvidas para uma plataforma específica. O sistema operacional destas plataformas é responsável por gerenciar os recursos do dispositivo. O desenvolvimento destas aplicações pode ocorrer através do uso de ambientes de desenvolvimento destinados à plataforma escolhida e programado com sua(as) respectiva(s) linguagem(s) de programação (SILVA; SANTOS, 2014).

Aplicativos nativos tem como principal limitação a plataforma em que é desenvolvido, causando um aumento de custos e do tempo de desenvolvimento caso seja necessário disponibilizar a aplicação para outra plataforma.

A interação de aplicações nativas com recursos disponíveis do dispositivo pode ser realizada através do uso de APIs, ferramentas que possibilitam a interação entre aplicações e outros elementos externos ao *software* (SILVA; SANTOS, 2014).

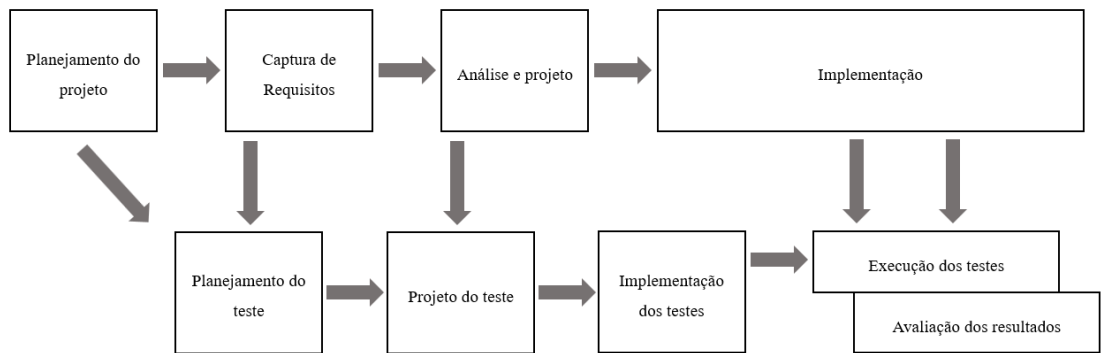
2.3.5 Testes de *software*

A realização de testes em um *software* é destinada a mostrar que um programa de computador executa as tarefas exatamente da forma como foi projetado para fazê-las. A execução de testes é, normalmente, realizada utilizando dados artificiais gerados especialmente para esta ocasião. São verificados os comportamentos quanto à erros, anomalias ou informações sobre atributos não funcionais do programa (SOMMERVILLE, 2021). O fluxograma apresentado na Figura 3 representa o processo de testes do *software* durante o desenvolvimento de uma aplicação.

Dentro do processo de realização de testes de um *software*, há dois objetivos distintos: 1 – demonstrar ao desenvolvedor e ao cliente que o programa atende a todos os requisitos e 2 – descobrir situações em que o comportamento do *software* não corresponde

ao esperado, podendo causar erros na execução como, por exemplo, travamentos, perda de informações e envio de informações indesejadas a outros sistemas (SOMMERVILLE, 2021).

Figura 3 - Fluxograma simplificado do processo de testes de uma aplicação



Fonte: Autoria própria

2.4 Aplicativos de escaneamento de *QR Code*

Considerando o universo de aplicativos disponíveis para dispositivos móveis, ferramentas que alinham e entrelaçam o *hardware* de dispositivos móveis (câmera, sensores, telas, entre outros) com realidade aumentada vem ganhando um espaço maior, incluindo a área de ensino (FERREIRA; RIBEIRO; CLEOPHAS, 2018).

O *QR Code*, exemplificado na Figura 4, também se utiliza de elementos de realidade aumentada, mas com o objetivo de armazenamento de dados que podem ser, posteriormente, convertidos em diversos tipos de informações, servindo como uma âncora física para espaços e ferramentas virtuais (FERREIRA; RIBEIRO; CLEOPHAS, 2018). O processo de leitura do *QR Code*, usualmente realizado através do uso da câmera de *smartphones*, pode ser visualizado na Figura 5.

Figura 4 - Exemplo de código do tipo *QR Code*



Fonte: Autoria própria

Figura 5 - Exemplo do processo de leitura de um *QR Code*

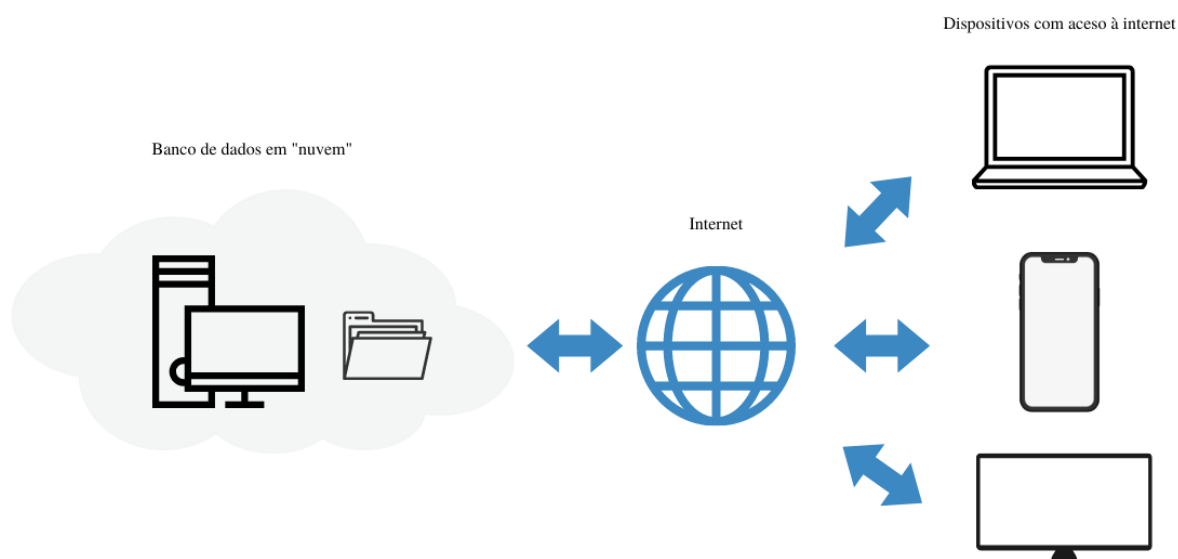


Fonte: Autoria própria

2.5 Firebase

O Firebase apresenta-se como uma ferramenta *online* que possui múltiplas funções complementares a um aplicativo, incluindo o recurso de armazenamento de informações do tipo banco de dados em nuvem e tem como principal objetivo fornecer ferramentas e infraestrutura para o desenvolvimento de aplicações em seu sistema operacional próprio, o Android. O diagrama esquemático que representa as conexões de um banco de dados em nuvem pode ser visualizado na Figura 6.

Figura 6 - Representação esquemática da troca de informações entre dispositivos e banco de dados em nuvem



Fonte: Autoria própria

Composta por múltiplas funcionalidades, entre elas, armazenamento de informações, autenticação, envio e recebimento de mensagens, acompanhamento das estatísticas de uso, *performance* da aplicação dentre outras funções, todas comumente usadas no desenvolvimento de aplicações Android, porém sendo disponibilizadas de forma mais acessível pela empresa (MORONEY, 2017).

Muitos dos recursos fornecidos pelo Firebase são disponibilizados pela Google sem custos para o desenvolvedor, incluindo o armazenamento de informações no banco de dados, com uma limitação de 1 GB de armazenamento com um limite de 100 conexões simultâneas.

O armazenamento de informações na plataforma possibilita acesso às informações do banco de dados por todos os dispositivos conectados, sendo toda a parte de execução das tarefas executadas localmente no dispositivo e não no servidor, como em outros tipos de banco de dados (MORONEY, 2017).

2.6 Kit de desenvolvimento de aplicações para Android

Após o grande crescimento do mercado de dispositivos móveis ao longo do tempo, tendo o Android uma participação significativa nessa expansão, viu-se a necessidade do

desenvolvimento de uma ferramenta para a criação de aplicações totalmente focada no sistema operacional, otimizando recursos e ferramentas a fim de tornar o desenvolvimento de aplicações para o sistema menos complexo (GERBER; CRAIG, 2015).

Uma ferramenta denominada Android Studio foi desenvolvida pela Google em conjunto com a empresa norte americana de tecnologia JetBrains, fazendo com que o Android Studio seja a plataforma oficial de desenvolvimento de aplicações para sistema operacional, contendo os recursos para o desenvolvimento de ponta a ponta (GERBER; CRAIG, 2015).

2.7 Formato de armazenamento de arquivo - PDF

O formato PDF foi desenvolvido pela multinacional norte-americana Adobe Systems em 1993 com o objetivo de representar textos, imagens, gráficos em um formato que independe do aplicativo, *hardware* e sistema operacional usado para criá-los. O PDF é um padrão de arquivo aberto, que possui licenças livres e pode ser facilmente gerado de forma gratuita por inúmeras ferramentas como, por exemplo, o editor de texto Microsoft Word (ADOBE, 2022).

3. METODOLOGIA

Considerando a necessidade de ferramentas para o desenvolvimento da aplicação para dispositivos Android, foram analisadas e escolhidas as que melhor atendem a necessidade do desenvolvimento do aplicativo.

3.1 Ferramentas utilizadas

Para o desenvolvimento do Projeto de Final de Curso (PFC) serão necessários o uso de programas de computador (*softwares*) destinados ao desenvolvimento de aplicativos bem como uma página de internet, de uso livre e gratuito, para a geração dos códigos do tipo QR a serem lidos pelo aplicativo.

Além disso, para a implementação do aplicativo, foi utilizado um computador pessoal do tipo portátil, da marca ASUS, modelo X510UR-BQ291T e um *smartphone* da marca Samsung, modelo Galaxy S8 com o sistema operacional Android 9.

3.1.1 Android – Sistema operacional

O sistema operacional Android é considerado relativamente novo e foi especialmente desenvolvido para dispositivos móveis como, *smartphones* e *tablets*. Desde a sua introdução, a participação do sistema operacional cresceu exponencialmente ao longo dos anos (TANENBAUM; BOS, 2016).

Grande parte do sucesso do sistema pode ser atribuído a disponibilidade livre para os fabricantes de dispositivos móveis bem como o fato da plataforma ser de código aberto, ou seja, pode ser customizada para diferentes dispositivos e suas respectivas aplicações.

