

**MINISTÉRIO DA EDUCAÇÃO  
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA INSTITUTO  
FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA FARROUPILHA  
CAMPUS PANAMBI**

**ROTA 66: APLICATIVO MOBILE REFORMULADO COM BASE NO  
DESIGN CENTRADO NO USUÁRIO**

**TRABALHO DE CONCLUSÃO DE CURSO**

**VICTOR AVILA CIECHOVICZ**

**Panambi, RS, Brasil  
2023**

# **ROTA 66: APLICATIVO MOBILE REFORMULADO COM BASE NO DESIGN CENTRADO NO USUÁRIO**

Por

VICTOR AVILA CIECHOVICZ

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Tecnólogo em Sistemas para Internet do Curso de Tecnologia em sistemas para Internet do Instituto Federal de Educação Ciência e Tecnologia Farroupilha - Campus Panambi

**Orientadora: Prof<sup>a</sup> Dra.Sirlei Rigodanzo**

PANAMBI  
2023

**ROTA 66: APLICATIVO MOBILE REFORMULADO COM BASE NO  
DESIGN CENTRADO NO USUÁRIO**

Trabalho de Conclusão de Curso  
apresentado como requisito parcial para  
obtenção do título de Tecnólogo em  
Sistemas para Internet do Curso de  
Tecnologia em sistemas para Internet do  
Instituto Federal de Educação Ciência e  
Tecnologia Farroupilha - Campus Panambi

Aprovado em:

**BANCA EXAMINADORA**

---

Sirlei Rigodanzo, Dr<sup>a</sup>.

(Orientadora)

---

Ederson Bastiani, Dr.

---

Everton Lutz, Dr.

## RESUMO

Este trabalho apresenta um modelo estrutural para a reestruturação do aplicativo "Rota 66", focando na melhoria da experiência do usuário final. Através deste modelo, busca-se enfatizar a importância do Design Centrado no Usuário, considerando o papel crucial que a interface do usuário desempenha na satisfação e retenção de usuários em aplicativos móveis. O trabalho também destaca a formação da usabilidade, centrada nos princípios de design, nos tópicos de estudo sobre a experiência do usuário e nas diretrizes recomendadas por especialistas em design de interfaces. O modelo de reestruturação do aplicativo – derivado de práticas centradas no usuário – serve como guia e referência para avaliar a eficácia e impacto da nova interface, seja em relação à facilidade de uso, à intuição, ou à personalização do aplicativo. O estudo é segmentado em vários capítulos que abordam respectivamente: revisão de literatura (Desafios dos Aplicativos Móveis; Princípios do Design Centrado no Usuário; Trajetória do Aplicativo Rota 66; Expectativas modernas dos usuários); coleta e análise de dados (Análise da interface atual do aplicativo Rota 66; Propostas e Prototipagem para a nova interface); contribuições do estudo; e as considerações finais sobre os resultados obtidos.

**Palavras-chave:** Experiência do usuário, Processo de desenvolvimento, Desenvolvimento móvel, Usabilidade.

## **ABSTRACT**

This work presents a structural model for the restructuring of the "Route 66" app, focusing on improving the end user's experience. Through this model, the importance of User-Centered Design is emphasized, considering the crucial role that the user interface plays in user satisfaction and retention in mobile applications. The work also highlights the formation of usability, centered on design principles, in topics of study about user experience, and in the guidelines recommended by interface design experts. The app restructuring model - derived from user-centered practices - serves as a guide and reference for assessing the effectiveness and impact of the new interface, whether in terms of ease of use, intuition, or app personalization. The study is segmented into several chapters that respectively address: literature review (Challenges of Mobile Applications; Principles of User-Centered Design; Trajectory of the Route 66 App; Modern User Expectations); data collection and analysis (Analysis of the current interface of the Route 66 app; Proposals and Prototyping for the new interface); contributions of the study; and the final considerations on the results obtained.

**Keywords:** User Experience, Development Process, Mobile Development, Usability.

## SUMÁRIO

<b>2 OBJETIVOS.....</b>	<b>10</b>
2.1 Objetivo Geral.....	10
2.2 Objetivos Específicos.....	10
<b>4 REFERENCIAL TEÓRICO.....</b>	<b>11</b>
4.1 Dispositivos móveis e aplicativos.....	12
4.1.1 Plataformas.....	12
<b>4.2 Desenvolvimento Mobile.....</b>	<b>12</b>
4.2.1 Mobile Web Apps.....	13
4.2.2 Aplicativos Nativos.....	13
4.2.3 Aplicações Híbridas.....	13
4.3 Desenvolvimento Híbrido / Multiplataforma.....	14
4.4 TypeScript.....	14
4.5 Frameworks versus Biblioteca.....	16
4.6 React.....	16
4.7 React Native.....	16
4.7.1 Arquitetura.....	17
4.7.2 Componentes React Native.....	19
4.8 Firebase.....	20
4.9 Psicologia das cores.....	21
<b>5 METODOLOGIA.....</b>	<b>22</b>
5.1 Design Centrado no Usuário (DCU).....	23
5.2 Escolha de Tecnologia.....	24
<b>6 O APLICATIVO.....</b>	<b>24</b>
<b>6.1 ROTA - 66: Aplicativo para estudo Bíblico.....</b>	<b>24</b>
6.1 Modelagem.....	25
6.1.1 Requisitos.....	26
6.1.2 Diagramas.....	26
6.2 Layout.....	27
6.3 Desenvolvimento.....	27
6.3.1 Ferramentas utilizadas para o desenvolvimento das melhorias do app Rota 66... 28	
6.3.2 Conexão com Firebase.....	29
6.3.2.1 Configuração Inicial do Firebase em React Native.....	29
6.3.2.2 Inicializando Firabese na Aplicação.....	30
6.3.2.3 Exemplo de uso Firestore.....	33
6.4 Resultados.....	34
6.4.1 Telas Públicas.....	34
6.4.2 Telas Autenticadas.....	36
6.4.3 Fluxo de Navegação.....	48
6.5 Coleta de Dados.....	49
6.5.1 Avaliação Geral.....	49

