

---

**Mani: aplicativo gerenciador de finanas  
pessoais utilizando React Native**

*Laís Lopes Resende*

---

PROJETO DE CONCLUSÃO DE CURSO

Data de Depósito: 29/11/2021

Assinatura: \_\_\_\_\_

# **Mani: aplicativo gerenciador de finanças pessoais utilizando React Native**

*Laís Lopes Resende*

*Valter Ribeiro Lima Júnior*

Monografia apresentada ao Centro Universitário de Formiga – UNIFOR-MG, como requisito parcial para obtenção do título de bacharel em Ciência da Computação, sob a orientação do Prof<sup>o</sup>. Valter Ribeiro Lima Júnior

UNIFOR-MG – Formiga

2021

*À minha família, amigos,  
professores e todas as pes-  
soas que, de alguma forma,  
contribuíram para que eu  
chegasse até aqui.*

# Agradecimentos

---

Neste momento único de alegria e sensação de dever cumprido, sou grata por todo o conhecimento adquirido ao longo de 4 anos de dedicação e estudo.

Dedico essa vitória à minha mãe Ermínia, meu padrasto Eduardo e minhas irmãs Bruna e Carol. É um presente ter uma família que deposita confiança no meu progresso e é um verdadeiro alicerce para a construção de todos os meus sonhos. Tudo que sou devo a cada um, pois são as minhas maiores referências de força, caráter, honestidade e amor.

Ao meu pai Orlando (*in memoriam*), que não está mais entre nós, mas é sempre minha melhor lembrança. Sua memória me inspira e me faz persistir.

Aos meus colegas e amigos, pela parceria, pelos momentos e pelo incentivo para chegar até aqui. Em especial àqueles que se tornaram uma das melhores partes da minha vida, Ana Paula e Bruno, espero levar vocês pra sempre comigo.

A todo o corpo docente, gratidão pelos ensinamentos técnicos e por aqueles que vou levar para a vida. Em especial ao meu orientador Valter, suas contribuições foram valiosas, inspiradoras e fizeram toda a diferença para a conclusão deste trabalho.

Por fim, agradeço a todos que, direta ou indiretamente, fizeram parte da minha formação. Serei sempre grata a vocês.

*”Talvez não tenha conseguido fazer o melhor,  
mas lutei para que o melhor fosse feito. Não  
sou o que deveria ser, mas Graças a Deus,  
não sou o que era antes”*

*Marthin Luther King*

# Sumário

---

---

<b>Lista de Siglas</b>	<b>iii</b>
<b>Lista de Figuras</b>	<b>iv</b>
<b>Lista de Tabelas</b>	<b>v</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Objetivos do Trabalho . . . . .	2
1.2 Justificativa . . . . .	2
1.3 Estrutura da Monografia . . . . .	2
<b>2 Referencial Teórico</b>	<b>3</b>
2.1 Aplicações para dispositivos móveis . . . . .	3
2.1.1 Desenvolvimento multiplataforma . . . . .	4
2.1.2 React Native . . . . .	4
2.1.3 Expo . . . . .	4
2.1.4 JavaScript . . . . .	5
2.2 Banco de dados . . . . .	5
2.2.1 Banco de dados não relacional . . . . .	5
2.2.2 JSON . . . . .	6
2.2.3 Firebase . . . . .	6
2.3 Visão computacional . . . . .	7
2.4 Considerações Finais . . . . .	7
<b>3 Estado da Arte</b>	<b>8</b>
3.1 Organize . . . . .	8
3.2 Mobills . . . . .	8
3.3 Fortuno . . . . .	9
3.4 Considerações Finais . . . . .	10

<b>4</b>	<b>Metodologia</b>	<b>11</b>
4.1	Projeto da aplicação . . . . .	11
4.2	Planejamento . . . . .	11
4.2.1	Levantamento de requisitos . . . . .	12
4.2.2	Casos de uso . . . . .	12
4.2.3	Prototipação . . . . .	16
4.2.4	Banco de dados . . . . .	16
4.3	Considerações Finais . . . . .	17
<b>5</b>	<b>Resultados</b>	<b>19</b>
5.1	Apresentação do sistema . . . . .	19
5.1.1	Tela de boas vindas . . . . .	19
5.1.2	Cadastro . . . . .	19
5.1.3	Login . . . . .	19
5.1.4	Tela principal . . . . .	20
5.1.5	Categorias . . . . .	20
5.1.6	Notícias . . . . .	20
5.1.7	Inserção de receitas . . . . .	22
5.1.8	Inserção de despesas . . . . .	22
5.1.9	Perfil . . . . .	22
5.1.10	Relatórios diários . . . . .	22
5.1.11	Relatórios mensais . . . . .	24
5.2	Avaliações . . . . .	24
<b>6</b>	<b>Conclusões</b>	<b>26</b>
6.1	Trabalhos futuros . . . . .	27
6.1.1	Leitura de código de barra . . . . .	27
6.1.2	Login através de biometria . . . . .	27
6.1.3	Sincronização . . . . .	27
6.1.4	Transações recorrentes . . . . .	27

# Lista de Siglas

---

API Application Programming Interface - Interface de Programação de Aplicativos

HTML HyperText Markup Language - Linguagem de Marcação de Hipertexto

IBGE Instituto Brasileiro de Geografia e Estatística

JS JavaScript

JSON JavaScript Object Notation - Notação de Objetos JavaScript

JSX JavaScript XML

PIB Produto Interno Bruto

SDK Software Development Kit - Kit de Desenvolvimento de Software

SQL Structured Query Language - Linguagem de Consulta Estruturada

UML Unified Modeling Language - Linguagem Unificada de Modelagem

XML Extensible Markup Language - Linguagem Extensível de Marcação



# Lista de Figuras

---

---

3.1	Estados da arte . . . . .	9
4.1	Diagrama de Casos de Uso - Mani . . . . .	13
4.2	Interfaces do aplicativo Mani . . . . .	16
4.3	Estrutura do banco de dados . . . . .	18
5.1	Mani - Telas de boas vindas, cadastro e login . . . . .	20
5.2	Mani - Variações da tela principal . . . . .	21
5.3	Mani - Categorias, nova categoria e registros de uma categoria . . . . .	21
5.4	Mani - Notícias . . . . .	22
5.5	Mani - Telas de inserção de receita e despesa . . . . .	23
5.6	Mani - Perfil . . . . .	23
5.7	Mani - Telas de relatórios diários e mensais . . . . .	24
5.8	Avaliações de usuários da Google Play . . . . .	25

# Lista de Tabelas

---

---

2.1	Participação no mercado de sistemas operacionais móveis no mundo. . . . .	3
3.1	Comparativo de funcionalidades . . . . .	10
4.1	UC001: Fazer cadastro . . . . .	13
4.2	UC002: Fazer login . . . . .	13
4.3	UC003: Inserir receita . . . . .	14
4.4	UC004: Inserir despesa - Manualmente . . . . .	14
4.5	UC005: Inserir despesa - Leitura de nota fiscal . . . . .	14
4.6	UC006: Consultar registros por categoria . . . . .	15
4.7	UC007: Consultar relatórios - Diários . . . . .	15
4.8	UC008: Consultar relatórios - Mensais . . . . .	15
4.9	UC009: Ler notícias . . . . .	15

# Resumo

---

RESENDE, L. L. *Mani: aplicativo gerenciador de finanças pessoais utilizando React Native*. Monografia (Graduação) — Centro Universitário de Formiga – UNIFOR-MG – Formiga, 2021.

Com o constante desenvolvimento da sociedade, os dispositivos móveis vêm ganhando cada vez mais espaço na vida das pessoas. À medida que o número de dispositivos no mercado cresce, aumenta também a procura por aplicativos móveis, que trazem soluções práticas para problemas enfrentados no dia a dia. Para tal, o propósito deste projeto é o desenvolvimento de um aplicativo multiplataforma focado em oferecer uma interface intuitiva e prática que auxilie o gerenciamento das finanças pessoais. O aplicativo foi implementado utilizando o *framework* React Native e os serviços de autenticação e banco de dados do Firebase. O aplicativo desenvolvido foi publicado na Play Store com o nome "Mani - Gerenciador de Finanças Pessoais" e encontra-se disponível para download <sup>1</sup>.

Palavras-chave: Gerenciador de Finanças, Aplicativo, Android, React Native.

---

<sup>1</sup><https://play.google.com/store/apps/details?id=com.laisresende.mani>

# Abstract

---

RESENDE, L. L. *Mani: personal finance manager application using React Native*. Monografia (Graduação) — Centro Universitário de Formiga – UNIFOR-MG – Formiga, 2021.

With the constant development of the society, mobile devices are gaining more space in people's lives. As the number of devices on the market grows, so does the demand for mobile apps, which bring practical solutions to everyday problems. For that, the purpose of this project is the development of a cross-platform application focused on offering an intuitive and practical interface that helps the management of personal finances. The application was implemented using the framework React Native and Firebase authentication and database services. The developed application was published on Play Store under the name "Mani - Personal Finance Manager" and is available for download <sup>1</sup>.

Keywords: Finance manager, Application, Android, React Native.

---

<sup>1</sup><https://play.google.com/store/apps/details?id=com.laisresende.mani>

---

# Introdução

---

A organização do orçamento é algo fundamental para se obter uma boa gestão financeira, mas também é uma das maiores dificuldades das pessoas na sociedade moderna. A falta de tempo para resolver problemas pessoais ou a falta de disciplina podem, ocasionalmente, gerar o esquecimento ou mesmo o desinteresse para essa organização. Porém, na última década, com a evolução tecnológica, a educação financeira teve grande avanço, pois passou a contar com a praticidade e agilidade de aplicações móveis.

Segundo Cerbasi (2015), a organização na vida financeira é essencial para que o indivíduo tenha maior controle sobre seu dinheiro, maior consciência sobre suas escolhas e maior eficiência no uso de sua renda. Uma pesquisa realizada pelo SPC Brasil mostrou que o número de brasileiros que faziam controle financeiro cresceu. Em 2018, passou de 55% para 63%, mas dessas pessoas, apenas 10% utilizavam aplicativos móveis (SPCBRASIL, 2019).