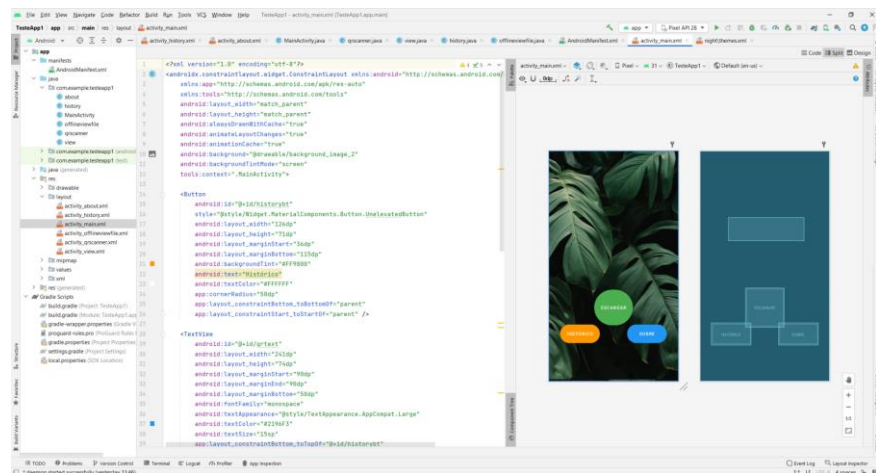
Considerando a ampla utilização do sistema, que se mostra como um forte vetor na era da digitalização, associada a uma gama de ferramentas disponíveis para desenvolvimento, o sistema operacional Android foi definido como a plataforma em que o aplicativo implementado nesse projeto será disponibilizado.

3.1.2 Android Studio

É um ambiente de desenvolvimento, com o foco especificamente em aplicações Android. Possui várias ferramentas que facilitam o processo de criação de aplicativos para a plataforma Android, incluindo fácil conexão com recursos do *smartphone* como, por exemplo, notificações. Possui compatibilidade com diversos sistemas operacionais, podendo ser instalado em Windows, MacOS e Linux (GERBER; CRAIG, 2015).

O *software* é gratuito e permite acesso aos vários recursos disponíveis para o desenvolvimento, possibilitando a concepção do aplicativo desde a etapa de projeto, definição de *layout*, botões e funções bem como a emulação do funcionamento de um dispositivo móvel Android já dentro da plataforma, facilitando os testes e aperfeiçoamentos do projeto. A interface de utilização pode ser visualizada na Figura 7.

Figura 7 - Interface do Android Studio



Fonte: Acervo pessoal

3.1.3 Firebase

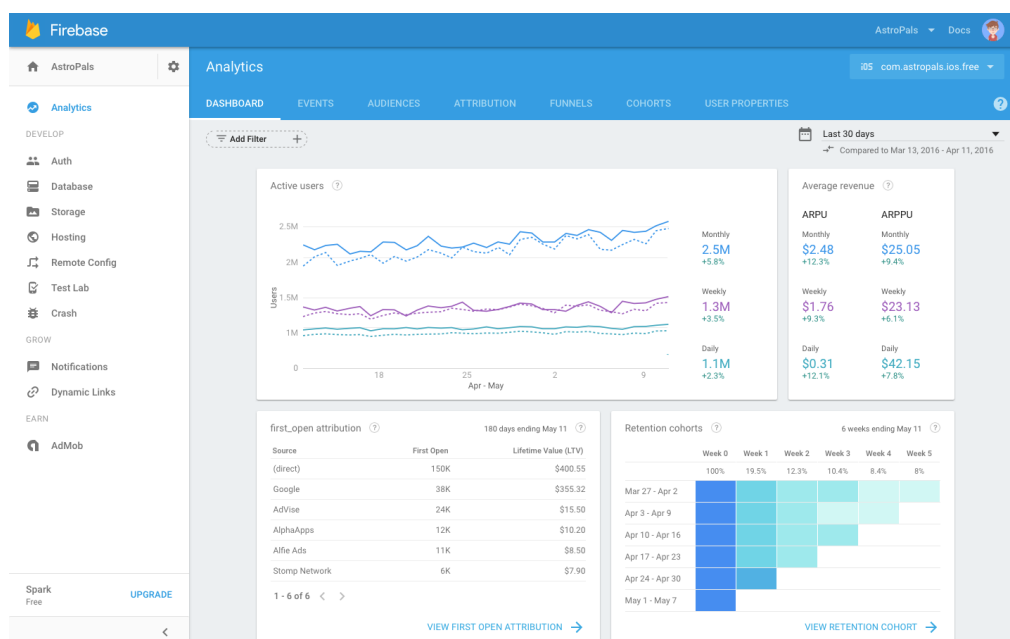
No planejamento da sequência de funcionamento da aplicação tem-se, após a captura do *QR Code* contido na *tag* da planta, a necessidade da utilização de um banco de dados para realização de consultas. O banco de dados disponível na nuvem, enviará as informações obtidas de volta ao dispositivo e permitirá a visualização do usuário.

Dessa forma, será necessário o uso de um serviço em nuvem capaz de armazenar essas informações bem como permitir a consulta delas pelo dispositivo.

O Firebase é um serviço em nuvem que se mostra adequado às necessidades do projeto, especialmente por ter foco em aplicações para dispositivos móveis. Assim como as outras ferramentas apresentadas anteriormente, também possuem uma interface intuitiva, exemplificada na Figura 8, que permite uma boa curva de aprendizado no serviço, facilitando o desenvolvimento de aplicativos.

Também possui muitas funcionalidades já integradas na plataforma, incluindo o recurso que permite que o desenvolvedor possa ter acesso a informações de erros ou eventuais problemas que possam ocorrer no aplicativo, facilitando o diagnóstico e tornando a entrega de reparos mais eficiente.

Figura 8 - Painel de Analytics do Firebase



Fonte: Acervo pessoal

Dessa forma, o serviço atuará como um banco de dados contendo as informações das plantas, como espécies, categorias, finalidade de uso e outros e, quando consultado, retornará com as informações dessa planta para o usuário.

O uso de uma metodologia de consulta em nuvem no projeto do aplicativo traz como principal vantagem um menor uso do espaço de armazenamento do dispositivo, já que não será necessário que o banco de dados completo esteja armazenado no *smartphone* do usuário, bem como a economia de recursos de processamento e memória para realizar a busca de informações.

Em contrapartida, tem-se que a solução será dependente de conexão com a internet para realizar novas consultas e apresentar as informações.

3.1.4 *QR Code*

O *QR Code* (Código QR), sigla para “*Quick Response*” em inglês, é um tipo de código em duas dimensões que contém barras e, de forma semelhante a outras matrizes bidimensionais como, o já conhecido código de barras, tem a capacidade de armazenar informações que podem ser lidas por outros dispositivos.

A leitura desse código pode ser realizada através de dispositivos eletrônicos como, por exemplo, a câmera de um *smartphone* moderno, que é capaz de ler as informações contidas no código e transmiti-las ao usuário. Por exemplo, as informações mais comumente armazenadas nesse código são endereços de páginas de *Web*, endereços de *e-mail*, números de telefones ou informações de geolocalização. A Figura 9 mostra um exemplo de *QR Code*.

Figura 9 - Exemplo de *QR Code* contendo o texto “Projeto de PFC da Unesp Sorocaba”.



Fonte: Autoria própria.

A principal diferença do *QR Code* para o código de barras, apresentado na Figura 10, é a sua maior capacidade de armazenamento de informações, com até 20 caracteres, que pode chegar a ser doze vezes maior quando comparado a um código de barras comum, em decorrência da sua capacidade de armazenamento de duas dimensões (GREGERSEN, 2022).

Figura 10 - Exemplo de código de barras contendo o texto “Unesp-2022”



Fonte: Autoria própria.

3.1.5 Google *ML kit*

O Google *ML kit* é um conjunto de ferramentas, que pode ser implementado dentro de uma aplicação e, dentre os recursos disponíveis, possui a funcionalidade de interpretar as imagens da câmera com a finalidade de identificar *QR Codes* e “traduzir” as informações do código para uma variável de texto, podendo ser interpretada pelo programa (GOOGLE DEVELOPERS, 2022)

Considerando o objetivo do projeto de identificar plantas através do código QR, a ferramenta do Google apresenta-se como uma opção para trazer a funcionalidade de leitura e interpretação de códigos do tipo QR no aplicativo.

4. DESENVOLVIMENTO

4.1 Definição das ferramentas utilizadas

A primeira etapa do desenvolvimento do aplicativo para a leitura de *tag* de identificação de plantas medicinais foi o entendimento dos requisitos de projeto e validação da sua exequibilidade, considerando as ferramentas disponíveis, de forma a garantir o bom andamento do desenvolvimento do *software*.

4.1.1 Formato de armazenamento

Primeiramente, foram analisados diversos formatos de arquivo e foi definido o mais adequado formato para o armazenamento das informações referentes as plantas medicinais. Nesta análise, foram levantados que as informações a serem exibidas na tela do usuário deveriam conter imagens e textos, a fim de possibilitar uma identificação visual da planta pelos usuários e a descrição das informações relevantes ao tipo de planta analisado e o formato de arquivo que melhor se adequa aos requisitos descritos é o *Portable Document Format* (PDF).

De forma a exemplificar a forma como as informações referentes a planta medicinal identificada serão exibidas, é possível visualizar na Figura 11 um exemplo de visualização de arquivo PDF contendo às informações referentes a uma planta.

Figura 11 – Exemplo de visualização de um arquivo no formato PDF a ser apresentado na tela do *smartphone*.



1. **Sinonímia Científica:** *Curcuma longa*
2. **Sinonímia Popular:** Cúrcuma, Açafração-da-índia e Açafração-da-terra.
3. **Parte Utilizada:** Raízes e rizomas.
4. **Finalidade de Utilização:** Usada contra histeria, gastralgia, coqueluche e na dentição das crianças. Ação anti-inflamatória, problemas de fígado e vesícula biliar, redução de colesterol e lipídios totais no sangue.

Fonte: Autoria própria.

Foram definidas as dimensões do documento a ser exibido de forma a facilitar a visualização e garantir uma boa maneabilidade das ferramentas em diversos dispositivos. A partir da definição inicial do projeto, que ressalta que a execução do aplicativo ocorrerá em um *smartphone*, foi definido um formato de exibição verticalizado compatível com as telas de proporção 16:9, sendo o formato mais comum em dispositivos móveis (BHAGAT; BAJAJ, 2022).

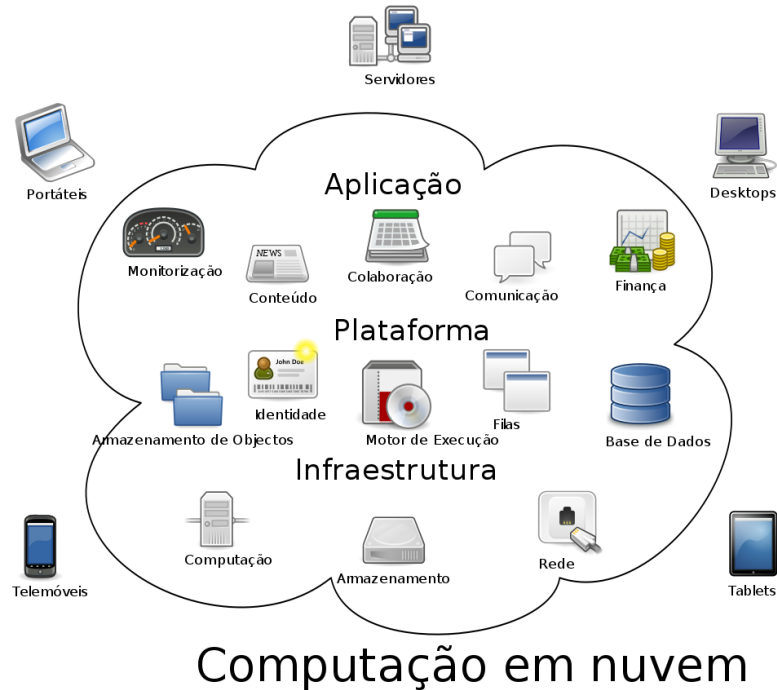
Considerando que, em alguns casos, a quantidade de informações a ser exibida pode ser maior do que uma página, será possível a rolagem vertical da página (*scroll*) para permitir a visualização de todo o seu conteúdo.