6.5.2 Localização de Recursos.....	51
6.5.3 Aprendizado e Curva de Aprendizado.....	52
Esses gráficos indicam que a nova interface é percebida como significativamente mais intuitiva do que a antiga. Isso implica que os usuários encontram a nova interface mais fácil de aprender e usar, o que é um indicador positivo para o design e a ergonomia da interface.....	53
6.5.4 Estética e Aparência.....	54
6.5.5 Preferência Geral.....	57
<b>CONCLUSÃO.....</b>	<b>58</b>
<b>REFERÊNCIAS.....</b>	<b>59</b>

## 1 INTRODUÇÃO

O Rota 66 é um aplicativo que disponibiliza e realiza a reprodução de estudos relacionados a um programa bíblico que foi transmitido pela Rádio Trans Mundial Dominicana<sup>1</sup>. Embora seja utilizado nos dias atuais e disponibilizado nas principais lojas, há a oportunidades de melhorar a experiência do usuário final para tornar o aplicativo mais fácil de usar, preciso e mais personalizado para as necessidades dos usuários.

Este trabalho propõe uma remodelagem do aplicativo com foco na melhoria da experiência do usuário final. Algumas das possíveis áreas de melhoria incluem:

1. **Interface do usuário:** A interface do usuário do aplicativo pode ser atualizada para tornar a navegação mais intuitiva e fácil de usar. As informações de navegação devem ser claramente apresentadas e a interface deve ser consistente em todas as telas.
2. **Facilidade de manuseio:** O aplicativo deve ser objetivo com interfaces que ajudem o usuário a encontrar o que deseja de forma mais organizada.
3. **Personalização:** O aplicativo deve proporcionar que o usuário consiga favoritar os áudios de sua preferência.

Com a remodelagem do aplicativo, antecipa-se uma maior adoção por parte dos usuários e uma elevada satisfação ao interagir com a interface renovada. Tendo em conta o conhecimento adquirido sobre os aplicativos existentes e as potenciais melhorias, é evidenciada uma problemática central: o aumento significativo de aplicativos móveis que, por negligenciar as expectativas e necessidades dos usuários, proporcionam uma experiência subpar, culminando no conseqüente abandono do software. (NIELSEN 2005)

---

<sup>1</sup> Rádio Trans Mundial é uma rede de rádio brasileira sediada em São Paulo, capital do estado homônimo. É ligada à Trans World Radio, que tem sede nos Estados Unidos, e foi inaugurada em 5 de fevereiro de 1970. Os programas da rede são distribuídos via satélite para diversas estações de rádio afiliadas no Brasil. Disponível em: [https://pt.wikipedia.org/wiki/R%C3%A1dio\\_Trans\\_Mundial](https://pt.wikipedia.org/wiki/R%C3%A1dio_Trans_Mundial)  
Acesso em: 22/09/2023.

De acordo com Nielsen (2005), o Design Centrado no Usuário surge como uma prática indispensável que privilegia a interação e as exigências do utilizador para gerar uma experiência de excelência. Assim, a questão essencial a ser explorada centra-se na reformulação das interfaces de aplicativos móveis com o intuito de potencializar a usabilidade e experiência do usuário, recorrendo à metodologia do Design Centrado no Usuário. Inúmeros softwares móveis falham ao não corresponderem às expectativas dos usuários, precipitando uma experiência insatisfatória e eventual desistência.

Para abordar tal problemática, alguns questionamentos se tornam pertinentes:

- Como estruturar eficazmente a arquitetura da informação?
- Como conceber interfaces mais instintivas?
- Como assegurar acessibilidade universal no aplicativo?

A resolução desses questionamentos poderá guiar uma reformulação bem-sucedida, ancorada nos princípios do Design Centrado no Usuário, alavancando a usabilidade e a satisfação do usuário. Este aprimoramento pode atingir um engajamento mais robusto, impulsionando o sucesso do aplicativo.

Em concordância com as perguntas propostas, algumas hipóteses emergem. Baseando-se nos princípios de Nielsen (2005), postula-se que ao implementar o Design Centrado no Usuário e ao renovar a interface, será possível proporcionar uma experiência mais enriquecedora aos usuários. Esta potencial melhoria pode refletir-se em uma maior retenção de usuários, mais avaliações positivas nas plataformas de distribuição e uma expansão no número de adesões. Consequentemente, o aplicativo reformulado aspira a atender de forma mais eficaz às demandas dos usuários, consolidando-se como uma escolha competitiva no panorama dos aplicativos móveis.

## **2 OBJETIVOS**

O objetivo deste projeto é reestruturar a interface do aplicativo, trazer uma melhor usabilidade e experiência para o usuário final. Para alcançar esse objetivo, será necessário realizar uma análise detalhada do aplicativo atual e identificar pontos de dor dos usuários através de questionários. Com base nessas informações, serão desenvolvidos protótipos que permitam testes com usuários para avaliar a usabilidade e a experiência do usuário. Serão realizadas pesquisas de usabilidade para avaliar a eficácia das soluções propostas e aprimorá-las.

### **2.1 Objetivo Geral**

Avaliar e otimizar o aplicativo para aprimorar a experiência do usuário.

### **2.2 Objetivos Específicos**

- a) Avaliar detalhadamente o aplicativo em sua forma atual para identificar seus pontos fortes e fracos.
- b) Identificar as principais dificuldades dos usuários ao interagirem com o aplicativo.
- c) Desenvolver uma interface otimizada, com foco na usabilidade e na experiência do usuário.
- d) Analisar o impacto das inovações implementadas na melhoria da experiência do usuário.

### 3 JUSTIFICATIVA

A necessidade de reestruturação do aplicativo com foco na melhoria da experiência do usuário surge por diversas razões. Em um mercado de aplicativos móveis altamente saturado e competitivo, os usuários possuem um vasto leque de opções. Nesse contexto, avaliações negativas nas lojas de aplicativos, especialmente aquelas que apontam para uma usabilidade inadequada, podem ser muito negativas para a reputação e o sucesso do aplicativo. A seguir temos um feedback na qual um usuário relata sua principal dificuldade: "Gostaria de sugerir a opção de aumentar a velocidade de reprodução é um índice com mais opções (por exemplo: estou estudando o livro de provérbios e toda vez é uma dificuldade em chegar até ele e começar o estudo, preciso rolar várias vezes a página e ficar procurando). O aplicativo é fantástico, obrigado. Gostaria" (ROTA 66 - APPLE STORE)

O Rota 66, sendo um aplicativo de reprodução de áudios, carrega a responsabilidade de entregar conteúdo de qualidade. Assim, é imperativo que sua interface seja fácil de usar, intuitiva e confiável. Desconsiderar a experiência do usuário pode resultar em frustração e, eventualmente, na perda de usuários para concorrentes que oferecem uma experiência mais refinada.