De acordo com Liftoff (2021), as instalações de aplicativos financeiros tiveram um ótimo desempenho de 2020, com um aumento de 15% em comparação a 2019. Além disso, passaram 16,3 bilhões de horas utilizando os aplicativos, 45% a mais do que no ano anterior. Somente no Brasil, houve um crescimento de 35% de horas gastas em aplicativos financeiros durante a primeira semana de pandemia, em março de 2020.

No entanto, é possível perceber que os aplicativos, de modo geral, não são capazes de interferir no padrão de evasão e fornecer educação financeira. A rotina de gerenciamento de despesas é muitas vezes trabalhosa e burocrática, e as formas de visualização das movimentações são, geralmente, pouco amigáveis, desestimulando o usuário.

Diante dessas informações, somos levados ao seguinte questionamento: Como fornecer uma experiência simples e intuitiva para o registro de ganhos e despesas?

## 1.1 Objetivos do Trabalho

Para tanto, o presente trabalho tem como objetivo o desenvolvimento de uma aplicação multiplataforma focado em oferecer uma interface intuitiva e prática que auxilie o gerenciamento das finanças pessoais. A fim de atingir satisfatoriamente o objetivo geral defendido, foram projetados alguns objetivos específicos:

- Por meio de funcionalidades assertivas, interface limpa e simples, registro de despesas através da leitura de QR code de notas fiscais, o aplicativo tem o intuito de manter boa usabilidade e praticidade;
- O aplicativo conta também com o propósito de trazer conhecimento e gerar interesse dos usuários, o que se revela na aba de notícias. Essa funcionalidade disponibiliza ao usuário informações sobre a área financeira, uma maneira de captar sua atenção e diminuir o índice de evasão do aplicativo;
- O Firebase foi utilizado com o intuito de fornecer uma autenticação segura, além de oferecer o banco de dados em tempo real que facilita na visualização dos dados de forma rápida.

## 1.2 Justificativa

Com a pandemia de Covid-19, o PIB do Brasil teve queda de 4,1% em 2020, segundo dados divulgados pelo IBGE, a maior contração em 24 anos (JIMÉNEZ, 2021). Neste cenário de crise, mais do que nunca, as pessoas estão buscando aplicativos para nortear seus gastos e para receber de forma rápida, visual e intuitiva o controle de suas finanças. Assim o presente trabalho justifica-se por apresentar o aplicativo Mani, que busca fornecer praticidade ao usuário para que consiga manter um gerenciamento de sua vida financeira.

## 1.3 Estrutura da Monografia

A estrutura do presente trabalho divide-se da seguinte maneira: O Capítulo 2 contém a fundamentação teórica, na qual são apresentadas as tecnologias para o desenvolvimento da aplicação e a base teórica para a realização do trabalho. Já o Capítulo 3 trata do estudo da arte, que expõe alguns projetos relacionados. O Capítulo 4 detalha o roteiro que foi seguido para o desenvolvimento do sistema, bem como toda a fundamentação técnica que envolve os métodos de desenvolvimento. O Capítulo 5 demonstra os resultados obtidos e também o *feedback* dos usuários do aplicativo. Por fim, no Capítulo 6 é apresentada a conclusão e trabalhos futuros do projeto.

---

## Referencial Teórico

---

Neste capítulo serão apresentadas descrições teóricas das ferramentas utilizadas e os principais motivos da escolha de utilizá-las no desenvolvimento deste projeto.

### 2.1 Aplicações para dispositivos móveis

De acordo com Kemp (2021), 5.22 bilhões de pessoas usam um dispositivo móvel nos dias de hoje, o que corresponde a 66.6% da população mundial. Com isso, os aplicativos móveis se tornaram uma necessidade em diversos campos.

Como visto em Rahimian e Ramsin (2008), existem alguns desafios no desenvolvimento de aplicações *mobile*, por exemplo as limitações de hardware (armazenamento, capacidade de processamento, tamanho da tela, etc) e também devido a características específicas como bateria, câmera, sensores, etc.

Na Tabela 2.1 é possível ver a distribuição da porcentagem de sistemas *mobile* dentre as principais plataformas.

Tabela 2.1: Participação no mercado de sistemas operacionais móveis no mundo.

Período	Android	iOS	Outros
2020Q1	73,27%	25,91%	0,82%
2020Q2	72,38%	27,03%	0,59%
2020Q3	74,43%	24,98%	0,59%
2020Q4	72,22%	27,19%	0,59%
2021Q1	71,88%	27,41%	0,71%

Fonte: Adaptado de Stats (2021)

Como mostra a Tabela 2.1, o sistema operacional Android tem uma parcela de mais de 70% desde o primeiro quadrimestre de 2020 e que vem se mantendo até o primeiro quadrimestre de

2021. Já o sistema operacional iOS vem em segundo lugar com uma média de 29%. Com uma parcela bem pequena estão todos os outros sistemas operacionais, que ocupam menos de 1%.

Para este estudo será considerada a abordagem do desenvolvimento multiplataforma, descrita na Seção 2.1.1 do presente capítulo.

### 2.1.1 Desenvolvimento multiplataforma

No desenvolvimento multiplataforma, existe apenas um único código fonte que é compilado em código nativo de diferentes sistemas operacionais. Isso significa que o aplicativo desenvolvido pode ser executado sem problemas de incompatibilidade em sistemas Android e iOS, por exemplo (CARVALHO, 2018).

Como principais vantagens do desenvolvimento multiplataforma, pode-se citar a capacidade de escrever um código e executá-lo em diferentes plataformas, o que reduz grandemente os custos e o tempo de desenvolvimento – ao contrário das aplicações nativas. Outra vantagem significativa é a facilidade na manutenção da aplicação, que será feita apenas em um código, e não de maneira individual para cada plataforma (PALMIERI, 2012).

Assim como no desenvolvimento nativo, o desenvolvimento multiplataforma também possui algumas desvantagens. Ao contrário das aplicações nativas, as aplicações multiplataforma não se integram perfeitamente com os sistemas operacionais, então algumas aplicações podem não funcionar de maneira ideal devido à comunicação inconsistente entre o código e os componentes nativos do dispositivo (MAHARANA, 2017).

### 2.1.2 React Native

Lançado em 2015 pelo Facebook, o React Native é um framework JavaScript que possibilita a criação de aplicativos Android e iOS com a mesma base de código, sem nenhuma – ou quase nenhuma – modificação (FACEBOOK, 2015).

Em Occhino (2015), os engenheiros do Facebook defendem que diferentes plataformas têm capacidades e sensações diferentes, então devem possuir aplicações diferentes, mas os desenvolvedores devem conseguir construir aplicações para qualquer plataforma sem ter que aprender diferentes tecnologias para isso. Esse conceito utilizado na construção do React Native é chamado de *"learn once, write anywhere"* ou, em português, "aprenda uma vez, escreva em qualquer lugar".

O React Native escreve suas aplicações utilizando o JSX, uma mistura de JavaScript e XML que permite adicionar HTML ao JavaScript. Internamente, o React Native faz uma chamada para as APIs nativas das plataformas – Java para Android ou Objective-C para iOS). Deste modo, os componentes renderizados na tela são completamente nativos (EISENMAN, 2016). Isso acontece através de uma ponte que faz a comunicação entre o JavaScript e o lado nativo do dispositivo (KHOROSHULIA, 2021).

### 2.1.3 Expo

Expo é um conjunto de ferramentas e serviços que ajudam a desenvolver, construir, implantar e iterar rapidamente em aplicativos iOS, Android e Web com React Native (EXPO, 2021b).



O Expo gerencia o máximo possível da complexidade envolvida no processo de criação de uma aplicação, o que é chamado de fluxo de trabalho gerenciado. Um desenvolvedor que usa o fluxo de trabalho gerenciado não precisa configurar módulos nativos, apenas desenvolve utilizando JavaScript (CRUZ, 2019).

O SDK do Expo oferece uma coleção de soluções prontas, facilitando o trabalho com câmera, notificações, localização, acelerômetro, etc. (EXPO, 2021a) Além disso, o Expo torna possível testar aplicações para iOS sem ter um Mac, algo que normalmente não é possível. Isso acontece por meio do aplicativo Expo disponível nas lojas para Android e iOS (DUARTE, 2020).

### 2.1.4 JavaScript

Como dito anteriormente, o React Native é um framework implementado na linguagem JavaScript, logo faz-se necessário defini-la. O JavaScript é uma linguagem de programação de alto nível criada, a princípio, para ser executada em navegadores e manipular comportamentos de páginas Web. Atualmente o JS não se restringe mais às páginas e navegadores: hoje já é possível utilizar JavaScript em aplicativos *mobile*, softwares para *desktop* e até mesmo em *back-end* (JAVASCRIPT, 2021).

Originalmente, o JS é uma linguagem *client-side*, ou seja, é executada do lado do usuário. Como nada é enviado a nenhum servidor externo para processamento, as respostas são imediatas. Atualmente existem diversas ferramentas que permitem a aplicação do JS no desenvolvimento de aplicações *mobile*. O que elas fazem, basicamente, é simular o ambiente de um navegador para simular o JS ou transpilar o JS para a linguagem nativa de cada dispositivo móvel (ROVEDA, 2020).

## 2.2 Banco de dados

Os bancos de dados se tornaram essenciais no cotidiano da sociedade moderna. No decorrer do dia, a maioria das pessoas se depara com atividades que envolvem alguma interação com os bancos de dados (ELMASRI; NAVATHE, 2005).

Segundo Guimarães (2003), um banco de dados é uma coleção de dados ou informações relacionadas entre si. Elas representam aspectos do mundo real com significado próprio e que desejamos armazenar para uso futuro.