4.1.2 Plataforma de armazenamento de informações em nuvem

As informações das plantas medicinais permanecerão armazenadas em um banco de dados em nuvem, que pode ser acessado pelo aplicativo através da conexão com a internet e um código de autenticação, denominado *token*.

O armazenamento de dados em nuvem é um modelo de armazenamento de dados computacionais em que as informações ficam localizadas em múltiplos servidores, geralmente distribuídos geograficamente em locais diferentes. Esses servidores computacionais estão conectados à internet e podem ser acessados por aplicações para enviar e receber informações. A grande vantagem de um sistema de armazenamento de dados em nuvem é a sua segurança, confiabilidade e a disponibilidade imediata de leitura ou escrita de informações (FIREBASE, 2021). A arquitetura de um sistema de bancos de dados em nuvem pode ser observada na Figura 12.

Figura 12 – Diagrama esquemático que representa diferentes formas de armazenamento de informações e múltiplos pontos de acesso a um servidor em nuvem.



Fonte: Johnston, 2013

O acesso às informações, seja para gravação de novas informações ou leitura dos dados já existentes podem ocorrer de diversas formas. No caso do desenvolvimento do aplicativo para leitura de *tag* de plantas medicinais esse acesso ocorrerá através da API do Firebase, uma plataforma de armazenamento e gerenciamento de aplicações para dispositivos móveis.

O armazenamento das informações no Firebase, em forma de banco de dados, possibilita o armazenamento de forma gratuita de até 1 *gigabyte* de informações em seu servidor, que pode ser utilizado de forma semelhante a uma plataforma de armazenamento de arquivos em nuvem, como apresentado na Figura 13.

Figura 13 – Captura de tela da seção de armazenamento do Firebase, contendo os arquivos armazenados no servidor.



Fonte: Acervo pessoal

As informações referentes às plantas medicinais, em formato PDF, podem ser carregadas ao banco de dados de várias formas como, por exemplo, utilizando um *script* para o *upload* de arquivos em uma linguagem de programação ou também de forma manual.

Analisando os requisitos do projeto, nota-se que as informações referentes às plantas medicinais a serem armazenadas no banco de dados em nuvem são estáticas, ou seja, não precisam sofrer alterações de forma contínua. Desta forma, para o desenvolvimento do aplicativo, foi usada a opção de envio das informações relevantes de forma manual, através do *upload* dos arquivos em formato PDF ao servidor.

O acesso às informações armazenadas no servidor pode ser realizado através de diferentes meios, sendo um deles a criação de um endereço de *web* (*link*) individualizado a cada arquivo PDF. Também é possível realizar o *download* do arquivo e salvá-lo localmente no dispositivo. Cada *link* gerado possui um *token*, código de segurança que impede que a informação seja acessada por usuários não autorizados.

Além do armazenamento, o Firebase também permite o acompanhamento de determinadas informações da aplicação e do banco de dados, de forma *online*, facilitando o monitoramento do funcionamento do aplicativo desenvolvido, bem como, rastreamento de anormalidades de forma a facilitar a solução de problemas (FIREBASE, 2021).

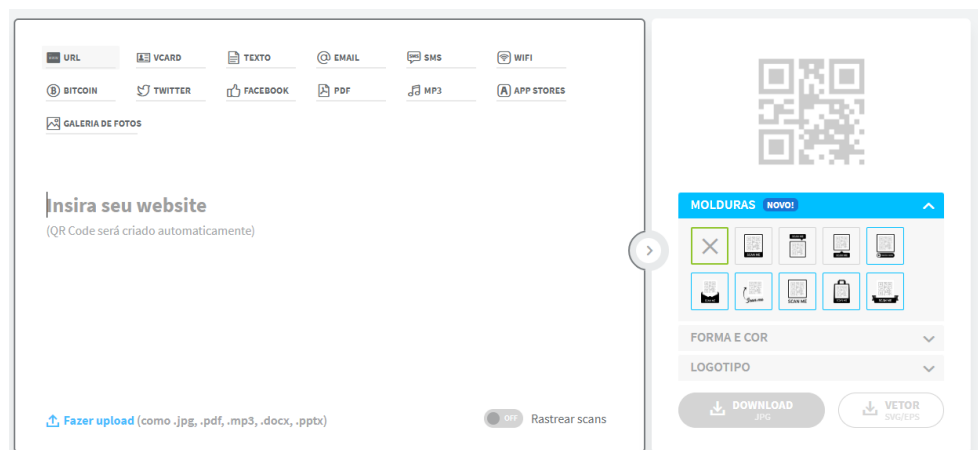
4.1.3 Tag de identificação

A *tag* que será utilizada para identificar a planta medicinal será do tipo *QR Code*, o que possibilita o armazenamento das informações a serem lidas pela câmera do *smartphone*. Cada uma das plantas possuirá sua própria *tag*, sendo única e contendo um endereço de *web* (URL) que possibilita o acesso ao arquivo contendo as informações relevantes para determinada planta.

A criação de um *QR Code* pode ser realizada através de diversas ferramentas, sendo algumas gratuitas e que podem ser acessadas pelo navegador de internet. Para o projeto, foi escolhido o uso da ferramenta de geração de código *QR*, *online* e gratuita, chamada *QR Code Generator* sendo fornecida pela empresa norte-americana Bitly Inc. e disponível através do endereço: <https://br.qr-code-generator.com/>.

A criação de um código através da ferramenta *online* pode ser realizada através da inserção da informação desejada como, por exemplo, um *link* e sem seguida, essa informação será convertida em um código do tipo *QR Code* que pode ser salvo localmente, como apresentado na Figura 14.

Figura 14 – Interface da plataforma de criação de códigos QR.



Fonte: Acervo Pessoal

4.1.4 Algoritmo de interpretação do *QR Code*


Para o funcionamento correto da ferramenta é necessário o uso de um algoritmo para a conversão dos dados do *QR Code* escaneado em um formato interpretável pelo *smartphone*. Semelhante ao que foi apresentado na secção 4.1.3, é necessária uma sequência de instruções que “traduza” as informações em formato de matriz contidas no código para um tipo interpretável pelo aplicativo, neste caso, sendo convertido em texto (*string*).

A ferramenta utilizada para realizar esta tarefa é o *Google ML Kit*, uma API que possibilita a leitura e interpretação de dados contidos no código. A ferramenta realiza a leitura e conversão do código de forma local, ou seja, utilizando recursos do próprio dispositivo, não sendo necessária a conexão com a internet para a execução da leitura.

O algoritmo não é exclusivo para códigos em forma de matriz como o código *QR*, mas também pode interpretar os tradicionais códigos de barra e formatos de código como Code 39, Code 93, Code 128, EAN-8, EAN-13, ITF, UPC-A, UPC-E. A leitura e detecção do tipo de código apresentado na câmera é feita automaticamente e o *kit* retorna em formato de *string* a informação lida. Como o objetivo do projeto é para a leitura de *tags* de plantas medicinais em formato de código *QR*, essa ferramenta se mostra compatível com as necessidades do projeto (GOOGLE DEVELOPERS, 2022).

A forma como o algoritmo decodifica a informação pode ser vista na Figura 15.

Figura 15 – Desenho esquemático exemplificando a forma como os dados de um código QR são interpretados.



Result	
Corners	(87,87) (612,87) (612,612) (87,612)
Raw value	WIFI:S:SB1Guest;P:12345;T:WEP;;
WiFi information	SSID SB1Guest
	Password 12345
	Type WEP

Fonte: Autoria própria.

4.1.5 Arquitetura do *software*

A arquitetura de um programa de computador refere-se à definição das estruturas fundamentais para o funcionamento do sistema. Cada uma das estruturas contém elementos de *software*, definição de relacionamento entre cada um dos elementos, bem como propriedades de cada uma das suas relações. Durante o desenvolvimento da arquitetura de um programa de computador são analisados vários fatores, como por exemplo:

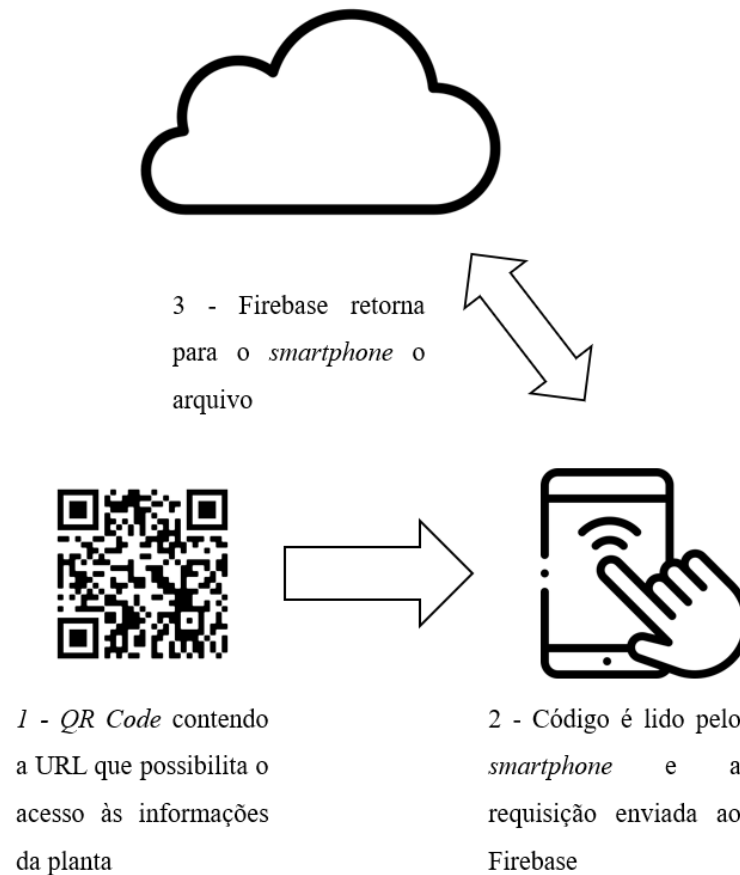
- Disponibilidade de recursos computacionais
- Possibilidade de acesso à internet
- Requisitos de *performance*
- Nível de confiabilidade
- Interface com o usuário

Analisando os requisitos do aplicativo, considerando fatores como: entregar uma boa usabilidade para o usuário e ser compatível com diversos dispositivos, foram estabelecidas algumas premissas para a definição da arquitetura:

- O processo de leitura do *QR Code* da planta deve possibilitar a interpretação das informações contidas no código e o retorno das informações relevantes da planta na tela do usuário
- Possibilidade de armazenamento local das informações para as consultas já realizadas
- Armazenamento das informações referentes às plantas medicinais em uma plataforma em nuvem
- Visualização das informações na tela do dispositivo de forma automática após escanear o código

Desta forma, a definição da arquitetura do *software* e sua consequente forma de funcionamento podem ser observadas na Figura 16.