A abordagem do Design Centrado no Usuário, proposta para a reestruturação, coloca o usuário no centro do processo de design. Isso não apenas resultará em interfaces mais intuitivas e amigáveis, mas também elevará a satisfação geral do usuário, fortalecendo a marca e aumentando a retenção. Dado o feedback negativo atual nas lojas de aplicativos, essa reestruturação é não apenas desejável, mas essencial para a posição do aplicativo no mercado e proporcionar uma experiência excepcional aos seus usuários.

## 4 REFERENCIAL TEÓRICO

Este capítulo fornece uma base teórica para a reestruturação do aplicativo móvel Rota 66, focando na experiência do usuário. Abordaremos tecnologias e frameworks cruciais como React Native Framework para desenvolvimento multiplataforma, TypeScript para legibilidade do código e Firebase para gerenciamento de dados.

### 4.1 Dispositivos móveis e aplicativos

Dispositivos móveis são tecnologias digitais que permitem a mobilidade e o acesso à internet, captura de fotos e vídeos, e execução de aplicativos. Eles geralmente têm a capacidade de se conectar a redes sem fio, como Wi-Fi e redes móveis, o que os torna ideais para uso em trânsito. Exemplos comuns incluem smartphones, tablets e smartwatches. (BOCARD, 2023)

O texto a seguir discute as principais plataformas de dispositivos móveis e aspectos do desenvolvimento mobile.

#### 4.1.1 Plataformas

Plataformas de dispositivos móveis referem-se aos sistemas operacionais que gerenciam o hardware e permitem o funcionamento de software em dispositivos móveis, como smartphones e tablets. Essas plataformas são o núcleo que possibilita a execução de diversas atividades, desde chamadas telefônicas e mensagens de texto até o uso de aplicativos e navegação na internet.

As mais proeminentes são o Android, desenvolvido pelo Google, e o iOS, desenvolvido pela Apple.

- **Android:** Desenvolvido pelo Google, o Android é um sistema operacional de código aberto que é usado em uma variedade de dispositivos de diferentes fabricantes, como Samsung, Google Pixel, LG, Sony e outros. A Play Store é a loja oficial de aplicativos do Android, e o sistema permite um grau relativamente alto de personalização (GOOGLE PLAY, 2023)

- **iOS:** Este é o sistema operacional desenvolvido pela Apple e é usado exclusivamente em dispositivos Apple, como iPhones e iPads. O iOS é conhecido por sua interface de usuário polida e seu ecossistema fechado, o que significa que todos os aplicativos devem ser baixados através da App Store da Apple (APPLE STORE., 2023).

## **4.2 Desenvolvimento Mobile**

O desenvolvimento de aplicações móveis é um campo crescente que atende a uma variedade de setores e necessidades do usuário. Conforme descrito no texto do site, esta área envolve a criação de software para dispositivos móveis, como smartphones e tablets, com foco principalmente nos sistemas operacionais Android e iOS. Para atender às diversas necessidades e limitações dos dispositivos móveis e plataformas, existem três abordagens principais de desenvolvimento: mobile web apps, aplicativos nativos e aplicações híbridas (IBM, 2023).

### **4.2.1 Mobile Web Apps**

Mobile web apps são aplicações acessadas via navegador da web em um dispositivo móvel. Eles são construídos usando tecnologias web padrão como HTML5, CSS e JavaScript. Uma vantagem dessa abordagem é que ela é relativamente mais fácil de manter e atualizar, pois as mudanças são feitas em um único código-fonte. No entanto, essas aplicações podem não oferecer uma experiência de usuário tão rica quanto os aplicativos nativos, e algumas funcionalidades podem ser limitadas, pois elas não têm acesso direto às APIs do sistema operacional do dispositivo. (MOURA, C. Z 2020)

### **4.2.2 Aplicativos Nativos**

Aplicativos nativos são desenvolvidos para uma plataforma específica, usando as linguagens de programação e ferramentas recomendadas por essa plataforma. Por exemplo, Swift para iOS e Kotlin para Android. Estes aplicativos têm a vantagem de serem mais rápidos e eficientes, com acesso direto às funcionalidades do dispositivo, como câmera e GPS. No entanto, o desenvolvimento

nativo pode ser mais caro e demorado, já que é necessário criar e manter versões separadas para cada sistema operacional. (MOURA, C. Z 2020)

### **4.2.3 Aplicações Híbridas**

Aplicações híbridas buscam combinar o melhor dos mundos nativo e web. Elas são construídas usando tecnologias web, mas são encapsuladas dentro de um "container" nativo que permite acesso às funcionalidades do dispositivo. Isso permite que os desenvolvedores mantenham um único código-fonte que pode ser usado em várias plataformas. Frameworks como React Native e Flutter são frequentemente usados para esse tipo de desenvolvimento. A desvantagem é que o desempenho pode ser comprometido, já que essas aplicações ainda dependem de uma "camada" para acessar as funcionalidades nativas. (MOURA, C. Z 2020)

### **4.3 Desenvolvimento Híbrido / Multiplataforma**

Com o avanço da tecnologia e a necessidade de desenvolvimento de dispositivos móveis de uma forma rápida e prática, surgiram algumas ferramentas que permitem aos desenvolvedores criar aplicativos móveis que funcionem em vários sistemas operacionais a partir de um único código-fonte. A ideia é produzir uma aplicação ou produto que funcione em mais de um ambiente específico. Esta abordagem tem a vantagem de economizar tempo e recursos. (MOURA, C. Z 2020)