### 2.2.1 Banco de dados não relacional

Com o desenvolvimento da internet, surgiu a necessidade dos sistemas computacionais lidarem com os dados de forma mais eficiente. Para Yegulalp (2017), os bancos de dados não relacionais (NoSQL) surgiram em resposta as limitações do modelo relacional. Sistemas NoSQL conseguem armazenar e gerenciar os dados de maneira que permitem uma alta velocidade operacional e flexibilidade para os desenvolvedores. A maioria dos bancos de dados NoSQL também podem processar dados altamente estruturados, mas não se limitam apenas a isso.

Sadalage e Fowler (2015) listam quatro tipos de bancos de dados NoSQL, sendo eles:

- **Chave-valor:** É o tipo mais simples de utilizar a partir da perspectiva de uma API. Por

sempre fazer o acesso pela chave primária, esse tipo de banco tem, geralmente, um ótimo desempenho e pode ser escalado facilmente;

- **Documento:** Esse tipo foi desenvolvido para armazenar dados semiestruturados, normalmente no formato JSON ou XML;
- **Coluna:** Trabalha com o conceito de tabelas, mas diferentemente dos bancos SQL, os dados em uma tabela não precisam seguir uma estrutura, ou seja, nem todas as linhas têm a mesma quantidade de colunas;
- **Grafo:** Esse tipo permite que sejam armazenadas entidades e também relacionamentos entre elas. A organização do grafo permite que os dados sejam armazenados uma vez e depois interpretados de formas diferentes baseados em relacionamentos.

## 2.2.2 JSON

Utilizado no modelo de banco de dados NoSQL baseado em documentos, JSON é um formato baseado em texto padrão para representar dados estruturados com base na sintaxe do objeto JavaScript. O JSON existe como uma string, útil quando você deseja transmitir dados por uma rede, pois permite uma troca que não consuma muitos recursos (MOZILLA, 2021).

Para seres humanos, é fácil de ler e escrever. Para máquinas, é fácil de interpretar e gerar (JSON, 2021). A seguir, é possível ver um modelo de objeto JSON.

```
{
  "historico" : {
    "hKcVveH4ALZ1uEfdWLE4dj0DaSs1" : {
      "-MkcFir06KmhVwmCd3Ps" : {
        "categoria" : "Alimentação",
        "data" : "27/09/2021",
        "descricao" : "almoço",
        "tipo" : "despesa",
        "valor" : 16
      }
    }
  }
}
```

## 2.2.3 Firebase

O Firebase é um *Backend-as-a-Service* – *BaaS* que fornece ferramentas para desenvolver aplicativos de alta qualidade. Adquirido pelo Google em 2014, o Firebase fornece a maior parte dos recursos de forma gratuita, mas limitada, e disponibiliza também planos de franquia ou o modo de pagamento por demanda (GOOGLE, 2014a).

Esplin (2016) descreve o Firebase como um serviço que liberta o desenvolvedor e permite que ele foque apenas no objetivo principal, pois não é necessário gerenciar servidores ou escrever APIs. O próprio Firebase é seu servidor, sua API e seu armazenamento de dados.

Como visto em Google (2014c), o Firebase disponibiliza diversos serviços, dentre eles os serviços que são utilizados neste trabalho:

- **Autenticação**, que possui diversos métodos para gerenciar usuários (GOOGLE, 2014b);
- **Realtime Database**, que armazena e sincroniza dados quase em tempo real, online ou offline (GOOGLE, 2014d);

## 2.3 Visão computacional

A visão computacional é uma área da ciência que estuda e desenvolve teorias voltados à extração de informações contidas em imagens, vídeos ou qualquer estrutura de dados multi-dimensional do mundo real (GARCIA, 2015).

Uma das principais tarefas de qualquer aplicação que envolva algum tipo de visão computacional é a interpretação ou análise de imagens, resultando em uma descrição da imagem que contenha os objetos de interesse encontrados dela com o mínimo de intervenção humana e com um razoável grau de confiabilidade (PEDRINI; SCHWARTZ, 2007).

No geral, sistemas de visão computacional tentam imitar a visão humana, extraíndo informações relevantes a partir de imagens. Ao longo dos anos, esses sistemas têm sido utilizados com diversos propósitos em inúmeras áreas de aplicação. A evolução das técnicas permite um contínuo aperfeiçoamento neste campo (ALIGER, 2020).

## 2.4 Considerações Finais

Nesse capítulo foram apresentadas as ferramentas utilizadas no desenvolvimento do projeto.

---

## Estado da Arte

---

Nesta seção serão apresentados trabalhos que possuem finalidade semelhante à solução proposta neste trabalho. A escolha dos aplicativos analisados foi feita com base nos mais bem avaliados da Google Play Store na categoria finanças pessoais.

### 3.1 Organizze

Lançado em 2015, o Organizze (2021) é um gerenciador financeiro pessoal disponível para Android, iOS e Web. Ele foi desenvolvido utilizando as linguagens Java/Kotlin para Android, RubyMotion para iOS e AngularJS/Ruby para Web. O banco de dados utilizado é o MySQL.

No Organizze, existem diversas opções para auxiliar na administração de recursos: é possível administrar várias contas e cartões de crédito, gerar relatórios precisos sobre os gastos, realizar planejamentos sobre investimentos, estipular metas pra economizar e receber alertas sobre estas metas, realizar backups dos dados, importações de planilhas e extratos bancários e muito mais.

Atualmente existem duas versões: a básica que possui uma quantidade limitada de recursos e a *premium* que dá acesso aos recursos de forma ilimitada, podendo ser visto posteriormente na Tabela 3.1.

Na Figura 3.1a é possível ver um exemplo de sua interface.

### 3.2 Mobills

Lançado em 2013, o Mobills (2021) é um sistema de controle financeiro pessoal que possui versões para Android, iOS e Web. Nele, existem diversas opções para auxiliar no gerenciamento de recursos: é possível administrar contas e cartões de crédito, sincronizar dados na nuvem com Android e iOS, gerar relatórios de orçamentos mensais, importar planilhas e muito mais.

Atualmente existem duas versões: a gratuita que possui uma quantidade limitada de recursos e a *premium* que dá acesso a todas as funcionalidades disponibilizados pela ferramenta, podendo

ser visto posteriormente na Tabela 3.1.

É importante mencionar que, atualmente, o Mobills é o aplicativo número 1 da lista de mais rentáveis na categoria finanças e também possui o selo Escolha dos editores, que disponibiliza uma seleção atualizada periodicamente dos melhores aplicativos, escolhidos por editores da Google Play Store.

Na Figura 3.1b é possível ver um exemplo de sua interface.

### 3.3 Fortuno

Lançado em 2018, o Fortuno (2021) é um aplicativo de controle financeiro pessoal disponível para Android. Ele foi desenvolvido utilizando as linguagens Java/Kotlin e o banco de dados MySQL.

O Fortuno disponibiliza diversas funcionalidades. Nele é possível controlar contas e cartões de crédito, gerar relatórios e gráficos precisos sobre gastos, definir orçamentos, cadastrar e receber alertas sobre pendências, sincronizar dados na nuvem e muito mais.

Atualmente existem duas versões: a básica que possui uma quantidade limitada de recursos e a *premium* que dá acesso a todos os recursos de forma ilimitada, podendo ser visto posteriormente na Tabela 3.1.

Na Figura 3.1c é possível ver um exemplo de sua interface.

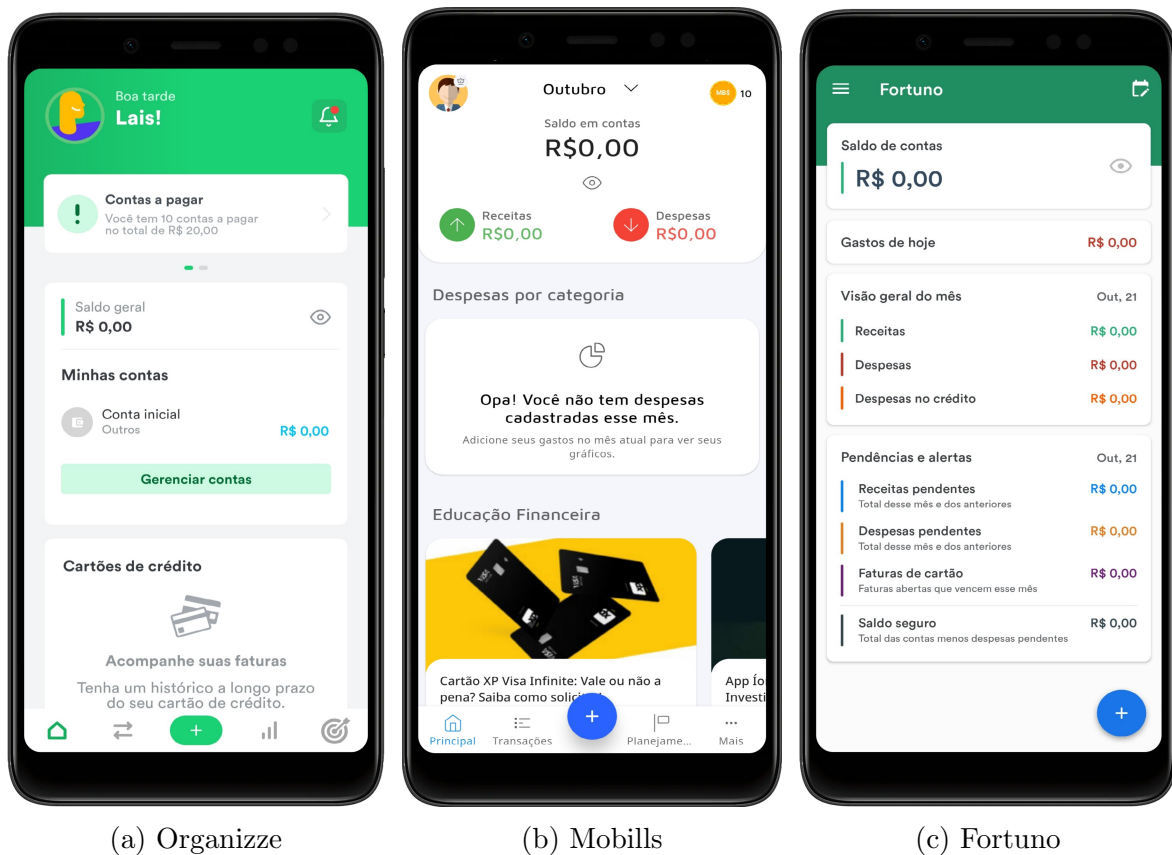


Figura 3.1: Estados da arte

## 3.4 Considerações Finais

Nesta seção, é apresentada uma comparação dos aplicativos analisados anteriormente, através da Tabela 3.1, que faz uma análise das principais funcionalidades disponíveis no plano básico de cada um.