Figura 16 – Diagrama esquemático representando o fluxo de informações ao realizar uma consulta no aplicativo



Fonte: Autoria própria

4.1.6 Interface do usuário

O termo “*Design* de interface de usuário” é utilizado para a descrição da ação de *design* de interfaces do tipo humano-máquina no desenvolvimento de *softwares* como por exemplo, aplicações para computadores, *smartphones*, eletrodomésticos dentre outras plataformas.

O principal objetivo da etapa de definição da interface gráfica do usuário é o foco na construção de interfaces como por exemplo, telas, *menus* e botões, de forma a garantir que a mensagem seja transmitida ao usuário de forma clara e concisa. Nesta etapa, também são consideradas formas de tornar a interação do usuário com a aplicação mais simples, com uma melhor curva de aprendizado para a utilização do aplicativo.

Considerando o principal uso do aplicativo desenvolvido neste trabalho, foram definidos, inicialmente, alguns requisitos de interface de usuário a fim de garantir uma boa experiência durante o uso da ferramenta, sendo:

- Uma tela inicial, contendo todos os botões para a realização das tarefas propostas do aplicativo
- Um botão para realizar a ação de escanear o *QR Code* que, ao ser selecionado, dá sequência nas ações para a leitura da *tag* e retorna as informações lidas ao usuário
- Um botão para a visualização do histórico de consultas, a fim de possibilitar que o usuário veja as últimas consultas realizadas. Nesta opção, também será possível a leitura das informações sobre as plantas medicinais já consultadas de forma *offline*, ou seja, mesmo sem o acesso à internet será possível acessar as informações já consultadas.
- Um botão com o título “Sobre” a fim de exibir informações relevantes do aplicativo como, por exemplo, a descrição das suas funcionalidades e dos autores.

4.2 Implementação do Aplicativo

Para o desenvolvimento e implementação do projeto de forma a garantir que a aplicação seja capaz de realizar as tarefas exibidas na Figura 17, iniciou-se com a configuração da plataforma de armazenamento em nuvem Firebase para, em seguida, realizar o desenvolvimento do aplicativo no Android Studio utilizando a linguagem de programação Java.

Figura 17 - Diagrama esquemático representando o funcionamento do aplicativo



Fonte: Autorial própria

4.2.1 Configuração do Firebase

Para a configuração da plataforma de armazenamento em nuvem e gerenciamento de aplicativos Firebase é necessário possuir uma conta Google e, em seguida, realizar o acesso na página de *web*: <https://firebase.google.com/>.

Como próximo passo, foi realizada a criação de um novo projeto, tendo inicialmente definido o nome do projeto como “QR Code – Plantas Mediciniais”. Após a criação do projeto, já é possível acessar o painel de controle com as informações relevantes da plataforma, denominada “Console” e realizar a navegação pelas páginas. O Console criado para a aplicação pode ser observado na Figura 18.

Figura 18 - Captura de tela do Firebase Console



Fonte: Acervo pessoal

Nesta etapa, também já é definido o endereço de *web* (URL) em que os arquivos serão armazenados e, posteriormente, consultados pelo aplicativo ao escanear a *tag* contida nas plantas.

Quanto ao uso de licenças e pacotes oferecidos pela plataforma, o plano básico é gratuito e atende aos requisitos de capacidade de armazenamento, número de requisições e recursos necessários para a correta execução do projeto, não sendo necessária a contratação de serviços adicionais para o bom funcionamento do aplicativo.

4.2.2 Desenvolvimento do *layout* da tela inicial

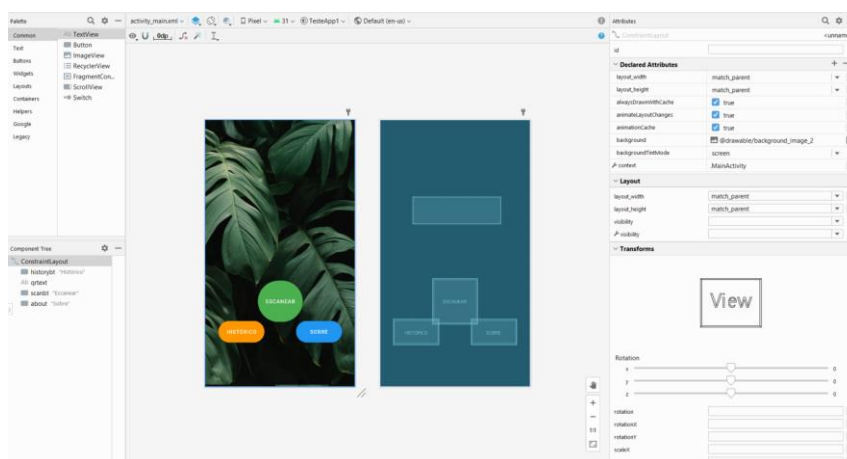
A primeira etapa no desenvolvimento da aplicação foi a criação do *layout* da tela inicial, de forma a realizar a criação dos botões definidos anteriormente, na fase de projeto, bem como o estabelecimento dos padrões de posicionamento de botões, paleta de cores e demais elementos visuais.

Para o desenvolvimento de telas, o ambiente de desenvolvimento utilizado possibilita dois meios diferentes que podem ser utilizados concomitantemente:

- 1 – Desenvolvimento por meio do uso da linguagem XML, utilizando a linguagem de programação para a definição de elementos visuais, nomenclatura, elementos de texto e posicionamentos dos elementos criados em relação às bordas da tela.

- 2 – Uso da interface visual do próprio ambiente de desenvolvimento para a inserção de elementos visuais, posicionamento dos elementos, edição de parâmetros de cores, definição de nomenclaturas e inserção de imagens, como apresentado na Figura 19.

Figura 19 - Tela inicial da aplicação no Android Studio



Fonte: Acervo pessoal

Foi utilizada, majoritariamente, a interface visual do ambiente de desenvolvimento para a criação das telas durante o desenvolvimento do projeto. Os elementos visuais e botões foram definidos de forma a atenderem os requisitos inicialmente planejados, sendo:

- Botão “Escanear”: Ao ser pressionado, iniciará a câmera do *smartphone* para a leitura do *QR Code* e, ao ser identificado, executará a consulta da informação no banco de dados do Firebase e retornará para o usuário as informações relacionadas à planta medicinal atrelada ao código escaneado. No *background*, será realizado o *download* automático das informações para posterior consulta *offline* na página “Histórico”.
- Botão “Histórico”: Esse botão possibilita a visualização das plantas já escaneadas, sendo que as informações das consultas já realizadas estão armazenadas localmente, não sendo necessária a conexão com a internet para a visualização.

- Botão “Sobre”: Carrega uma tela contendo informações relevantes sobre o aplicativo.

4.2.3 Desenvolvimento da tela “Histórico”

Para a tela histórico, foi necessária a utilização de um recurso visual do ambiente de desenvolvimento nativo do Android denominado *listView*. O elemento *listView* possibilita a exibição de dados em forma de lista bem como incorpora diversos recursos necessários para a exibição de dados como, por exemplo, possibilidade de rolagem da página, a funcionalidade de clique para que os elementos da lista sejam interativos, dentre outros. A configuração e definição das funcionalidades do elemento ocorre através do código e da lógica de programação definida para as funcionalidades desejadas.

A tela “Histórico” do aplicativo pode ser observada na Figura 20, com o destaque (à direita) dos elementos visuais presentes na tela.

Figura 20 - Desenvolvimento da tela "Histórico"

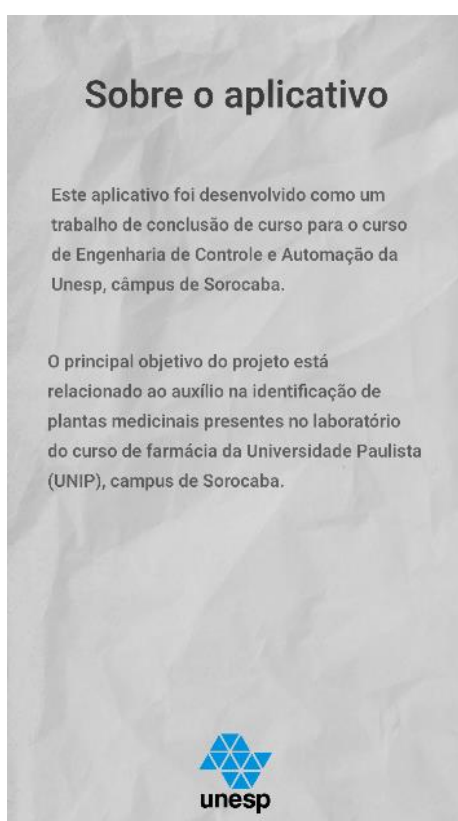


Fonte: Acervo pessoal

4.2.4 Desenvolvimento da tela “Sobre”

A tela “Sobre” do aplicativo tem como principal funcionalidade informar ao usuário sobre o propósito do desenvolvimento do aplicativo, autores e demais informações relevantes. Em termos técnicos, a tela possui somente elementos de textos e imagens para a exibição dos conteúdos. O conteúdo da tela pode ser visualizado na Figura 21.

Figura 21 - Tela "Sobre" do aplicativo



Fonte: Acervo pessoal

4.2.5 Desenvolvimento do código em Java

O desenvolvimento da lógica de programação para a execução das funcionalidades previstas do aplicativo deu-se início logo após o desenvolvimento das telas e botões. Nesta etapa, foram definidas as ações de cada botão, ou seja, qual a sequência de ações o *software* deve executar ao receber os comandos estabelecidos anteriormente: Escanear, Histórico e Sobre.

Como primeira etapa, foram definidos na atividade “*Main Activity*” dos três botões e realizadas as associações entre os elementos visuais criados da tela inicial e as variáveis.

Seguidamente, foram definidas as ações a serem executadas quando o usuário clicar no botão “Escanear”.

Como o processo de escaneamento de código QR, requisição das informações ao banco de dados, salvamento automático das informações e exibição na tela para o usuário envolve várias etapas e múltiplas ferramentas, foi criada uma atividade, denominada *qrscanner*, com o objetivo de realizar as ações.