### **4.4 TypeScript**

TypeScript é uma extensão tipada de “*JavaScript*”<sup>2</sup>, desenvolvida pela Microsoft, que adiciona um sistema de tipagem estática opcional à linguagem original. Em outras palavras, TypeScript é um superconjunto de JavaScript: todo código JavaScript é automaticamente um código TypeScript válido. A adição de tipos estáticos torna o desenvolvimento mais previsível e auxilia na detecção de erros, o

---

<sup>2</sup> JavaScript é uma linguagem de programação de alto nível, dinâmica e interpretada, amplamente utilizada para desenvolvimento web do lado cliente, mas também utilizada em servidores através do Node.js. Originalmente criada por Brendan Eich em 1995, a linguagem agora é padronizada como ECMAScript e desenvolvida pelo consórcio ECMA.

que é especialmente útil em projetos de grande escala. Ao ser compilado, o código TypeScript é transformado em JavaScript puro, permitindo a execução em qualquer ambiente que suporte JavaScript. (TYPESCRIPT 2023)

"TypeScript é um superconjunto tipado de JavaScript que compila para JavaScript simples." Documentação Oficial do TypeScript (TYPESCRIPT, 2023)

Abaixo na Figura 1 está um exemplo do funcionamento das as diferentes etapas onde cada tecnologia atua.

Figura 1- Processo ilustrativo do funcionamento do TypeScript



Fonte: React Native: o que é e tudo sobre o Framework. Disponível em: <https://www.alura.com.br/artigos/react-native>. Acesso em: 02 set. 2023.

Os recursos que o TypeScript oferece:

- 1) **Tipagem Estática:** Permite especificar o tipo de variável no momento da declaração, ajudando a capturar erros mais cedo.
- 2) **Interface:** Oferece uma maneira de definir contratos dentro do seu código e entre seu código e código externo.
- 3) **Classes e Objetos:** Incorpora recursos de orientação a objetos, como classes, interfaces e herança.
- 4) **Genéricos:** Fornece uma maneira de criar componentes que podem trabalhar com vários tipos, ao invés de um único tipo.

- 5) **Configuração Flexível:** Possui um arquivo de configuração ("*tsconfig.json*"<sup>3</sup>) que permite ajustar o comportamento do compilador TypeScript.
- 6) **Downlevel Compilation:** Compila para várias versões de JavaScript, garantindo a compatibilidade com browsers mais antigos (TYPESCRIPT, 2023).

#### 4.5 Frameworks versus Biblioteca

Frameworks e bibliotecas são duas abordagens de desenvolvimento de software que fornecem um conjunto de funcionalidades para ajudar os desenvolvedores a criar aplicações mais rapidamente e eficientemente. Embora ambos sejam úteis, eles têm diferenças fundamentais que afetam como os desenvolvedores interagem com eles. (MOURA, C. Z 2020)

Frameworks fornecem uma estrutura geral que guia o fluxo de controle da aplicação, muitas vezes seguindo o princípio de "convenção sobre configuração" (Hansson, 2008), enquanto bibliotecas oferecem um conjunto de funcionalidades específicas que os desenvolvedores podem chamar conforme necessário, mantendo o controle do fluxo da aplicação (Wulf, 1981).

#### 4.6 React

Desenvolvida pelo Facebook e lançada em 2013, a biblioteca JavaScript React tem como objetivo a construção de interfaces de usuário de forma eficiente e modular (REACT, 2023). Utilizando um modelo de componente, o React torna a criação de UIs complexas mais acessível e o código mais reutilizável (Abramov, 2019). Além disso, o React se beneficia de uma forte comunidade de desenvolvedores e de um vasto ecossistema de bibliotecas complementares.

Uma das principais vantagens do React é a sua virtual DOM (Document Object Model), que otimiza o desempenho ao minimizar operações de atualização direta no DOM real. Isso permite que as aplicações React sejam altamente dinâmicas e responsivas, garantindo uma experiência de usuário suave. Além disso, a facilidade de integração com diversas APIs e outras bibliotecas JavaScript amplia

---

<sup>3</sup> O arquivo *tsconfig.json* é utilizado em projetos TypeScript para configurar opções do compilador. Ele especifica coisas como a versão do JavaScript a ser usada, quais arquivos incluir ou excluir na compilação, e outras preferências. É essencialmente o guia de configuração para o compilador TypeScript saber como processar seu código.

as capacidades do React, tornando-o uma escolha popular para o desenvolvimento de single-page applications e interfaces de usuário interativas em várias plataformas (REACT, 2023).

#### **4.7 React Native**

O React Native é um framework de desenvolvimento de aplicativos móveis que permite aos desenvolvedores usar JavaScript ou TypeScript e React para criar aplicações nativas para iOS e Android (REACT NATIVE, 2023). Este framework, que também foi desenvolvido pelo Facebook, difere do React tradicional ao oferecer componentes que se traduzem para views nativas em vez de HTML. (Starczewski, 2019)

Isso torna possível desenvolver aplicativos com desempenho comparável ao de aplicações nativas construídas com Swift ou Java (Nader, 2020). Ele também suporta a integração com código nativo, oferecendo uma via para funcionalidades que exigem acesso direto às APIs nativas. (Axel, 2017)

##### **4.7.1 Arquitetura**

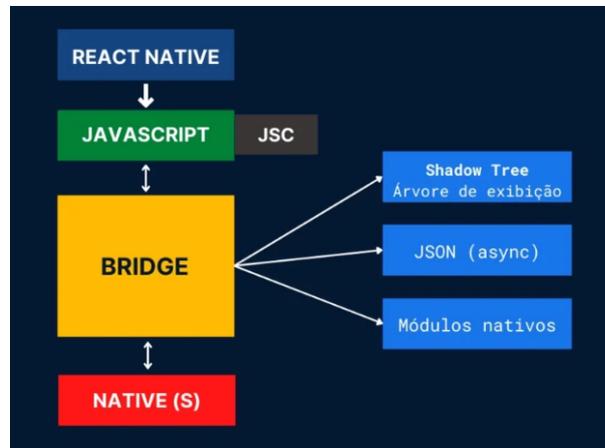
A arquitetura do React Native é uma mistura de componentes nativos e JavaScript. Ele usa uma ponte para facilitar a comunicação entre os dois. Isso permite que o código JavaScript controle componentes nativos, resultando em uma experiência de usuário similar a um aplicativo nativo. (ALURA, 2023)

A arquitetura do React Native pode ser dividida em 4 seções principais:

- O código React, em que o desenvolvedor escreve o App;
- O JavaScript que é eventualmente interpretado a partir do código que a pessoa desenvolvedora escreveu;
- Uma série de elementos conhecidos como The Bridge;
- O lado nativo (REACT NATIVE, 2023).