Tabela 3.1: Comparativo de funcionalidades

	<b>Organizze</b>	<b>Mobills</b>	<b>Fortuno</b>
<b>Lançamentos</b>	Ilimitado	Ilimitado	Ilimitado
<b>Metas</b>	3 metas <sup>1</sup>	Limitado <sup>1</sup>	Ilimitado
<b>Contas</b>	2 contas <sup>1</sup>	Limitado <sup>1</sup>	2 contas
<b>Cartões de crédito</b>	1 cartão <sup>1</sup>	Limitado <sup>1</sup>	1 cartão
<b>Categorias</b>	Ilimitado	Ilimitado	Ilimitado
<b>Tags</b>	3 tags <sup>1</sup>	Limitado <sup>1</sup>	Não
<b>Relatórios</b>	Ilimitado	Limitado <sup>1</sup>	Ilimitado
<b>Bloqueio por senha e biometria</b>	Sim	Sim	Não
<b>Importação e exportação de dados</b>	Sim	Sim	Não <sup>3</sup>
<b>Multiplataforma</b>	Sim	Android, iOS <sup>2</sup>	Não

Analisando a tabela comparativa é possível ver que, dentre os aplicativos analisados, os aplicativos Organizze e Mobills possuem a mesma quantidade de funcionalidades. Apesar disso, o aplicativo Organizze possui mais funcionalidades ilimitadas na versão gratuita.

No geral, os três aplicativos possuem as mesmas funcionalidades principais, com exceção do Fortuno. Além de não possuir o bloqueio por biometria, ele não é multiplataforma, estando disponível apenas no sistema operacional Android.

Em questões como navegação e *design*, o Fortuno também fica para trás. Todos os aplicativos disponibilizam o modo escuro, ótima opção para economia de bateria e acessibilidade, mas apenas o Fortuno não disponibiliza a opção de personalizar a tela inicial do app, deixando a desejar no quesito navegação.

Feito esse levantamento de características entre os aplicativos, foi possível captar os pontos positivos e negativos entre eles. Tudo isso foi levado em consideração durante a idealização e desenvolvimento do projeto apresentado neste trabalho.

---

<sup>1</sup>Ilimitado na versão *premium*.

<sup>2</sup>Android, iOS e Web na versão *premium*.

<sup>3</sup>Apenas na versão *premium*.

---

# Metodologia

---

Nesse capítulo serão descritos os estudos e a forma na qual as ferramentas citadas no Capítulo 2 foram utilizadas no desenvolvimento do projeto.

## 4.1 Projeto da aplicação

Para desenvolver este projeto, foi realizado um estudo sobre a construção de aplicações para dispositivos móveis e suas características, além de uma pesquisa sobre o desenvolvimento multiplataforma e o *framework* adequado para construir esta aplicação.

Para a criação da aplicação foi usado o editor de código Visual Studio Code, que agilizou o processo de construção da aplicação por possuir ferramentas de apoio ao desenvolvimento de software.

O *framework* React Native foi utilizado na criação da interface do usuário juntamente com o Expo, que possibilitou a execução da aplicação em um dispositivo físico através do aplicativo Expo Go.

Foram utilizados dois serviços do Firebase: a autenticação, que auxiliou no processo de desenvolvimento do cadastro e login no aplicativo, e o banco de dados em tempo real, para o armazenamento dos dados.

## 4.2 Planejamento

Para um desenvolvimento eficiente é necessário, inicialmente, de um bom planejamento do software. Existem diversos recursos para auxiliar nesse processo, inclusive os que serão citados posteriormente nesta seção.

### 4.2.1 Levantamento de requisitos

O levantamento de requisitos faz-se muito importante na construção de um software, pois define as funcionalidades necessárias para seu desenvolvimento (MENDONÇA, 2014). É importante que os requisitos sejam bem definidos para um desenvolvimento eficiente e um resultado de sucesso.

Após reunir o conhecimento adquirido através de pesquisas e estudo sobre aplicativos na área financeira, foi realizado o levantamento de requisitos do sistema, que podem ser observados abaixo:

- Cadastro de usuário: o usuário precisa criar uma conta para ter acesso à aplicação;
- Login: permitir ao usuário realizar o login por meio de email e senha;
- Registros: o usuário deve conseguir registrar suas receitas manualmente, e as despesas de maneira manual ou automática (através do leitor de QR code);
- Relatórios: disponibilizar ao usuário seus relatórios financeiros mensais e diários;
- Categorização: todas as movimentações podem possuir etiquetas de categorias para serem encontradas de maneira mais fácil futuramente;
- Notícias: apresentar notícias sobre o mundo financeiro em uma aba específica.

### 4.2.2 Casos de uso

Os diagramas de caso de uso fazem parte da UML e permitem a representação visual do sistema com suas funcionalidades e relacionamentos. Neste diagrama, procura-se identificar os atores (usuários, outros sistemas e até mesmo algum hardware especial) que de alguma forma serão utilizados no software, bem como os serviços que o sistema disponibilizará a eles (GUEDES, 2005).

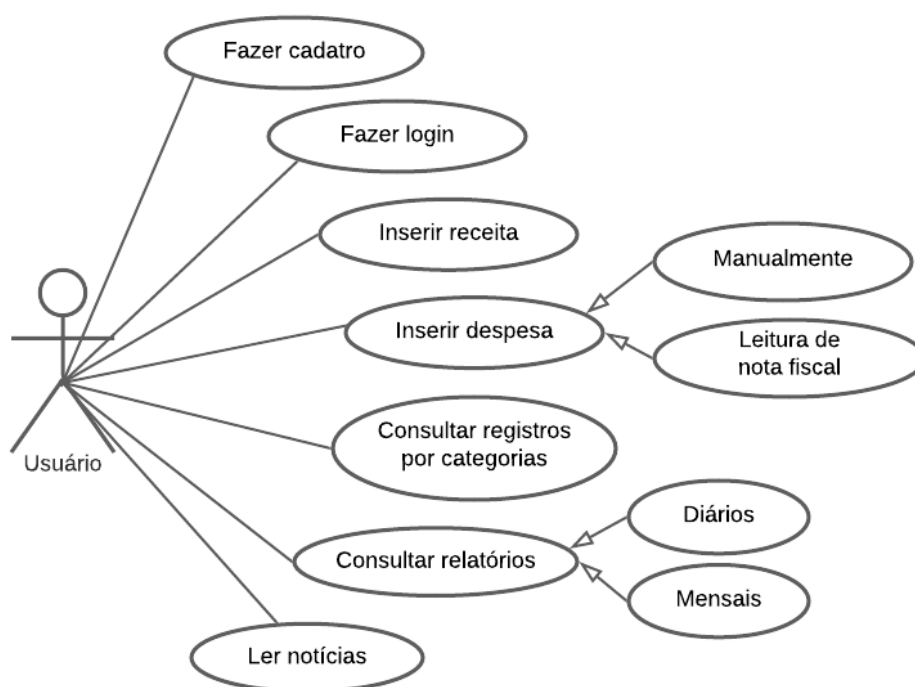
Na Figura 4.1 é possível visualizar o diagrama de casos de uso do aplicativo Mani, construído através do software online Lucidchart.<sup>1</sup> No diagrama estão todos os casos da aplicação, sendo eles: cadastro, login, inserção de receitas, inserção de despesas seja manualmente ou por leitura de nota fiscal, consulta de registros por categoria, consulta de relatórios diários, consulta de relatórios mensais e leitura de notícias. Cada um dos casos será detalhado posteriormente em sua determinada tabela.

---

<sup>1</sup><https://www.lucidchart.com/pages/>



Figura 4.1: Diagrama de Casos de Uso - Mani



Fonte: Autor

Na Tabela 4.1 está especificado o cadastro de usuário. Para se cadastrar, o único requisito é possuir um endereço de email. Após o cadastro, o usuário pode acessar a aplicação.

Tabela 4.1: UC001: Fazer cadastro

Ator	Usuário
Fluxo principal	1) Acessar a aplicação; 2) Entrar na página de cadastro; 3) Preencher os campos solicitados (nome, email e senha).
Fluxo alternativo	1) Deixar algum campo vazio.
Precondições	1) Possuir endereço de email.
Pós-condições	1) O usuário estará habilitado para acessar a aplicação.

Na Tabela 4.2 está especificado o login na aplicação. Para realizar o login, o usuário deve possuir um cadastro. Após realizar o login, o usuário poderá acessar a aplicação.

Tabela 4.2: UC002: Fazer login

Ator	Usuário
Fluxo principal	1) Acessar a aplicação; 2) Entrar na página de login; 3) Preencher os campos solicitados (email e senha).
Fluxo alternativo	1) Deixar algum campo vazio ou preencher incorretamente.
Precondições	1) Possuir cadastro na aplicação.
Pós-condições	1) O usuário estará logado na aplicação.

Na Tabela 4.3 está descrito o processo de inserção de receita. Para inserir uma receita é

necessário estar logado na aplicação e logo depois, o usuário é direcionado à página inicial onde poderá ver suas últimas movimentações.