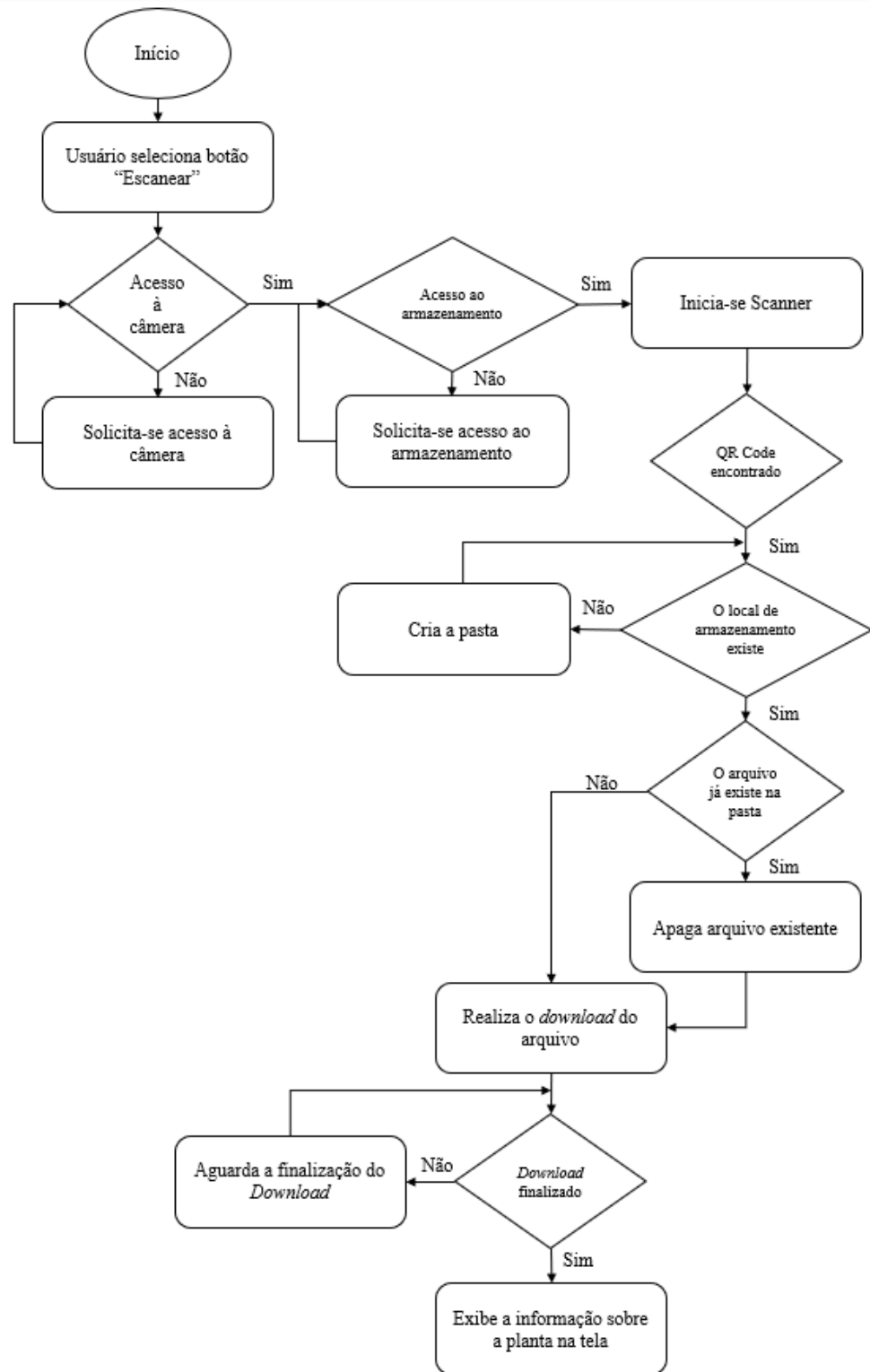
Desta forma, na seção “*Main Activity*”, foi inserido o evento de chamada da atividade *qrscanner*. A atividade contempla as etapas de solicitação de permissão de acesso à câmera para a leitura do código, solicitação de acesso ao armazenamento interno do dispositivo para salvar as informações das consultas e também, se refere à uma API responsável pela interpretação do código QR.

A primeira etapa, que ocorre sem notificação ao usuário, refere-se a implementação da API, responsável por realizar a interpretação dos dados do código QR, definição de parâmetros e conexão do aplicativo com a base de dados do Firebase.

Em seguida, a lógica de programação realiza a solicitação ao usuário para o acesso à câmera que fará o processo de leitura do *QR Code*, e o acesso ao armazenamento do dispositivo, para salvar as informações lidas. Esta ação é deve ser realizada uma vez pelo usuário, tendo em vista que o sistema operacional Android armazena as permissões concedidas, não sendo necessárias autorizações recorrentes durante o uso da aplicação.

Após o atendimento às solicitações de acesso, é realizada a leitura do código QR e as informações obtidas são armazenadas em uma variável de texto do tipo *String*, denominada “Data”. Após o armazenamento das informações na variável, segue-se para a próxima etapa: O *download* das informações obtidas para posterior acesso e a exibição das informações na tela do usuário. A sequência de instruções para o escaneamento da *tag* pode ser visualizada no fluxograma da Figura 22.

Figura 22 - Fluxograma contendo as instruções executadas para o escaneamento do código QR e a exibição de informações na tela



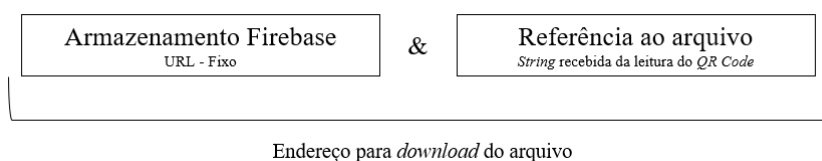
Fonte: Autoria própria

Após o escaneamento das informações contidas na *tag*, o código da aplicação executa uma instrução denominada *View*, que tem como objetivo a realização de duas ações após receber a informação lida no *QR Code*: Armazenar as informações no dispositivo para consultas futuras e exibir as informações na tela do usuário.

- Armazenamento

Utilizando o endereço do espaço de armazenamento do Firebase concatenado às informações lidas no *QR Code*, como apresentado na Figura 23, que se referem ao tipo de planta escaneado, é utilizada uma função denominada *DownloadManager*, nativa do IDE para realizar o *download* do arquivo e salvá-lo localmente.

Figura 23 - Diagrama esquemático representando a composição do endereço para *download* do arquivo



Fonte: Autoria própria

Para um melhor gerenciamento dos arquivos baixados para visualização posterior, foi definido um local de armazenamento em uma subpasta na pasta *Downloads* do dispositivo, sendo este local padrão do sistema operacional Android e de fácil acesso através do gerenciador de arquivos nativo do sistema operacional. A subpasta possui um nome padrão “app_farmlabscan”.

De forma a melhorar o gerenciamento do armazenamento interno do dispositivo e evitar o uso desnecessário de espaço de armazenamento, o algoritmo verifica antes de realizar o *download* das informações se já existe um arquivo de mesmo nome na pasta. Caso exista, o *script* apaga a informação anterior e realiza o armazenamento da nova versão do arquivo. Desta forma, não ocorre o evento de armazenamento de informações repetidas e, como a nova informação substitui a antiga, em caso de atualização

das informações no banco de dados será mantida no dispositivo sempre a versão mais recente.

As etapas de exibição e armazenamento de informações são dependentes de conexão com a internet, já que é necessário a consulta ao banco de dados armazenado em nuvem.

Todo o processo da etapa “Armazenamento” é realizado em segundo plano, ou seja, ocorre de forma totalmente automática sem que haja qualquer tipo de interação com o usuário, como a exibição de informações na tela ou solicitação de ações por parte do usuário.

O código para exibição da função *View* contendo as etapas de visualização e armazenamento do arquivo pode ser consultado no Apêndice A.

- Exibição das informações na tela para o usuário

É realizada por meio de uma ferramenta de leitura de PDF de código aberto, que recebe o endereço de onde o arquivo correspondente ao código escaneado foi armazenado no dispositivo e realiza o carregamento das informações na tela para o usuário.

De forma simplificada, o leitor de PDF integrado pode ser equiparado a uma aplicação para leitura de PDF comum, disponível para *download* e instalação nas lojas de aplicativo do dispositivo, porém esta está embarcada em uma aplicação e devidamente configurado para atender aos requisitos de projeto.

A fim de possibilitar uma melhor visualização e acessibilidade das informações, foram também definidos alguns parâmetros para a exibição do arquivo como, por exemplo, a possibilidade de o usuário aumentar o tamanho do conteúdo exibido na tela (*Zoom*), bem como da rolagem da página para a visualização de todas as informações fornecidas pela consulta.

Com a etapa inicial de leitura do código e retorno das informações finalizadas, logo depois foi dado o início do desenvolvimento da página “Histórico” do aplicativo,

responsável por exibir as consultas já realizadas e permitir ao usuário visualizar as informações quando desejar, sem conexão com a internet.

Primeiramente, para a exibição das informações armazenadas no dispositivo, é necessária a autorização para a leitura de informações do armazenamento interno do dispositivo, sendo imprescindível a inserção das etapas de solicitação de permissão no código. Assim como as permissões anteriormente solicitadas, a resposta do usuário é armazenada pelo sistema, não sendo preciso conceder permissão mais de uma vez.

As informações serão exibidas ao usuário em um elemento visual nativo denominado *ListView*, que deve receber as informações em um arranjo de dados (*Array*).