Figura 2 mostra de maneira visual como a arquitetura do React Native é dividida.

Figura 2 - A Arquitetura do React Native



Fonte: React Native: o que é e tudo sobre o Framework. Disponível em: <https://www.alura.com.br/artigos/react-native>. Acesso em: 02 set. 2023.

## JavaScript

Nessa parte, o código é executado em um mecanismo JavaScript. O React Native usa score, que é um mecanismo JavaScript de código aberto para WebKit. Ele é executado dentro de nosso aplicativo em uma das threads, sendo que um aplicativo móvel possui várias threads - e o javascript é uma delas. (ALURA, 2023)

## Na parte nativa

No lado nativo, as pessoas desenvolvedoras desenvolvem em Object Swift, no caso do iOS, ou com Java/Kotlin se for para Android. Usaremos as linguagens nativas específicas da plataforma que usamos antes, e a thread principal da interface do usuário estará disponível, como de costume.

Em todas as plataformas, só precisamos alterar a interface do usuário da thread principal e poderemos criar diferentes threads em segundo plano, conforme necessário. (ALURA, 2023)

## The Bridge

O the bridge, ou então a "ponte" do React Native, é uma forma de comunicar o nosso código JavaScript com a parte nativa (o código do Android ou iOS).

Ele funciona como se fosse o coração do React Native, se comunicando com os dois lados de forma assíncrona. Basicamente, é assim que acontece:

1. Escrevemos o código de um componente botão;
2. O JavaScriptCore vai compilar o código do componente e enviar para a bridge;
3. A bridge, por sua vez, vai pegar esse código compilado e enviar para a parte nativa, para que possa ser exibido o botão;
4. O nativo entende e monta o componente do botão em tela;
5. Se houver uma interação no botão, ou seja, se for clicado, esse evento acontece primeiro no lado nativo, que vai para o bridge, e depois para a parte em que foi programada em JavaScript para executar alguma função, por exemplo (REACT NATIVE, 2023)

#### **4.7.2 Componentes React Native**

Os componentes em React Native são as unidades fundamentais de código que compõem a interface do usuário (UI) de um aplicativo. Eles são semelhantes aos componentes React para desenvolvimento web, mas foram adaptados para o ambiente móvel. Esses componentes incluem elementos básicos como View, que atua como um contêiner para outros componentes, Text para exibição de texto, Image para renderização de imagens, e muitos outros (Axel, 2017). A reutilização de componentes torna o desenvolvimento mais eficiente e mantém o código mais organizado.

#### **Componentes Básicos**

No contexto do desenvolvimento com React Native, os componentes representam os elementos fundamentais que compõem a interface de usuário de um aplicativo. Eles são modularizados e reutilizáveis, permitindo aos desenvolvedores construir interfaces de usuário complexas de maneira organizada e eficiente. Cada componente possui atributos e comportamentos específicos, sendo responsável por

renderizar uma parte da UI (W3SCHOOLS, 2023). A seguir, descrevemos alguns componentes básicos essenciais no React Native.

**View:** O componente `<View>` atua como um contêiner, semelhante ao elemento `<div>` em HTML utilizado no desenvolvimento web com React. É um componente versátil que suporta estilo e layout flexível, servindo como base para outros componentes da interface de usuário.

**Text:** O componente `<Text>` é utilizado para exibir texto na tela. É equivalente aos elementos HTML `<p>` ou `<span>`, oferecendo várias opções para personalização e estilização do texto apresentado.

**Image:** Este componente é empregado para renderizar imagens na interface de usuário, comparável ao elemento HTML `<img>`. Permite a exibição de imagens de diferentes formatos e origens, com opções para ajustar propriedades como dimensão e estilo.

**ScrollView:** O `<ScrollView>` facilita a rolagem do conteúdo na tela, assemelhando-se ao comportamento de uma `<div>` com rolagem no desenvolvimento web com React. É uma solução eficaz para a apresentação de conteúdos que excedem as dimensões da tela.

**TextInput:** Este componente proporciona um campo para entrada de texto pelo usuário, similar ao elemento HTML `<input>`. É amplamente utilizado para capturar dados do usuário, como texto e números, oferecendo opções para validação e manipulação desses dados.

**Button:** O componente `<Button>` é usado para criar botões clicáveis na interface de usuário, equivalente ao elemento HTML `<button>`. Ele é essencial para facilitar a interatividade e a execução de ações específicas no aplicativo (REACT NATIVE, 2023).

## 4.8 Firebase

O Firebase é uma plataforma de desenvolvimento de aplicativos móveis e web desenvolvida pela Google. Desde seu lançamento em 2012 e subsequente aquisição pela Google em 2014, ele se tornou uma das soluções mais populares e abrangentes para desenvolvedores. A plataforma oferece uma variedade de ferramentas e serviços que simplificam tarefas complexas em desenvolvimento, como autenticação de usuários, armazenamento de dados, e escalonamento de aplicações (Google Inc., 2016).

Para o desenvolvimento de aplicativos móveis, a integração do Firebase com React Native é especialmente eficaz, oferecendo uma série de bibliotecas e SDKs que facilitam a implementação de funcionalidades complexas com menos esforço e mais eficiência (Newman, 2021).

### Funcionalidades Principais

- Autenticação: Firebase oferece diversos métodos de autenticação como e-mail/senha, OAuth, e até mesmo autenticação por telefone.
- Realtime Database: Firebase Realtime Database permite o armazenamento e sincronização de dados em tempo real.
- Storage: O Firebase também oferece armazenamento de arquivos como imagens e vídeos (FIREBASE, 2023).

Tabela 2 - Funcionalidades do Firebase utilizadas

Firestore	React Native Firestore	Descrição
Realtime Database	firebase.database()	Banco de dados em tempo real
Authentication	firebase.auth()	Autenticação de usuários
Storage	firebase.storage()	Armazenamento de arquivos

## 4.9 Psicologia das cores

Para realizar a prototipação das interfaces foi levado em consideração a psicologia das cores. Com esta abordagem, além de tornar mais organizada a listagem dos áudios tornando as cores das interfaces mais intuitivas.