Tabela 4.3: UC003: Inserir receita

Ator	Usuário
Fluxo principal	1) Acessar a aplicação; 2) Entrar na página de cadastro de receita; 3) Preencher os campos solicitados (valor, categoria, descrição e data).
Fluxo alternativo	1) Deixar algum campo obrigatório vazio.
Precondições	1) Estar logado na aplicação.
Pós-condições	1) A receita estará cadastrada.

Nas Tabelas 4.4 e 4.5 estão detalhadas as duas maneiras de inserir uma despesa: manualmente e através da leitura de nota fiscal. Para inserir uma despesa é necessário estar logado na aplicação e, no caso da inserção por nota fiscal, também é necessário permitir o acesso à câmera. O valor da despesa é obtido por meio da URL captada através leitura do QR code contido na nota fiscal. Em ambos os casos, logo após a inserção o usuário é direcionado à página inicial onde poderá ver suas últimas movimentações.

Tabela 4.4: UC004: Inserir despesa - Manualmente

Ator	Usuário
Fluxo principal	1) Acessar a aplicação; 2) Entrar na página de cadastro de despesa; 3) Preencher os campos solicitados (valor, categoria, descrição e data).
Fluxo alternativo	1) Deixar algum campo obrigatório vazio.
Precondições	1) Estar logado na aplicação.
Pós-condições	1) A despesa estará cadastrada.

Tabela 4.5: UC005: Inserir despesa - Leitura de nota fiscal

Ator	Usuário
Fluxo principal	1) Acessar a aplicação; 2) Entrar na página de cadastro de despesa; 3) Clicar no botão para ler a nota fiscal; 5) Preencher os campos solicitados (categoria, descrição e data).
Fluxo alternativo	1) Deixar algum campo obrigatório vazio.
Pre-condições	1) Estar logado na aplicação; 2) Permitir acesso à câmera.
Pós-condições	1) A despesa estará cadastrada.

Na Tabela 4.6 está especificado o caso de consulta de registros por categoria. Neste caso o usuário consegue selecionar uma categoria e assim filtrar os registros referentes a ela.

Tabela 4.6: UC006: Consultar registros por categoria

Ator	Usuário
Fluxo principal	1) Acessar a aplicação; 2) Acessar a seção de categorias; 3) Selecionar a categoria desejada.
Fluxo alternativo	
Precondições	1) Estar logado na aplicação.
Pós-condições	1) O usuário conseguirá ver os registros referentes a categoria selecionada.

Nas Tabelas 4.7 e 4.8 estão detalhadas as duas maneiras de ver relatórios através de data: selecionando pelo dia ou mês. O usuário escolhe o dia ou mês desejado e conseguirá ver as movimentações referentes à data selecionado.

Tabela 4.7: UC007: Consultar relatórios - Diários

Ator	Usuário
Fluxo principal	1) Acessar a aplicação; 2) Acessar a seção de relatórios diários; 3) Selecionar o dia desejado.
Fluxo alternativo	
Precondições	1) Estar logado na aplicação.
Pós-condições	1) O usuário conseguirá ver os registros referentes ao dia selecionado.

Tabela 4.8: UC008: Consultar relatórios - Mensais

Ator	Usuário
Fluxo principal	1) Acessar a aplicação; 2) Acessar a seção de relatórios; 3) Selecionar o ano desejado; 4) Selecionar o mês desejado.
Fluxo alternativo	
Precondições	1) Estar logado na aplicação.
Pós-condições	1) O usuário conseguirá ver os registros referentes ao mês selecionado.

Na Tabela 4.9 está descrito o caso de leitura de notícias. As notícias utilizadas nesta seção são as notícias do blog Mobills<sup>2</sup>, que foram disponibilizadas pela equipe através de uma API.

Tabela 4.9: UC009: Ler notícias

Ator	Usuário
Fluxo principal	1) Acessar a seção de notícias.
Fluxo alternativo	
Precondições	1) Estar logado no sistema.
Pós-condições	1) O usuário conseguirá ver notícias do blog Mobills.

<sup>2</sup><https://www.mobills.com.br/blog/>

### 4.2.3 Prototipação

A prototipação é a construção de um modelo representativo da aplicação. Nela é possível identificar funcionalidades, componentes a serem reutilizados e o fluxo que o usuário irá seguir.

Com a visualização geral do aplicativo que a prototipação proporciona, se torna mais fácil identificar problemas e corrigí-los.

No desenvolvimento do protótipo deste projeto foi utilizado um editor online de *design* gráfico chamado Figma <sup>3</sup>, que tem ênfase na prototipagem de interfaces gráficas e estruturas de *design* de experiência de usuário. Ele possibilita a criação de protótipos navegáveis, assim, através de um *link*, é possível navegar pelas telas da aplicação antes mesmo dela ser desenvolvida.

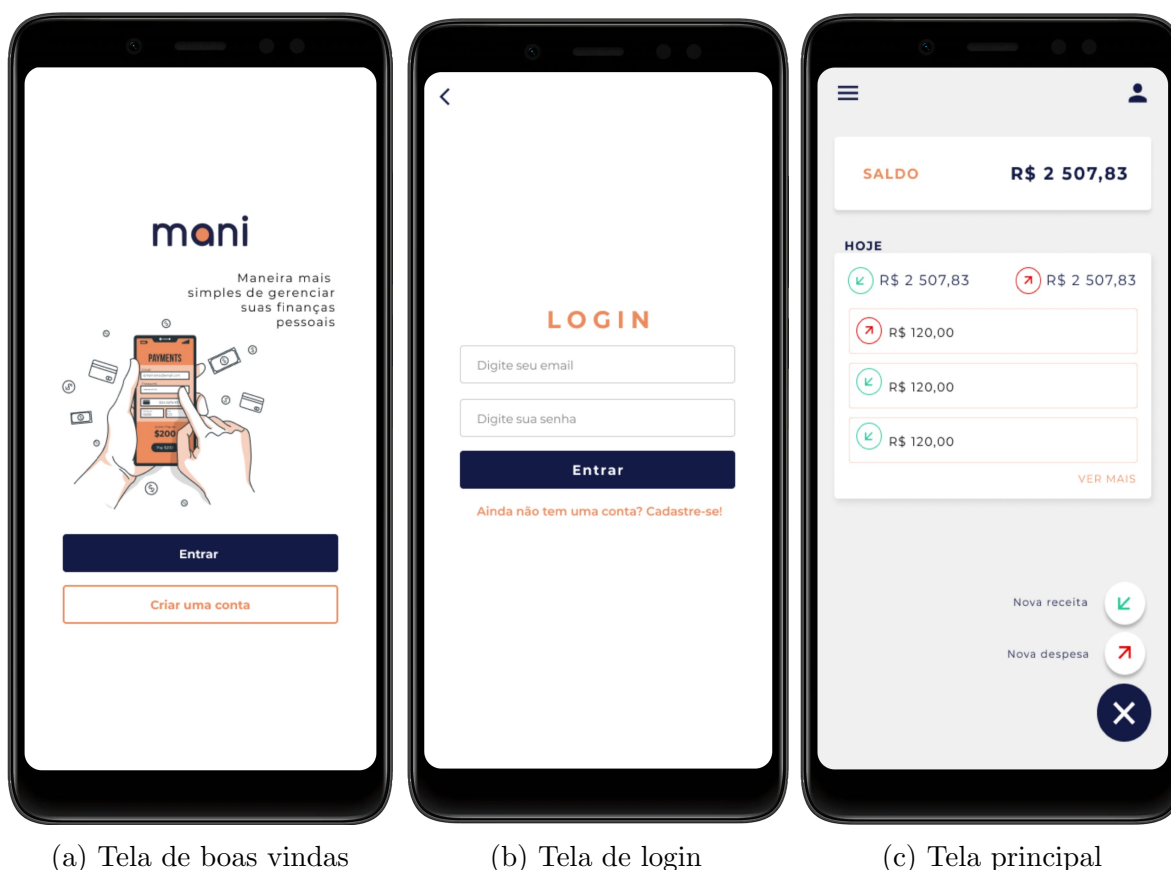


Figura 4.2: Interfaces do aplicativo Mani

Na Figura 4.2 estão três exemplos da interface do aplicativo Mani planejados no Figma, tratando-se a Figura 4.2a da tela de boas vindas, a Figura 4.2b da tela de login e a Figura 4.2c da tela principal.

### 4.2.4 Banco de dados

O banco de dados utilizado na aplicação é o Realtime Database, ofertado pelo Firebase. Ele é hospedado na nuvem e seus dados são armazenados em documentos JSON, sincronizados em tempo real com todos os clientes conectados.

<sup>3</sup><https://www.figma.com/>

O banco será responsável pelo armazenamento de dados dos usuários do aplicativo, como seu saldo, movimentações e informações pessoais.

Na Figura 4.3 é possível visualizar a estrutura desses dados no banco. Na coleção de usuários, cada um possui seu identificador único que é utilizado para encontrar seus dados em outras coleções, e também possui um nome e um saldo. Na coleção do histórico estão as movimentações de cada usuário, separadas pelo identificador de cada um. Cada movimentação possui também um identificador, além de possuir categoria, data, descrição, tipo e valor. Já as categorias possuem seu nome e um tipo, que pode ser receita ou despesa.

### **4.3 Considerações Finais**

Este capítulo abordou o planejamento e metodologia utilizada para realizar o desenvolvimento da aplicação, utilizando a linguagem UML para visualizar, modelar e documentar o projeto da aplicação. A conclusão deste capítulo possibilitou uma melhor visualização e simplificou o processo de desenvolvimento do projeto.

mani-2a17c-default-rtdb



Figura 4.3: Estrutura do banco de dados

---

# Resultados

---

O objetivo deste capítulo é demonstrar todas as funcionalidades do aplicativo, juntamente com o *feedback* do usuário e os resultados obtidos com o desenvolvimento.

## 5.1 Apresentação do sistema

O aplicativo Mani foi finalizado em seis meses de desenvolvimento e conta com um total de 12 telas: boas vindas, cadastro, login, tela principal, categorias, notícias, inserção de receita, inserção de despesa, perfil, editar perfil, relatórios diários e relatórios mensais. Cada uma das telas será abordada nas subseções seguintes.