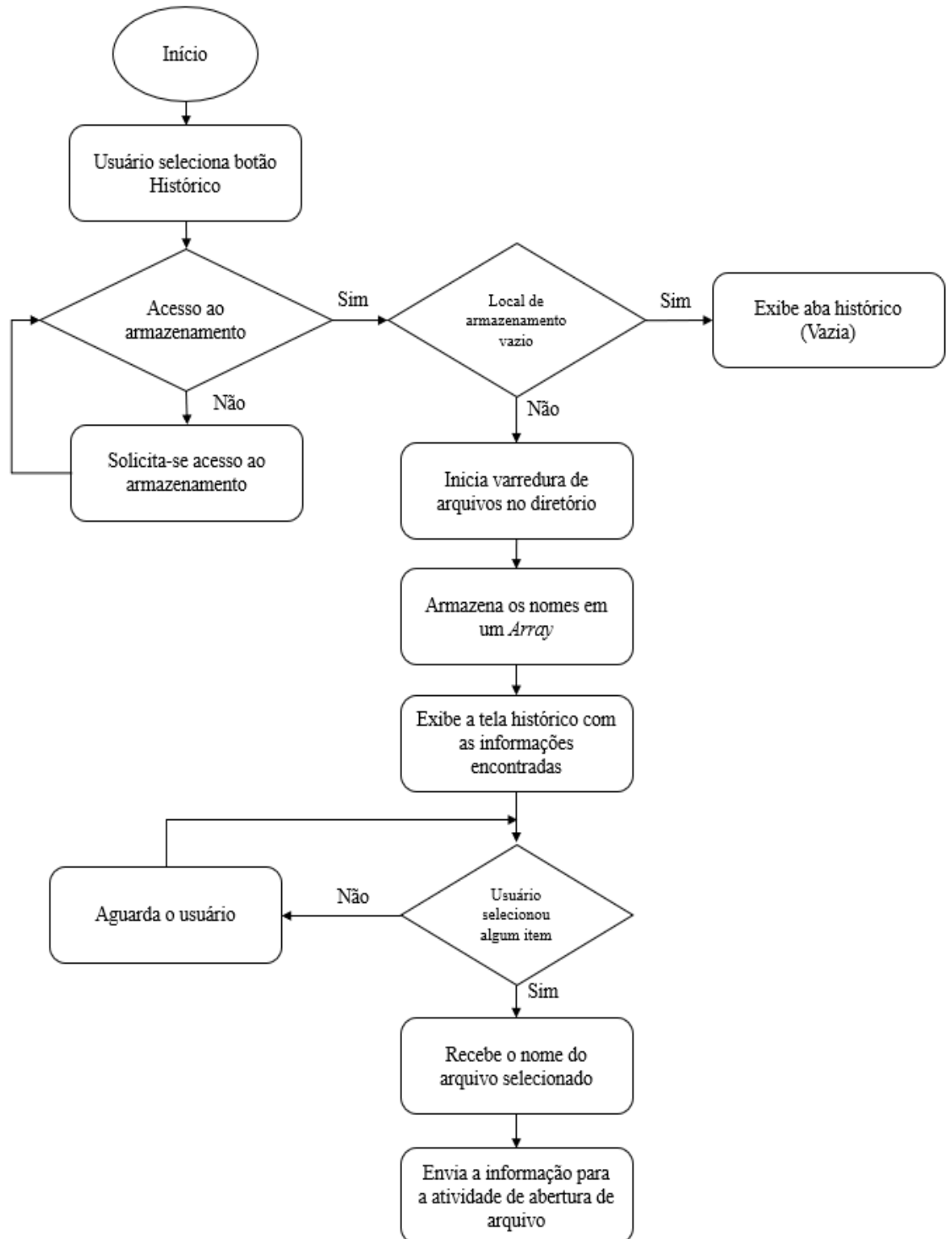
Ao carregar a página, o algoritmo realiza a tarefa de leitura das informações contidas na pasta “Download/app_farmlabscan”, local onde os arquivos foram armazenados após a consulta ao banco de dados durante o processo de leitura do *QR Code*. Durante o processo, o código armazena a lista de arquivos encontrados na pasta em um *array*, a fim de ser exibido em forma de lista na tela do usuário por meio do elemento *ListView*.

Após a criação da lista, é necessário definir as ações para, no evento de clique do usuário em algum dos elementos da lista, o arquivo seja carregado e as informações referentes à planta anteriormente consultada sejam exibidas na tela.

Para a realização desta ação, foi definida uma nova atividade no programa para a exibição do arquivo, denominada “*offlineviewfile*”. Essa atividade é executada quando o usuário clica em algum dos elementos da lista sendo enviados como parâmetros o nome do arquivo a ser visualizado.

De forma resumida, a lógica de instruções utilizada para o carregamento da aba “Histórico”, verificação dos arquivos já consultados, exibição da lista na tela e, caso o usuário selecione alguma opção, o carregamento do arquivo na tela pode ser observado no fluxograma da Figura 24.

Figura 24 - Sequência de instruções (simplificada) durante o processo de visualização das últimas consultas realizadas



Fonte: Autoria própria

A atividade *offlineviewfile* chamada na etapa anterior tem como função principal a abertura do arquivo em PDF utilizando a mesma ferramenta de visualização de PDF embarcada utilizada pela função *View*, executada quando o usuário seleciona o botão escanear. Neste caso, a ferramenta de leitura de PDF embarcada também é chamada pela função *offlineviewfile* quando o usuário seleciona algum item na lista da aba histórico.

4.2.6 Definição de parâmetros gerais e permissões

No desenvolvimento de um aplicativo para o sistema operacional Android, é necessária a definição de parâmetros iniciais da aplicação durante seu desenvolvimento como, por exemplo, a listagem de todas as permissões a serem utilizadas durante o uso do aplicativo bem como a declaração de eventuais provedores e componentes externos. A inserção destas informações ocorre no manifesto do projeto, em um arquivo denominado *Android_Manifest.xml*.

Considerando todas as implementações realizadas durante o projeto, foi inserida a listagem de permissões solicitadas para o uso do aplicativo: 1 – Acesso à câmera; 2 – Acesso à internet; 3 – Escrita no armazenamento; 4 – Leitura de dados do armazenamento e 5 – Permissão para realizar *downloads* sem a notificação do usuário.

As declarações do arquivo podem ser visualizadas no Apêndice B.

5. RESULTADOS

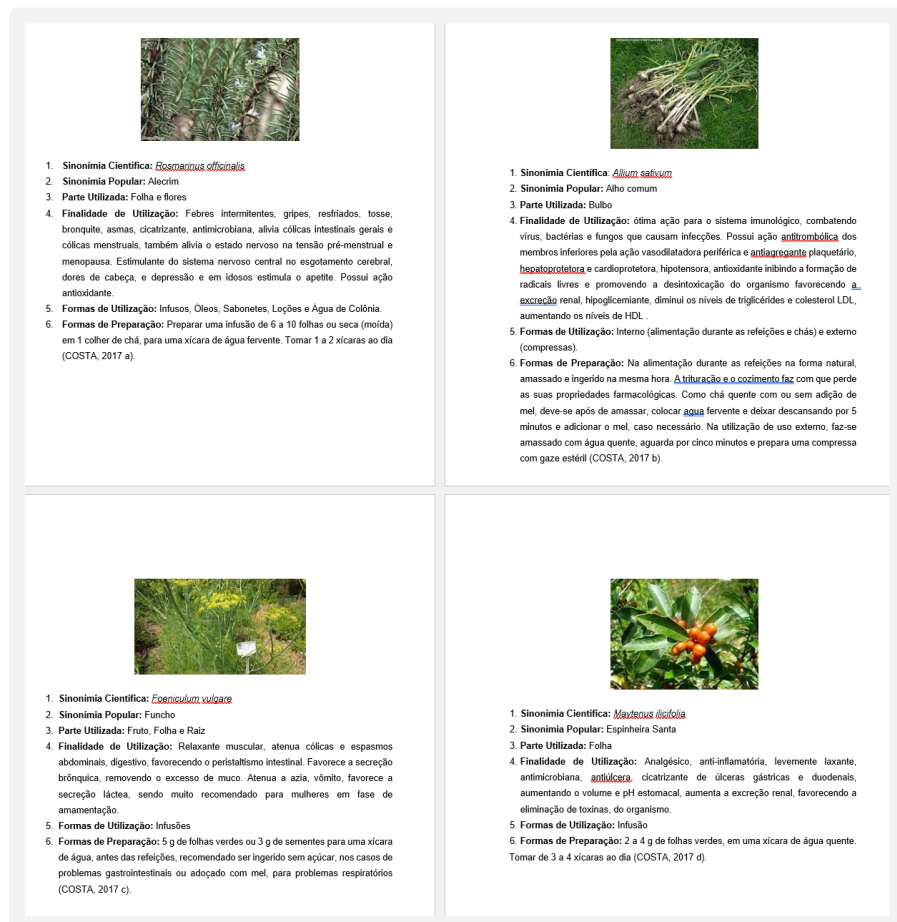
Após o desenvolvimento e implementação do projeto, foram analisados os resultados a fim de verificar se o funcionamento da aplicação corresponde ao esperado.

5.1 Inserção das informações no banco de dados

As informações referentes a 18 plantas medicinais presentes no laboratório do curso de farmácia da UNIP Sorocaba foram enviadas pela coorientadora do trabalho, Profª Patrícia Moriguchi para serem “carregadas” no banco de dados.

Inicialmente, foi criado um documento em PDF para cada uma das 18 plantas a serem inseridas no banco de dados contendo informações como: foto da planta, nome científico, nome popular, usos mais comuns, dentre outras informações relevantes. O modelo de criação do arquivo pode ser observado na Figura 25.

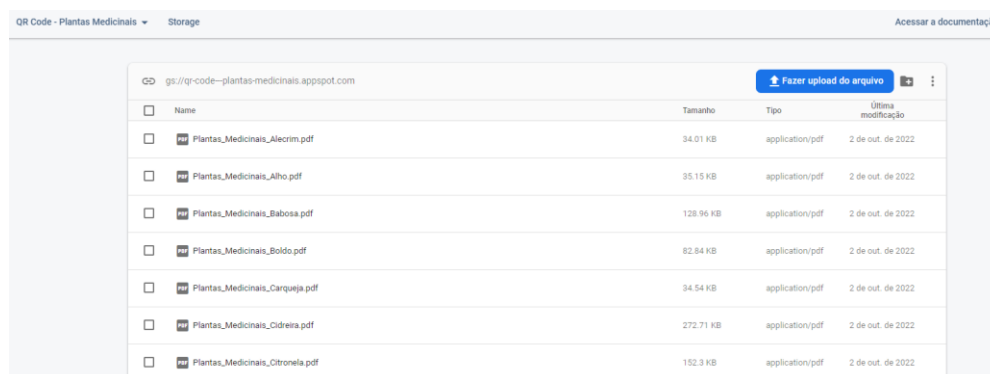
Figura 25 - Exemplo da criação dos arquivos em PDF para ser “carregado” no banco de dados.



Fonte: Autoria própria

Após a criação dos 18 arquivos em formato PDF, as informações foram inseridas no banco de dados do Firebase, como pode ser observado na Figura 26.

Figura 26 - Informações das plantas armazenadas no banco de dados.



The screenshot shows the Google Cloud Storage interface for a bucket named 'QR Code - Plantas Mediciniais'. It displays a list of eight PDF files, each with a checkbox, name, size, type, and last modification date. The files are: 'Plantas_Mediciniais_Alecrim.pdf' (34.01 KB), 'Plantas_Mediciniais_Alho.pdf' (95.15 KB), 'Plantas_Mediciniais_Babosa.pdf' (128.96 KB), 'Plantas_Mediciniais_Boldo.pdf' (82.84 KB), 'Plantas_Mediciniais_Carqueja.pdf' (94.54 KB), 'Plantas_Mediciniais_Cidreira.pdf' (272.71 KB), and 'Plantas_Mediciniais_Citronela.pdf' (152.3 KB). All files were last modified on 2 de out. de 2022.