Figura 4 - Paleta de cores predominantes



Fonte: Elaborado pelo autor

Aqui estão as cores predominantes extraídas da interface. Vou discutir a psicologia associada a cada uma delas:

**Escala de Cinza:** As cores cinza tendem a ser associadas à neutralidade, equilíbrio e profissionalismo. Elas podem ser percebidas como modernas e sofisticadas. No contexto de um aplicativo bíblico, o cinza pode ser usado para criar um fundo neutro, permitindo que o conteúdo e as informações se destaquem.

**Escala de Azul:** O azul é uma cor associada à confiança, serenidade e sabedoria. É uma cor comum em ambientes corporativos e educacionais devido à sua conotação de estabilidade e confiabilidade. Para um aplicativo bíblico, o azul pode ser usado para evocar sentimentos de paz, reflexão e espiritualidade.

**Gradientes:** Gradientes são transições suaves entre duas ou mais cores. No design de interface, eles podem adicionar profundidade e um senso de dinamismo. O uso de azul nos gradientes, neste contexto, reforça a sensação de confiabilidade e serenidade.

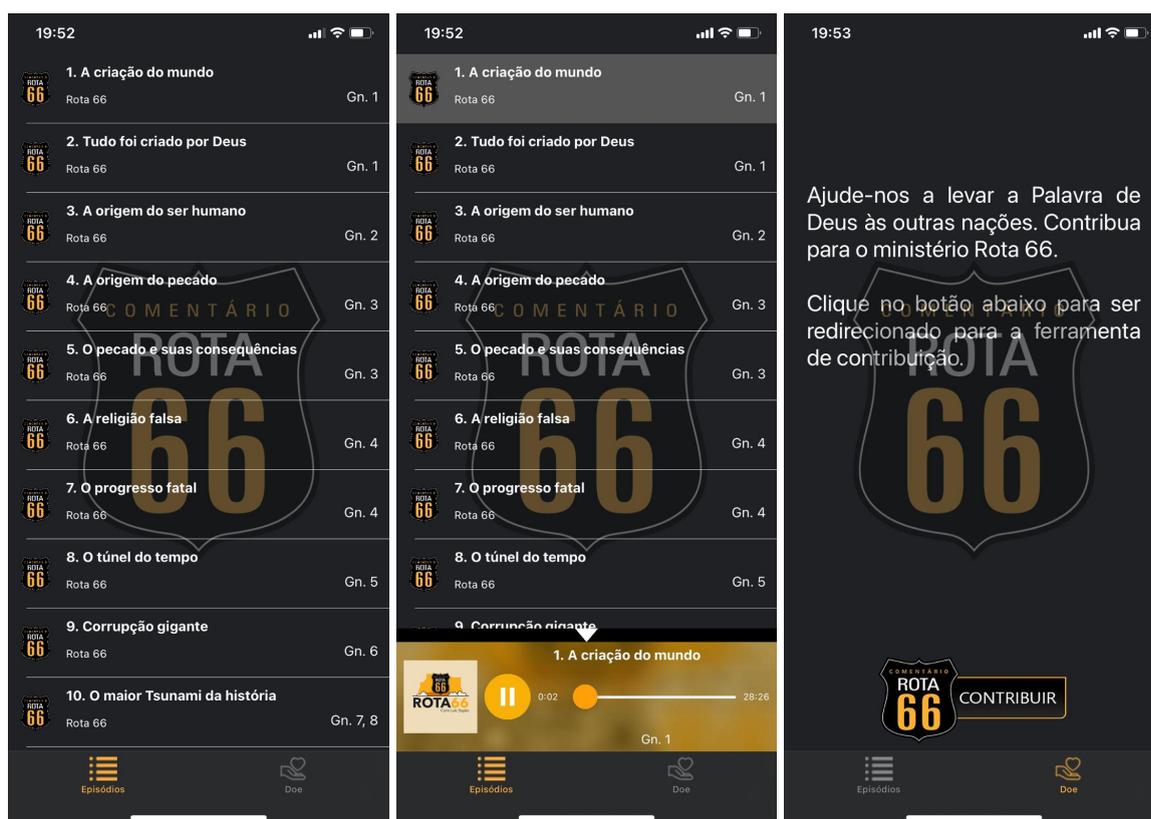
Cada cor na interface tem o potencial de evocar certos sentimentos ou ações do usuário. No design de interfaces, a escolha da cor é crucial não apenas para a estética, mas também para a funcionalidade e experiência do usuário. O esquema de cores desta interface parece bem equilibrado, combinando cores neutras com cores mais vibrantes para chamar a atenção quando necessário.

## 5 METODOLOGIA

O desenvolvimento de aplicativos móveis com foco na experiência do usuário tem se mostrado crucial para o sucesso dessas plataformas (Smith, 2021). O aplicativo 'Rota 66' apresentou diversas oportunidades para melhorias na usabilidade e na interface do usuário. Para abordar essas questões de forma sistemática, este trabalho adotou o Design Centrado no Usuário (DCU) como metodologia.

Conforme podemos visualizar na Figura 3 temos a aplicação na versão antiga com apenas duas telas principais e os áudios distribuídos em uma única listagem, isso deixa a desejar quando vamos falar sobre usabilidade e Design Centrado no Usuário (DCU) .

Figura 3 - Telas atuais do aplicativo Rota 66 (VERSÃO ANTIGA)



Fonte: Loja de aplicativos Google Play. Disponível em:

[https://play.google.com/store/apps/details?id=solutions.hubner.rota66&hl=pt\\_BR&gl=US](https://play.google.com/store/apps/details?id=solutions.hubner.rota66&hl=pt_BR&gl=US)

Acesso em: 07/09/2023.

Para o desenvolvimento do aplicativo, foi utilizado a linguagem de programação React Native, que permite criar aplicativos para os sistemas operacionais Android e IOS, sem a necessidade de reescrever um código separado nativo para cada um dos sistemas. Também foi utilizado o Firebase como banco de dados.