### 5.1.1 Tela de boas vindas

A primeira tela é a tela de boas vindas, que pode ser vista na Figura 5.1a. Ela contém uma breve apresentação do sistema e possibilita o usuário seguir para o Cadastro ou para o Login.

### 5.1.2 Cadastro

A Figura 5.1b apresenta a tela de cadastro, que é o resultado da implementação do caso de uso UC001 mostrado na Tabela 4.1. Para conseguir acessar o aplicativo, é necessário que o usuário tenha um cadastro, que pode ser feito nesta tela. Ele pode inserir seus dados e criar uma conta, ou se já tiver uma conta, pode navegar para a tela de Login.

### 5.1.3 Login

A Figura 5.1c apresenta a tela de login, que é o resultado da implementação do caso de uso UC002 mostrado na Tabela 4.2. Nesta tela, o usuário pode inserir os dados requisitados e entrar em sua conta ou, se não possuir um cadastro ainda, pode navegar para a tela de Cadastro.

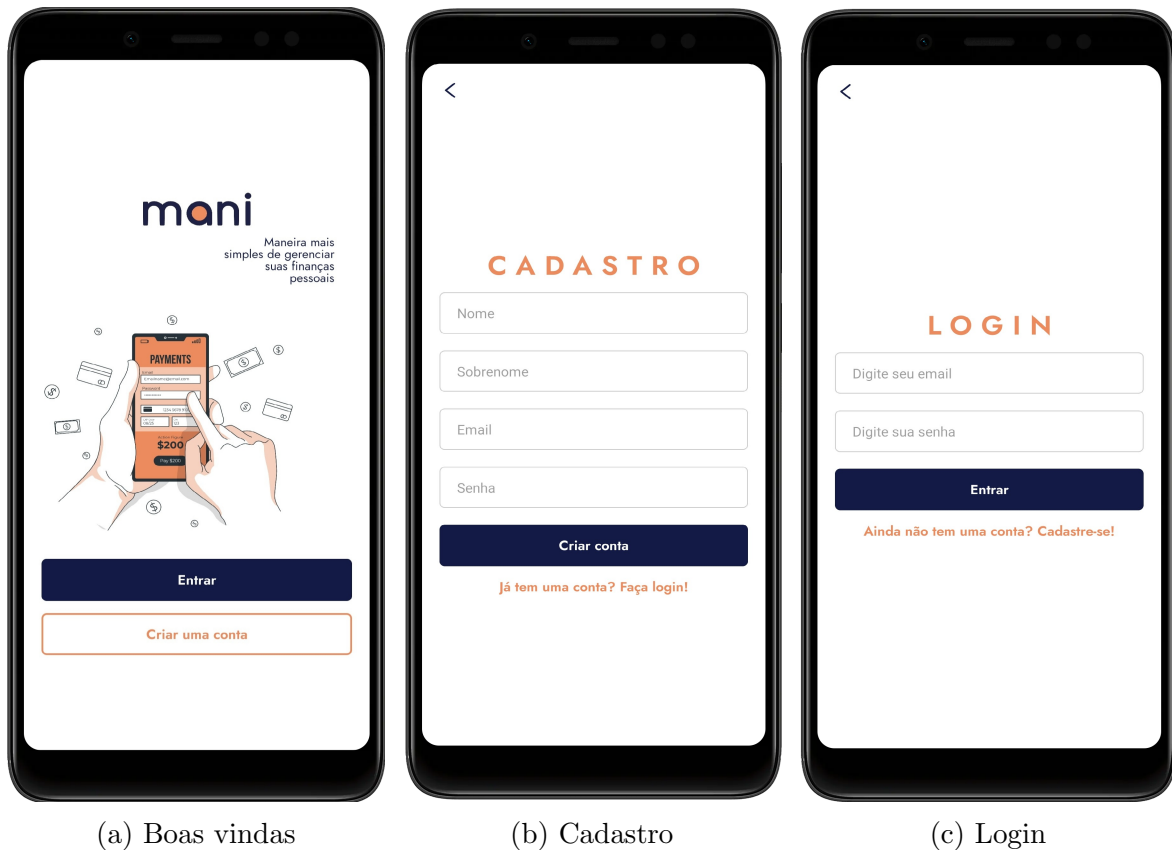


Figura 5.1: Mani - Telas de boas vindas, cadastro e login

### 5.1.4 Tela principal

Após a autenticação, o usuário é redirecionado para a tela inicial do aplicativo, representada na Figura 5.2. Nela são exibidas algumas informações, como o saldo total e os últimos registros do dia atual.

### 5.1.5 Categorias

A Figura 5.3 apresenta variações da tela de categoria, resultantes da implementação do caso de uso UC006 mostrado na Tabela 4.6. Para que o usuário consiga visualizar facilmente suas despesas e receitas, o aplicativo disponibiliza categorias. Na Figura 5.3 estão representadas três situações: 1) as categorias disponíveis, 2) a inserção de uma nova categoria e 3) os registros contidos dentro de uma categoria.

### 5.1.6 Notícias

A Figura 5.4 apresenta a tela de notícias, que é o resultado da implementação do caso de uso UC009 mostrado na Tabela 4.9. A tela de notícias contém as últimas publicações do blog Mobills<sup>1</sup>, buscadas através de uma API disponibilizada pela equipe e possibilita a leitura na íntegra através do botão "Ler mais".

<sup>1</sup><https://www.mobills.com.br/blog/>



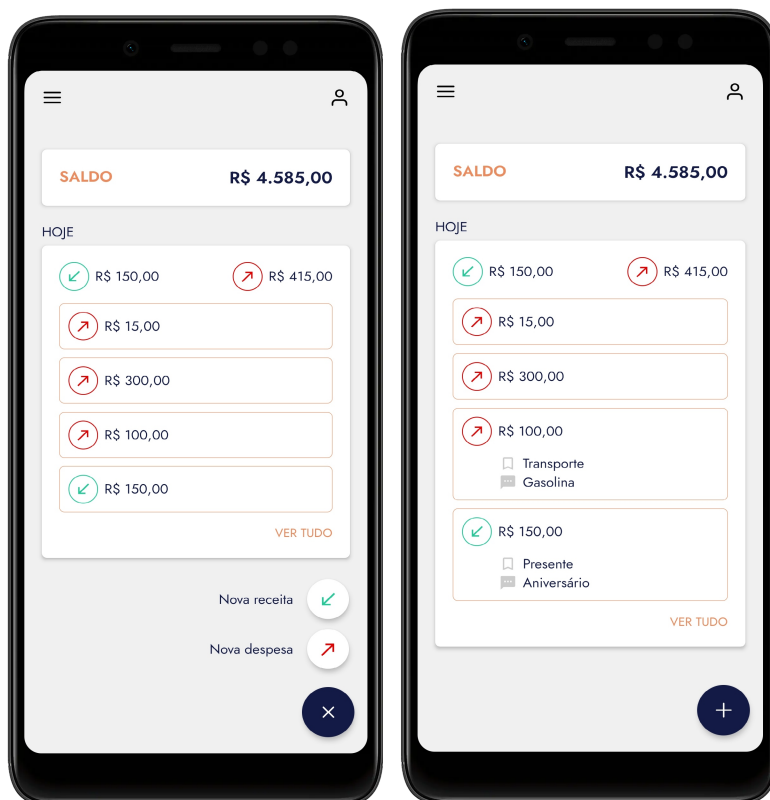


Figura 5.2: Mani - Variações da tela principal

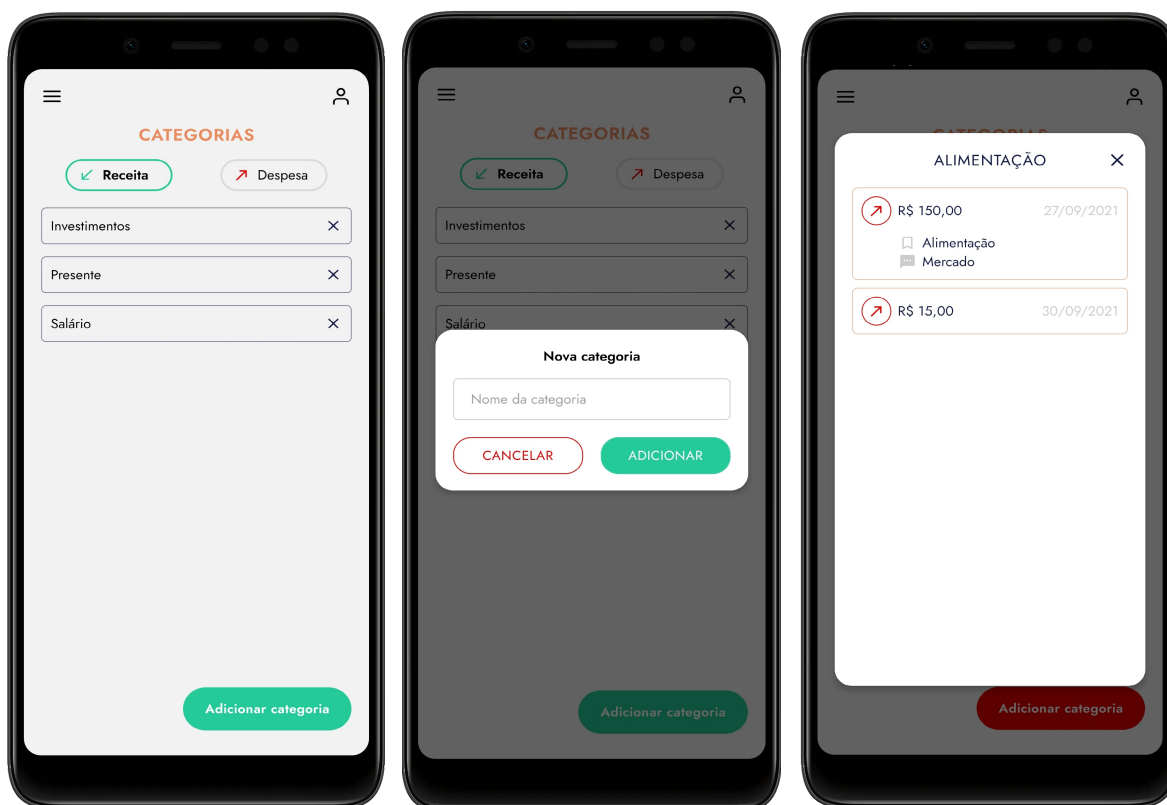


Figura 5.3: Mani - Categorias, nova categoria e registros de uma categoria



Figura 5.4: Mani - Notícias

### 5.1.7 Inserção de receitas

A Figura 5.5a apresenta a tela de inserção de uma nova receita, que é o resultado da implementação do caso de uso UC003 mostrado na Tabela 4.3. A inserção deve ser feita manualmente, com a inserção dos dados requisitados.