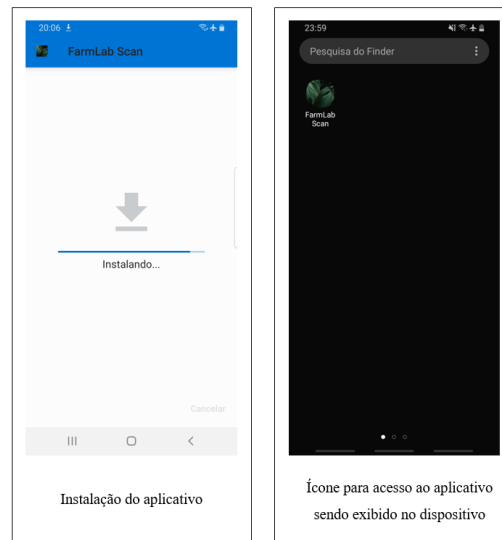
<input type="checkbox"/>	Name	Tamanho	Tipo	Última modificação
<input type="checkbox"/>	Plantas_Mediciniais_Alecrim.pdf	34.01 KB	application/pdf	2 de out. de 2022
<input type="checkbox"/>	Plantas_Mediciniais_Alho.pdf	95.15 KB	application/pdf	2 de out. de 2022
<input type="checkbox"/>	Plantas_Mediciniais_Babosa.pdf	128.96 KB	application/pdf	2 de out. de 2022
<input type="checkbox"/>	Plantas_Mediciniais_Boldo.pdf	82.84 KB	application/pdf	2 de out. de 2022
<input type="checkbox"/>	Plantas_Mediciniais_Carqueja.pdf	94.54 KB	application/pdf	2 de out. de 2022
<input type="checkbox"/>	Plantas_Mediciniais_Cidreira.pdf	272.71 KB	application/pdf	2 de out. de 2022
<input type="checkbox"/>	Plantas_Mediciniais_Citronela.pdf	152.3 KB	application/pdf	2 de out. de 2022

Fonte: Autoria própria

Após essa etapa, foi gerado o *link* de acesso para cada um dos 18 arquivos e, em seguida, gerados 18 *QR Codes*, 1 para cada planta que será utilizado como *tag* de identificação da planta. Todos os códigos *QR* referentes aos arquivos estão presentes no apêndice C.

5.2 Instalação no *smartphone* e utilização

Utilizando a plataforma de desenvolvimento Android Studio, foi produzido o arquivo de instalação (*apk*) que foi armazenado em um diretório de armazenamento em nuvem, sendo o *link* de acesso e *download* gerado, bem como a criação de um *QR Code* a partir do *link* para possibilitar a instalação do aplicativo de forma simplificada. O processo de instalação do aplicativo pode ser visualizado na Figura 27. O *QR Code* para *download* da aplicação está presente no apêndice C.

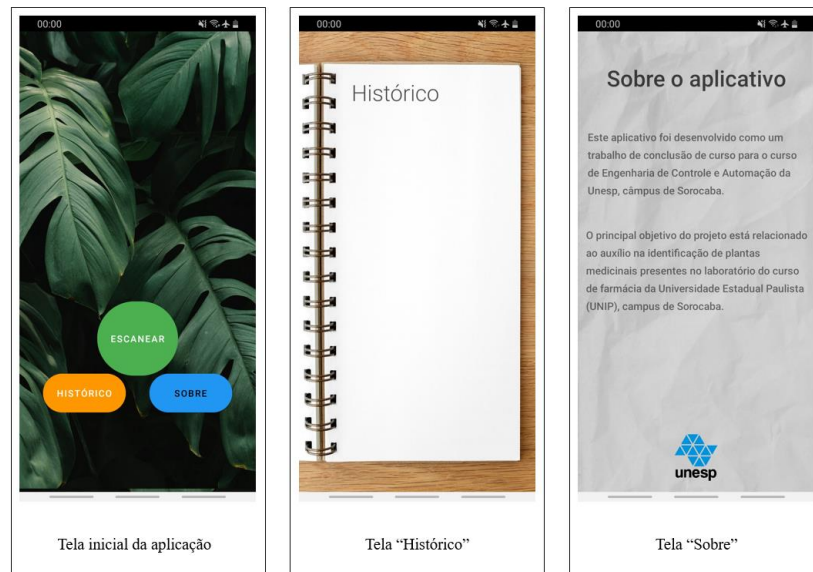
Figura 27 - Captura de tela do processo de instalação do aplicativo

Fonte: Autoria própria

A instalação foi realizada em um *smartphone* com sistema operacional Android 9 (Samsung Galaxy S8) e foram realizados vários testes de funcionalidade da aplicação.

No primeiro teste, foi aberta a página inicial da aplicação e em seguida, selecionada a opção “Histórico”, a fim de verificar a funcionalidade da aba. Como logo após a instalação nenhuma informação previamente consultada havia sido armazenada, a aba “Histórico” foi exibida sem informações das últimas plantas consultadas. Na sequência, foi selecionada a opção “Sobre” e verificou-se que o funcionamento correspondeu ao esperado. Os passos descritos podem ser observados na Figura 28.

Figura 28 - Telas da aplicação após a instalação.



Fonte: Autoria própria

Logo após, foi pressionado o botão “Escanear” e, com a câmera do dispositivo “aberta”, ela foi apontada para o código QR. Em seguida, após a identificação do código, as informações consultadas foram mostradas na tela. As etapas para o escaneamento do código podem ser observadas na Figura 29.

Figura 29 - Capturas de tela do processo de escaneamento da *tag*.



Fonte: Autoria própria

Na próxima etapa, foi realizado o retorno à tela inicial através do botão “Voltar”, nativo do sistema Android e presente em todas as telas da aplicação.

Após os testes iniciais, foi realizada novamente a consulta da tela “Histórico” com a finalidade de verificar se as informações consultadas estavam disponíveis para a visualização. Conforme exibido na Figura 30, observa-se que a consulta realizada está disponível para visualização posterior com apenas um toque.

Figura 30 – Captura de tela da consulta realizada utilizando a aplicação



Fonte: Acervo pessoal

5.3 Tratamento de erros

Ao longo do uso da aplicação por diversos usuários, é possível que algumas condições adversas ocorram, como por exemplo: conexão com a internet instável, conexão com a internet indisponível, código QR não pertencente a planta escaneado ou informações não encontradas no banco de dados. Diante da possibilidade de ocorrência dessas condições, foi adicionado ao aplicativo um tratamento de erro que informa ao usuário caso alguma das falhas mencionadas anteriormente ocorra.

A tela contendo a mensagem de erro presente na Figura 31, pode vir a ser exibida durante o uso da opção “Escanear” da aplicação caso alguma condição fora do controle ocorra. Para sair da tela é necessário pressionar o botão “voltar”, que leva o usuário até a tela inicial.

Figura 31 - Mensagem de erro

Fonte: Autoria própria.

5.4 Teste do Aplicativo

Após a finalização do desenvolvimento e implementação do aplicativo, bem como de testes de funcionalidade, o *link* para *download* e instalação do aplicativo foi enviado para outras pessoas a fim de que elas pudessem verificar e testar o funcionamento em dispositivos Android de diferentes fabricantes bem como garantir que a usabilidade da aplicação atenda às necessidades do público geral.

Desta forma, o aplicativo foi testado por 5 pessoas, com perfis descritos a seguir, que relataram as seguintes experiências durante a utilização da aplicação:

1 – Pessoa do meio acadêmico, adulta e com o hábito de utilização de ferramentas tecnológicas no dia a dia: relatou facilidade para instalação e utilização do aplicativo. O usuário realizou testes escaneando os códigos das plantas presentes no Apêndice C e relatou sucesso na execução das consultas.

2 – Duas pessoas de fora do meio acadêmico, adultos, sem o hábito de utilização de ferramentas tecnológicas no dia a dia: eles destacaram a facilidade de instalação, sendo necessária a alteração de uma permissão de instalação de aplicação desconhecida em um dos dispositivos testados, mas o usuário conseguiu ajustar a configuração e prosseguir com a instalação por conta própria. Ambos relataram que a interface é de fácil utilização,

com as funcionalidades claramente descritas nos botões. Foi possível executar as consultas dos códigos presentes no Apêndice C com facilidade.

3 – Duas pessoas, estudantes da Universidade Paulista e alunos do curso de Farmácia: ambos relataram facilidade no processo de instalação e simplicidade na utilização. Um deles ainda ressaltou, em decorrência da facilidade de uso, a grande acessibilidade do aplicativo para diversos perfis de usuários, especialmente os que possuem mais dificuldades com relação ao manuseio de *smartphones* e aplicativos móveis.

5.5 Vídeo da Utilização do Aplicativo

Para demonstrar a utilização de todas as funcionalidades do aplicativo desenvolvido, foi feito um vídeo explicando o passo a passo de como utilizar o aplicativo, com todos os cenários possíveis.

Inicia-se o vídeo com o processo de instalação do aplicativo e logo após, são escaneados 3 códigos referentes às plantas medicinais. Por fim, é escaneado um código inválido a fim de demonstrar a mensagem de erro.

Após os escaneamentos, é realizada a demonstração das funcionalidades da aba “Histórico”, com a possibilidade de consultas das plantas previamente cadastradas sem a necessidade de conexão com a internet.

O vídeo de demonstração pode ser visualizado através do *link*: https://bit.ly/app_unesp_demonstracao

6. CONCLUSÕES

Durante a elaboração desse trabalho, foi possível observar a grande relevância que o uso de ferramentas digitais para *smartphones* possui, sendo o desenvolvimento de ferramentas para auxiliar o ensino uma das vertentes dentro do universo da digitalização. A implementação do aplicativo mostrou a grande gama de possibilidades para o desenvolvimento de ferramentas para facilitar o aprendizado através da utilização de dispositivos móveis e, também, o potencial de associação desses com outras tecnologias como, por exemplo, armazenamento em nuvem e uso de códigos de identificação.

Após o desenvolvimento do aplicativo, ele foi testado por cinco pessoas, sendo o perfil delas tanto do meio acadêmico quanto externas a ele. Elas também não atuam na área de desenvolvimento de *softwares*. Dentre os principais relatos, destaca-se a facilidade de instalação do aplicativo, bem como a simplicidade na utilização das suas principais funções, como o escaneamento da *tag* e a visualização do histórico. Os testes mostraram que o aplicativo possui acessibilidade simples para diversos públicos.

Conclui-se que todas as funcionalidades necessárias para uso de *tags* em plantas medicinais foram inseridas no aplicativo e que, desta forma, o aplicativo pode ser um facilitador para o aprendizado dos alunos do Curso de Graduação em Farmácia da Universidade Paulista (UNIP), Câmpus de Sorocaba e, posteriormente, para usuários de outras instituições.