### **5.1 Design Centrado no Usuário (DCU)**

A metodologia de DCU foi implementada seguindo as etapas de pesquisa de usuário, desenvolvimento de protótipos, testes de usabilidade e interação através da coleta dos dados. Essa abordagem assegura que as necessidades e expectativas dos usuários sejam atendidas de forma eficaz.

### **5.2 Escolha de Tecnologia**

A linguagem de programação JavaScript, juntamente com o framework React Native, foi escolhida por permitir um desenvolvimento ágil e eficiente. A natureza multiplataforma do React Native também foi um fator crucial na escolha da tecnologia. Além da facilidade de implementação com Firebase como banco de dados, assim deixando o aplicativo mais segura com todos os recursos dispostos pela tecnologia.

## **6 O APLICATIVO**

### **6.1 ROTA - 66: Aplicativo para estudo Bíblico**

O Rota 66 é uma aplicativo que já está em funcionamento nos dias atuais nas principais lojas de aplicativos *Google Play* e *Apple Store*, onde sua proposta é trazer comentário bíblico através de áudios, gravados pelo pastor Luiz Alberto Teixeira Sayão e transmitidos pela Rádio Trans Mundial Dominicana . A trilha dos áudios é um caminho pelos 66 livros da Bíblia de maneira não sequencial, para que haja uma compreensão do texto bíblico em sua totalidade. Além disso, utiliza-se de expressões da cultura brasileira com bastante humor, como piadas e frases de

efeito, para fazer ponte com o texto bíblico, visando torná-lo mais simples e acessível.

A construção inicial deste aplicativo que está em funcionamento nos dias atuais não levou em consideração muitos fatores de organização ou construção de uma interface intuitiva, nem organização para tornar a usabilidade deste aplicativo boa para o usuário, gerando assim alguns problemas que são relatados pelos usuários nos comentários de avaliação do aplicativo. Como podemos observar, este é um exemplo de prática inadequada de usabilidade, toda a listagem dos áudios é feita através de uma única lista (Figura 3) gerando um fluxo muito oneroso ao usuário procurar o áudio que deseja escutar.

A partir dessa constatação e pensando em inovar um grupo de representantes da Rádio Trans Mundial Dominicana criaram uma iniciativa de projeto voluntário para a remodelagem deste aplicativo, com o objetivo de melhorar tanto a questão da interface ser mais intuitiva ao usuário quanto de aplicar tecnologias mais recentes de código para a construção do mesmo.

Assim sendo, iniciamos a descrição do processo de prototipagem para o aplicativo Rota 66, que se segmenta em quatro componentes críticos: a modelagem, que inclui requisitos e diagramas; a configuração do layout; o desenvolvimento, abarcando os processos e as ferramentas empregadas, e finalmente, os resultados, representados pelas interfaces de usuário.

## **6.1 Modelagem**

Nesta subseção, empregando os princípios de engenharia de requisitos, delineamos as funcionalidades do aplicativo com base em requisitos funcionais extraídos da descrição inicial apresentada na introdução. A enumeração desses requisitos é crucial para o desenvolvimento inicial das interfaces do aplicativo. Adicionalmente, os diagramas de caso de uso serão fornecidos para oferecer uma visão abrangente da solução proposta.

### **6.1.1 Requisitos**

Os requisitos listados a seguir surgiram a partir da ideia principal das melhorias que o aplicativo e do ele poderia ser futuramente. Diante disso, conclui-se que a aplicação deve permitir:

1 . Que um usuário se cadastre:

- a) O Cadastro para usuários não deve ser muito burocrático, apenas exigido dos básicos de acesso e necessidade para o login, como: Nome, email e senha.

2 . Ao usuário já cadastrado:

- a) Realizar login.
- b) Inserir, alterar nome do perfil.
- c) Alterar senha de acesso a conta.
- d) Acesso aos áudios.
- e) Favoritar livros e estudos

3. A listagem de áudios será feita através dos capítulos que o mesmo pertencer.

4. O acesso a listagem de áudios só é realizada posteriormente a um cadastro do usuário na aplicação.

5. Listagens de estudos ou livros favoritos

6. Ao Administrador

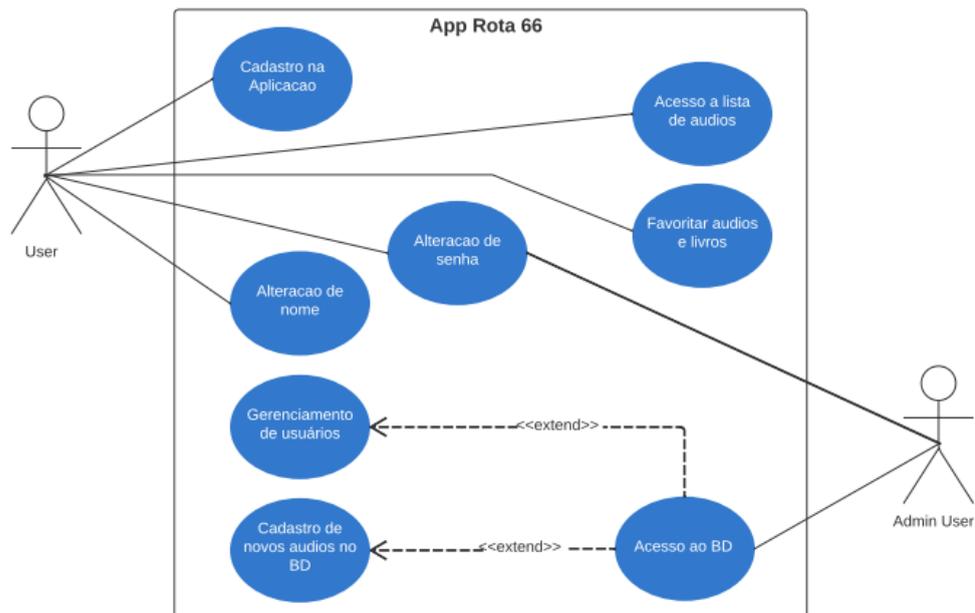
- a) Acesso ao banco de dados, através da plataforma disponível do Firebase.
- b) Cadastro de novos áudios, através da plataforma disponível do Firebase.
- c) Gerenciamento de usuários, através da plataforma disponível do Firebase.
- d) Alteração de senha, através da plataforma disponível do Firebase.