### 5.1.8 Inserção de despesas

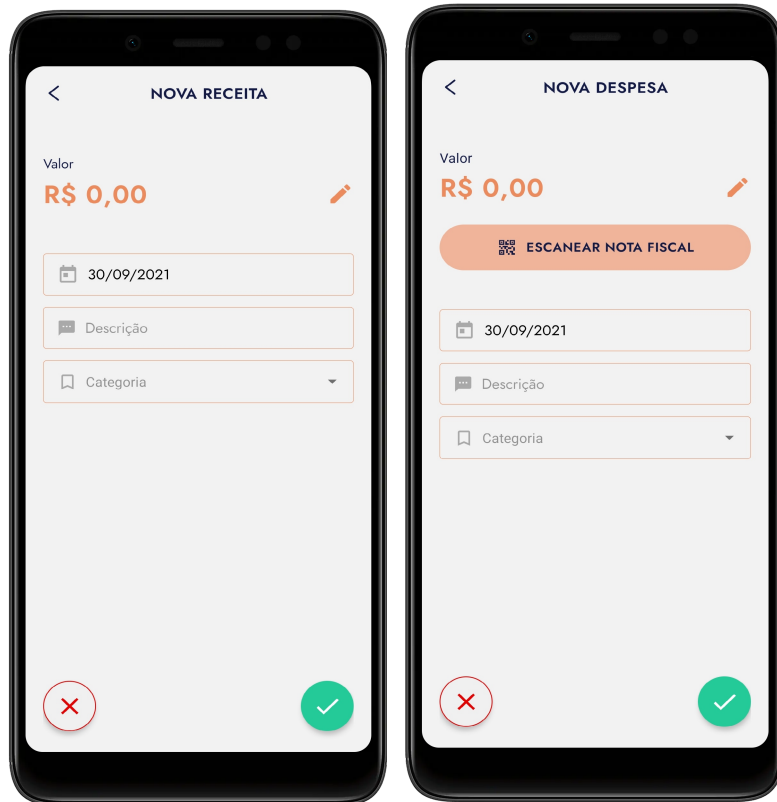
A Figura 5.5b apresenta a tela de inserção de uma nova despesa, que é o resultado da implementação dos casos de uso UC004 e UC005, respectivamente mostrados nas Tabelas 4.4 e 4.5. A despesa pode ser registrada manualmente com a inserção dos dados requisitados ou de forma automática, com a leitura de QR code contido em notas fiscais.

### 5.1.9 Perfil

Ao navegar para o Perfil, o usuário consegue visualizar seus dados, além de se desconectar da aplicação e navegar para a tela de edição de dados. Se desconectando da aplicação, o usuário será levado de volta à tela de boas vindas. Se navegar para a tela de edição de dados, conseguirá alterar os dados cadastrais. Essas possibilidades estão representadas na Figura 5.6.

### 5.1.10 Relatórios diários

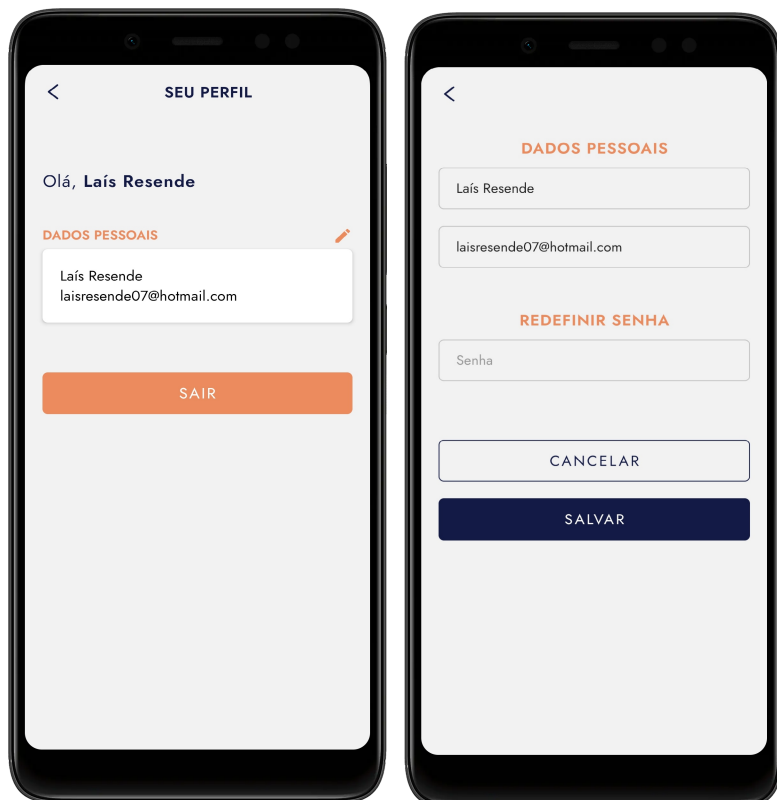
A Figura 5.7a apresenta a tela de relatórios diários, que é o resultado da implementação do caso de uso UC007 mostrado na Tabela 4.7. Nesta tela, o usuário pode selecionar o dia desejado no calendário e poderá ver todos os registros referentes àquele dia.



(a) Nova receita

(b) Nova despesa

Figura 5.5: Mani - Telas de inserção de receita e despesa



(a) Perfil

(b) Edição de dados

Figura 5.6: Mani - Perfil

### 5.1.11 Relatórios mensais

A Figura 5.7b apresenta a tela de relatórios mensais, que é o resultado da implementação do caso de uso UC008 mostrado na Tabela 4.8. Nesta tela, o usuário pode selecionar o ano e o mês desejados e poderá ver todos os registros referentes àquele mês.

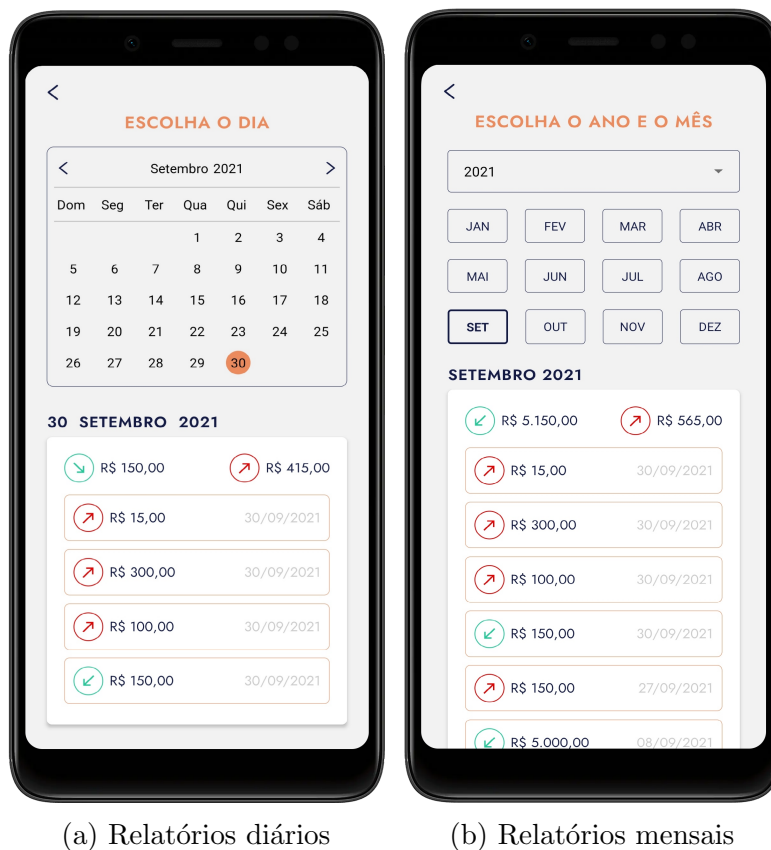


Figura 5.7: Mani - Telas de relatórios diários e mensais

## 5.2 Avaliações

Após o fim do desenvolvimento, o aplicativo foi disponibilizado ao público na Google Play Store para ser testado e avaliado por usuários reais. Isso foi possível através do Google Play Console<sup>1</sup>, que é a plataforma destinada ao desenvolvedor para que possa publicar e administrar os aplicativos.

O aplicativo teve um total de 42 instalações e conta com 23 avaliações, 100% delas com a nota máxima disponível, como apresentado na Figura 5.8. Analisando estas avaliações, é possível concluir que o aplicativo teve seus objetivos alcançados, visto que os usuários perceberam e aprovaram suas principais características, pensadas desde o planejamento, como a praticidade, as funcionalidades objetivas e a interface intuitiva e simples.





<sup>1</sup><https://play.google.com/console/>

## RESENHAS

 Política de avaliações e informações

5,0  
★★★★★  
Total:  23



-  **★★★★★** 15 de outubro de 2021  
Dps de muitos, encontrei um q facilita e agiliza. Faz o necessário sem exageros de interface. Gostei, muito útil, simples e rápido de usar, para qualquer momento.
-  **★★★★★** 17 de outubro de 2021  
Não apenas prático, como lindo e bem organizado. As funções são super diretas e facilitam muito na hora do planejamento e controle financeiro!!
-  **★★★★★** 25 de setembro de 2021  
Excelente aplicativo, muito útil e necessário!! Obrigada e parabéns!!!
-  **★★★★★** 21 de novembro de 2021  
O app é sensacional, facilita muito na hora de cuidar das finanças. Amei.