Durante o desenvolvimento do projeto, percebeu-se que o envio das informações para o banco de dados necessitava de um certo nível de conhecimento técnico da plataforma para a utilização. De forma a tornar a solução mais escalável, possibilitando atingir um público maior, observa-se a possibilidade de projetos futuros objetivando a criação de uma funcionalidade para o envio de informações ao banco de dados de forma mais fácil. Isso possibilitará o carregamento de novas informações por usuários sem grandes conhecimentos em áreas relacionadas ao desenvolvimento de aplicações e banco de dados.

REFERÊNCIAS

SOMMERVILLE, I. *Software Engineering*. 9ª Edição. Boston: Pearson, 2021.

TANENBAUM, A. S.; BOS, H. *Sistemas Operacionais Modernos*. 4ª edição. São Paulo: Pearson, 2016.

GERBER, A.; CRAIG, C. *Learn Android Studio: Build Android Apps Quickly and Effectively*. Nova Iorque: Apress, 2015

MORONEY, L. *The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform*. Seattle: Apress, 2017.

GRESCZYSCZYN, M. C. C.; FILHO, P. S. DE C.; MONTEIRO, E. L. Aplicativos Educacionais para *Smartphone* e sua Integração Com o Ensino de Química. **Revista de Ensino, Educação e Ciências Humanas**. v. 17, n. 5, p. 398-403. 2016

FERREIRA, T. V.; RIBEIRO, J. de S.; CLEOPHAS, M. das G. A ciência pelas lentes dos *smartphones*: o potencial do aplicativo *QR CODE* no ensino de Química. **Revista Thema**, [S. l.], v. 15, n. 4, p. 1217–1233, 2018.

LEITE, R. A. F.; BRITO, E. S.; SILVA, L. M. C.; PALHA, P. F.; VENTURA, C. A. A. Acesso à informação em saúde e cuidado integral: percepção de usuários de um serviço público. **Interface - Comunicação, Saúde, Educação**, v. 51, n. 18, p. 61-67, 2014.

NAKASHIMA, R.; BARROS, D. M. V.; AMARAL, S. F. O uso pedagógico da lousa digital associado a Teoria dos Estilos de Aprendizagem. **Revista de Estilos de Aprendizagem**, v. 4, n. 4, p. 1-12, 2009.

BARROS, A. F. O uso das tecnologias na educação como ferramentas de aprendizado. **Revista Científica Semana Acadêmica**, v. 1, n. 156, 2019.

SILVA, M. C.; SANTOS, M. T. P. Os Paradigmas de Desenvolvimento de Aplicativos para Aparelhos Celulares. **T.I.S. São Carlos**, v. 3, n. 2, p. 162-170, 2014.

ANDROID STUDIO. **Informações do Usuário**. *Android*. Disponível em: <<https://developer.android.com/studio/intro?hl=pt>>. Acesso em 05 ago. 2022.

GOOGLE DEVELOPERS. **Barcode Scanning**. Google. Disponível em: <<https://developers.google.com/ml-kit/vision/barcode-scanning>>. Acesso em 11 ago. 2022.

BHAGAT, H. R.; BAJAJ, K. ***The 18:9 display dilemma: Will the new smartphone screens make our lives easier or do the opposite?***. Economic Times. Disponível em: <<https://economictimes.indiatimes.com/articleshow/62662023.cms>> Acesso em 26 jul. 2022.

CHAN, S. ***Global App Spending Approached \$65 Billion in the First Half of 2021, Up More Than 24% Year-Over-Year***. Sensor Tower. Disponível em <<https://sensortower.com/blog/app-revenue-and-downloads-1h-2021>>. Acesso em 26 jul. 2022

FIREBASE. **Documentação**. Google. Disponível em: < <https://firebase.google.com>>. Acesso em 05 ago. 2021.

GREGERSEN, E. **QR Code**. Britannica. Disponível em <<https://www.britannica.com/technology/QR-Code>>. Acesso em 16 jul. 2022.

LAVADO, T. **Uso da internet no Brasil cresce, e 70% da população está conectada**. G1/Globo. Disponível em: <<https://g1.globo.com/economia/tecnologia/noticia/2019/08/28/uso-da-internet-no-brasil-cresce-e-70percent-da-populacao-esta-conectada.ghtml>> . Acesso em 11 ago. 2022.

APPSFLYER. **Brasil cresce 55% em número de instalações de apps, destacando-se na América Latina.** Proxima. Disponível em: <<https://www.proxima.com.br/home/proxima/noticias/2020/06/30/brasil-cresce-55-em-numero-de-instalacoes-de-apps-destacando-se-na-america-latina.html>>. Acesso em 11 ago. 2022.

JETBRAINS. **IntelliJ IDEA – Recursos.** JetBrains. Disponível em <<https://www.jetbrains.com/pt-br/idea/features/>>. Acesso em 11 ago. 2022.

ADOBE. **What is a PDF? Portable Document Format | Adobe Acrobat DC.** Adobe. Disponível em <<https://www.adobe.com/acrobat/about-adobe-pdf.html>>. Acesso em 11 ago. 2022

BITLY. **About Bitly.** Bitly. Disponível em <<https://bitly.com/pages/about>>. Acesso em 11 ago. 2022

GLOSSÁRIO

Linguagem de programação. É uma ferramenta utilizada para a instrução de tarefas a serem realizadas por computador. Definido por um conjunto de regras sintáticas e semânticas que possibilitam a interpretação por parte do computador dos comandos realizados pelo programador. Através dessa ferramenta, o programador pode especificar exatamente quais tarefas o computador deve realizar, em que momento elas devem ser realizadas, o modo como são executadas, o local das informações a serem lidas ou armazenadas dentre outros parâmetros que podem ser definidos durante a construção do Código.

Programador. Profissional com conhecimentos em linguagens de desenvolvimento *Softwares* com atuação focada no desenvolvimento de novas soluções, melhorias em *softwares* já existentes e correção de problemas.

Software. É um conjunto de instruções a serem seguidas por um computador com a finalidade de executar uma determinada ação ou um conjunto de ações como a manipulação de dados, codificação ou decodificação de um arquivo, alterações em dados já existentes e criação de recursos visuais. Geralmente requer interação do usuário através dos dispositivos de interação (ex. mouse, tela *touchscreen*) e pode apresentar resposta visual e/ou sonora para informar ao usuário informações relevantes sobre a execução da tarefa.

Aplicativo. Programa de computador com o objetivo de realizar tarefas pré-determinadas (ex. envio de mensagens, edição de fotos) facilitando e reduzindo o tempo de execução de uma tarefa pelo usuário.

Link. Referência que se encontra numa página da Internet, para um outro documento ou site, ou para outra área dentro da mesma página ou site.

Tag. Código ou sequência de caracteres que identifica dados ou fornece informações ou especificações numa parte de texto (ex.: *tags* de formatação).

Script. Conjunto de instruções em código, ou seja, escritas em linguagem de computador. É uma linguagem de programação que executa diversas funções no interior de um programa de computador. As linguagens de script são ferramentas utilizadas para controle de um determinado programa ou aplicativo; para configuração ou instalação em sistemas operacionais.

Upload. Upload é um termo da língua inglesa com significado referente à ação de enviar dados de um computador local para um computador ou servidor remoto, geralmente através da internet.

Token. É um tipo de chave eletrônica que contém um código que pode ser utilizado para acessar ou decodificar determinadas informações ou arquivos de computador.

Background. Termo em inglês utilizado, comumente no meio de tecnologia, para se referir a ações ou tarefas executadas em segundo plano, ou seja, sem a necessidade de ação do usuário para o início, continuação ou término da ação.

Zoom. Palavra em inglês para a definição do efeito de afastamento ou aproximação visual de um objeto, podendo ser realizado digitalmente ou através de lentes, como em câmeras fotográficas.

APÊNDICE A

```
[...]
// Função para realizar o download do arquivo através da URL fornecida
// pelo QRcode e, seguidamente, exibir as informações na tela
public class view extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // Inicialização do código
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view);
        // Definição de variáveis
        String firebaseappstorage =
"https://firebasestorage.googleapis.com/v0/b/qr-code---plantas-
medicinais.appspot.com/o/";
        String googleview =
"https://drive.google.com/viewerng/viewer?embedded=true&url="; // URL de
ferramenta do Google para visualização de PDF
        String url = firebaseappstorage +
getIntent().getStringExtra("url"); // URL escaneada que remete às
informações para serem consultadas no Firebase
        // Cria pasta no diretório para salvar arquivos baixados
        File f = new
File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DO
WNLOADS), "app_farmlabscan");
        if (!f.exists()) {
            f.mkdirs();
        }
        // Configuração do Download Manager para realizar o download do
arquivo
        DownloadManager.Request request = new
DownloadManager.Request(Uri.parse(url));
        String filetype = URLUtil.guessFileName(url, null, null);
        request.setTitle(filetype);

request.setNotificationVisibility(DownloadManager.Request.VISIBILITY_HIDDEN
);
request.setDestinationInExternalPublicDir("/Download/app_farmlabscan",
filetype);
        // Verifica se o arquivo já existe no local
        File oldV = new File(f, filetype);
        if(oldV.exists()) {
            boolean delete = oldV.delete();
        }
        DownloadManager downloadManager =
(DownloadManager) getSystemService(DOWNLOAD_SERVICE);
        downloadManager.enqueue(request);
        // Exibe visualização
        WebView webView = findViewById(R.id.web);
        webView.setWebViewClient(new WebViewClient());
        webView.getSettings().setJavaScriptEnabled(true);
        webView.setVerticalScrollBarEnabled(true);
        webView.setHorizontalScrollBarEnabled(true);
        webView.loadUrl(googleview + url);
    }
    @Override
    public void onBackPressed() {
        Intent intent= new Intent(getApplicationContext(),
MainActivity.class);
        startActivity(intent);
    }
} [...]

```

APÉNDICE B

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.testeappl">
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission
android:name="android.permission.DOWNLOAD_WITHOUT_NOTIFICATION" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.TesteApp1" >
        <activity
            android:name=".offlineviewfile"
            android:exported="false" />
        <activity
            android:name=".view"
            android:exported="false" />
        <activity
            android:name=".qrscanner"
            android:exported="false" />
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER"
/>
                </intent-filter>
            </activity>
        <activity
            android:name=".about"
            android:exported="false" />
        <activity
            android:name=".history"
            android:exported="false" />
        <provider
            android:name="androidx.core.content.FileProvider"
            android:authorities="${applicationId}.provider"
            android:exported="false"
            android:grantUriPermissions="true">
            <meta-data
                android:name="android.support.FILE_PROVIDER_PATHS"
                android:resource="@xml/provider_paths" />
            </provider>
        </application>
</manifest>

```

APÊNDICE C

Download do aplicativo*QR Codes plantas medicinais*

Alecrim



Alho



Babosa



Boldo



Carqueja



Cidreira



Citronela



Cúrcuma



Espinheira Santa



Funcho



Gengibre



Guaco



Hortelã



Lavandula



Manjeriçao



Melissa



Tomilho



Erva Preta