[LER TODOS OS COMENTÁRIOS](#)

Figura 5.8: Avaliações de usuários da Google Play

---

## Conclusões

---

Como visto anteriormente no Capítulo 1, é importante que as pessoas encontrem recursos que os auxiliem no controle do orçamento e os incentivem a buscar conhecimento na área financeira.

Com base nos resultados e avaliações dos usuários que podem ser vistos no Capítulo 5, pode-se concluir que os objetivos foram alcançados. A própria aprovação do aplicativo na Google Play ajuda a atestar sua qualidade, visto que passou pelos testes da loja antes de ser disponibilizado para download. Entre várias opiniões dos usuários, as características do aplicativo que mais se destacaram foram a praticidade e a usabilidade.

Todas as fases do desenvolvimento foram indispensáveis para que o projeto resultasse em um aplicativo útil, intuitivo e performático.

Como os usuários buscam novas tecnologias, o desenvolvimento do aplicativo foi desenvolvido com linguagem e *framework* atuais, garantindo assim um bom desempenho e usabilidade. O banco de dados utilizado também foi pensado na experiência do usuário, tornando possível a atualização dos dados em tempo real.

Apesar dos resultados positivos obtidos, existem funcionalidades que não foram implementadas que podem deixar o aplicativo ainda mais prático e comercial, como a sincronização com contas bancárias e cartões de crédito. Essa funcionalidade está presente em alguns aplicativos disponíveis na Google Play, mas analisando-os, foi possível observar que a sincronização possui algumas falhas, como por exemplo um tempo de carregamento muito alto, causando uma disfunção. Dito isso, algumas melhorias serão desenvolvidas futuramente para que haja um melhor planejamento e para que sejam implementadas da melhor maneira.

## **6.1 Trabalhos futuros**

Após o desenvolvimento do aplicativo, surgiram novas ideias de melhorias e funcionalidades para facilitar a vida dos usuários, além de tornar o aplicativo um pouco mais comercial e ter valor de mercado.

### **6.1.1 Leitura de código de barra**

Da mesma forma que é possível realizar o registro de uma despesa por meio de leitura de QR code contido em notas fiscais, a ideia é implementar o registro também por meio de leitura de código de barra contido em boletos bancários.

### **6.1.2 Login através de biometria**

Para tornar o processo de login ainda mais simples, a ideia é implementar o login através de biometria, possibilitando o acesso à aplicação com menos cliques do usuário.

### **6.1.3 Sincronização**

Para simplificar o gerenciamento, a ideia é implementar um sistema que possibilite ao usuário importar dados de suas contas e cartões de crédito de maneira automática, não sendo necessário o registro manual para as compras realizadas.

### **6.1.4 Transações recorrentes**

De modo a facilitar o controle, a ideia é implementar a possibilidade de, ao registrar uma despesa ou receita, conseguir marcá-la como fixa, fazendo com que seja descontada ou compensada automaticamente todo mês, sem que haja a necessidade de o usuário fazer isso manualmente.

# Referências Bibliográficas

---

- ALIGER. *Visão computacional e suas aplicações*. 2020. Disponível em: <<https://www.aliger.com.br/blog/saiba-o-que-e-visao-computacional/>>.
- CARVALHO, L. P.; DIAS, F. S.; FREIRE, A. P. An analysis of five different native and web-hybrid platforms for building android apps and their accessibility for screen readers. *SBC Journal on Interactive Systems*, 2018.
- CERBASI, G. *Como Organizar Sua Vida Financeira*. [S.l.]: Sextante, 2015.
- CODD, E. F. *A relational model of data for large shared data banks*. New York: Communications of the ACM, 1983.
- CRUZ, R. *React Native + Expo*. 2019. Disponível em: <<https://rogercruz.medium.com/react-native-com-expo-um-exemplo-40e5574c6904>>.
- DUARTE, L. *Introdução ao React Native*. 2020. Disponível em: <<https://www.luiztools.com.br/post/introducao-ao-react-native/>>.
- EISENMAN, B. *Learning React Native*. [S.l.]: O'Reilly Media, Inc., 2016.
- ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados*. São Paulo: Pearson Addison Wesley, 2005.
- ESPLIN, C. *What is Firebase?* 2016. Disponível em: <<https://howtofirebase.com/what-is-firebase-fcb8614ba442>>.
- EXPO. *API Reference*. 2021. Disponível em: <<https://docs.expo.io/versions/latest/>>.
- EXPO. *Expo*. 2021. Disponível em: <<https://docs.expo.io/>>.
- FACEBOOK, I. *React Native - A framework for building native apps with React*. 2015. Disponível em: <<https://github.com/facebook/react-native>>.
- FORTUNO. *Fortuno*. 2021. Disponível em: <<https://fortuno.app/>>.
- GARBADE, D. M. J. *Native vs. cross-platform app development: pros and cons*. 2018. Disponível em: <<https://codeburst.io/native-vs-cross-platform-app-development-pros-and-cons-49f397bb38ac>>.
- GARCIA, G. B.; SUAREZ, O. D.; ARANDA, J. L. E.; TERCERO, J. S.; GRACIA, I. S.; ENANO, N. V. *Learning Image Processing with OpenCV*. [S.l.]: Packt Publishing, 2015.



GOOGLE, I. *Firestore*. 2014. Disponível em: <<https://firebase.google.com/>>.

GOOGLE, I. *Firestore Authentication*. 2014. Disponível em: <<https://firebase.google.com/products/auth>>.

GOOGLE, I. *Firestore Products*. 2014. Disponível em: <<https://firebase.google.com/products-build>>.

GOOGLE, I. *Firestore Realtime Database*. 2014. Disponível em: <<https://firebase.google.com/products/realtime-database>>.

GUEDES, G. T. A. *UML 2 - Guia de Consulta Rápida*. [S.l.]: Novatec, 2005.

GUIMARÃES, C. C. *Fundamentos de bancos de dados: modelagem, projeto e linguagem SQL*. Campinas: SP: UNICAMP, 2003.

JAVASCRIPT. 2021. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/JavaScript>>.

JIMÉNEZ, C. *PIB de 2020 no Brasil cai 4,1 resultado em 24 anos*. 2021. Disponível em: <<https://www.redebrasilatual.com.br/economia/2021/03/desempenho-economia-brasil>>.

JSON. *Introdução ao JSON*. 2021. Disponível em: <<https://www.json.org/json-pt.html>>.

KEMP, S. *DIGITAL 2021: GLOBAL OVERVIEW REPORT*. 2021. Disponível em: <<https://datareportal.com/reports/digital-2021-global-overview-report>>.

KHOROSHULIA, S. *HOW REACT NATIVE APP DEVELOPMENT WORKS UNDER THE HOOD*. 2021. Disponível em: <[https://mobidev.biz/blog/how-react-native-app-development-works#:~:text=React%20Native%20compiles%20the%20JavaScript,developers%20can%20create%20new%](https://mobidev.biz/blog/how-react-native-app-development-works#:~:text=React%20Native%20compiles%20the%20JavaScript,developers%20can%20create%20new%20)>.

LIFTOFF. *Relatório de apps mobile de finanças*. 2021. Disponível em: <<http://info.liftoff.io/pt-br/2020-mobile-finance-apps-report/>>.

MAHARANA, N. *Cross Platform Mobile Apps and Its Pros and Cons*. 2017. Disponível em: <<https://medium.com/andolasoft/cross-platform-mobile-apps-and-its-pros-and-cons-9c257ec64e94>>.

MENDONÇA, R. A. R. de. *Levantamento de requisitos no desenvolvimento ágil de software*. [S.l.]: Semana da Ciência e Tecnologia da PUC Goiás, 2014.

MOBILLS. 2021. Disponível em: <<https://www.mobills.com.br/>>.

MOZILLA. *Trabalhando com JSON*. 2021. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Objects/JSON>>.

OCCHINO, T. *React Native: Bringing modern web techniques to mobile*. 2015. Disponível em: <<https://engineering.fb.com/2015/03/26/android/react-native-bringing-modern-web-techniques-to-mobile/>>.

ORGANIZZE. *Organizze*. 2021. Disponível em: <<https://www.organizze.com.br/>>.

PALMIERI, M.; SINGH, I.; CICCETTI, A. Comparison of cross-platform mobile development tools. *16th International Conference on Intelligence in Next Generation Networks*, 2012.

PEDRINI, H.; SCHWARTZ, W. R. *Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações*. [S.l.]: Cengage Learning, 2007.

RAHIMIAN, V.; RAMSIN, R. Designing an agile methodology for mobile software development: A hybrid method engineering approach. *2008 Second International Conference on Research Challenges in Information Science*, 2008.

ROVEDA, U. *JavaScript: o que é, para que serve e como funciona o JS?* 2020. Disponível em: <<https://kenzie.com.br/blog/javascript/>>.

SADALAGE, P. J.; FOWLER, M. *NoSQL Essencial: Um guia conciso para o mundo emergente da persistência poliglota*. São Paulo: Novatec, 2015.

SPCBRASIL. *Cresce para 63finanças, revelam CNDL/SPC Brasil e Banco Central*. 2019. Disponível em: <<https://www.spcbrasil.org.br/pesquisas/pesquisa/5873>>.

STATS, G. *StatCunter*. 2021. Disponível em: <<https://gs.statcounter.com/os-market-share/mobile/worldwide/>>.

WHITE, J. Going native (or not): Five questions to ask mobile application developers. *Australasian Medical Journal*, 2013.

YEGULALP, S. *What is NoSQL? Databases for a cloud-scale future*. 2017. Disponível em: <<https://www.infoworld.com/article/3240644/what-is-nosql-databases-for-a-cloud-scale-future.html>>.