### **6.1.2 Diagramas**

Após a validação dos requisitos e para obter uma visão abrangente da questão em foco, tornou-se imperativo desenvolver um diagrama de caso de uso. Esse diagrama tem como objetivo identificar os atores participantes, mapear suas interações com o software e estruturar a aplicação em termos de níveis de acesso e funcionalidades.

O diagrama da Figura apresenta todas as funcionalidades, listadas anteriormente na seção 6.1.1.

Figura 5 - Caso de Uso



Fonte: Elaborado pelo autor

## 6.2 Layout

O layout foi inicialmente criado utilizando apenas componentes do próprio React Native. Serviram de inspiração alguns aplicativos como CastBox.

## 6.3 Desenvolvimento

Esta seção aborda os aspectos técnicos por trás do desenvolvimento do aplicativo, desde a arquitetura até a estilização. A arquitetura do aplicativo foca na disponibilização de conteúdo em áudio.

### 6.3.1 Ferramentas utilizadas para o desenvolvimento das melhorias do app Rota 66

Para a elaboração dos diagramas, optamos pelo Lucidchart, uma ferramenta online versátil e fácil de usar.

Para o desenvolvimento de código, utilizamos o Visual Studio Code, um editor fornecido pela Microsoft que é compatível com Windows, Linux e MacOS. O editor oferece funcionalidades como depuração, controle Git embutido, realce de sintaxe e autocompletar de código inteligente.

O ambiente de desenvolvimento foi configurado após instalar as dependências necessárias e inicializar o projeto utilizando o comando **`npx react-native@latest init Rota 66`**.

A seguir, listamos os módulos adicionais instalados via NPM, juntamente com uma breve descrição de sua utilidade:

- a) **React Navigation:** Fornece a capacidade de navegar entre diferentes telas na aplicação.
- b) **React Native SVG:** Permite a manipulação e renderização de gráficos SVG no aplicativo.
- c) **Expo-AV:** Facilita a reprodução e manipulação de áudio e vídeo.
- d) **Expo Google Fonts:** Facilita a incorporação de uma variedade de fontes do Google.
- e) **AsyncStorage:** Oferece um mecanismo de armazenamento local assíncrono.
- f) **Animation Lottie:** Para integração de animações complexas de forma eficiente.
- g) **Firebase Cloud Firestore:** Utilizado para armazenamento e consulta de dados em tempo real.
- h) **Firebase Cloud Auth:** Gerência autenticação de usuários.
- i) **Firebase Storage:** Permite o armazenamento de arquivos, como imagens e áudios, na nuvem.

Com cada nova implementação, tornou-se crucial gerenciar tanto o versionamento do código quanto às tarefas pendentes no aplicativo. Para facilitar

esse processo, utilizamos o GIT como sistema de controle de versão e o GitHub como plataforma de hospedagem do repositório de código.

### 6.3.2 Conexão com Firebase

O Firebase, uma plataforma de desenvolvimento oferecida pela Google, foi escolhido como o backend para nosso aplicativo React Native devido à sua escalabilidade, eficiência e fácil integração. O Firebase oferece diversas soluções, mas neste projeto focamos principalmente na utilização do Firestore, sua oferta de banco de dados NoSQL, e do sistema de autenticação.(FIREBASE, 2023)

#### 6.3.2.1 Configuração Inicial do Firebase em React Native

1. *Criar um projeto Firebase:* Se você ainda não tem um, vá ao Firebase Console e crie um novo projeto.
2. *Registrar seu aplicativo:*

- Clique no ícone para adicionar um aplicativo e selecione a plataforma do aplicativo (iOS ou Android).
- Siga as instruções para registrar seu aplicativo. Você receberá um arquivo de configuração: google-services.json para Android e GoogleService-Info.plist para iOS. (FIREBASE, 2023)

3. *Instalar a biblioteca do Firebase:*

Como mencionado anteriormente, o primeiro passo é instalar o pacote principal: **npm install @react-native-firebase/app** (FIREBASE, 2023)

4. *Adicionar arquivo de configuração:*

- Para Android: Coloque o arquivo google-services.json dentro da pasta android/app do seu projeto React Native.
- Para iOS: Abra seu projeto no Xcode, depois arraste e solte o arquivo GoogleService-Info.plist em seu projeto. Certifique-se de adicionar ao target da sua aplicação.(FIREBASE, 2023)

5. *Configurar o Firebase para Android:*

- No arquivo android/build.gradle, adicione:

```
buildscript {
    dependencies {
        // ...
        classpath 'com.google.gms:google-services:4.3.3'
    }
}
```

- E no android/app/build.gradle adicione ao final:

```
apply plugin: 'com.google.gms.google-services'
```

6. *Configurar o Firebase para iOS (se estiver desenvolvendo para iOS):*

- Certifique-se de que você tenha o CocoaPods instalado.
- No diretório do seu projeto, navegue até a pasta ios/ e execute pod install. (FIREBASE, 2023)

### **6.3.2.2 Inicializando Firabese na Aplicação**

Para termos acesso às funcionalidades desta ferramenta temos que realizar uma configuração para inicializar o Firebase em um projeto, permitindo que o projeto se comunique com os serviços do Firebase como autenticação (Auth) e Firestore (base de dados). Vamos entender cada parte:

- Importação de módulos:

```
import { initializeApp } from "firebase/app";
```

```
import { getAuth } from "firebase/auth";
```

```
import { getFirestore } from "firebase/firestore";
```

Estas linhas importam as funções necessárias para inicializar e interagir com o Firebase. Especificamente, initializeApp é usado para iniciar a conexão com o Firebase usando as configurações do projeto, enquanto getAuth e getFirestore são utilizados para acessar os serviços de autenticação e Firestore, respectivamente.

- Configuração do Firebase:

```
const firebaseConfig = {  
  ...  
};
```

Esta é a configuração do projeto Firebase. Quando você cria um projeto no Firebase Console, o Firebase fornece essas credenciais que permitem que seu aplicativo se conecte aos serviços do Firebase. Cada campo desempenha um papel específico:

- **apiKey**: A chave da API pública para acessar o Firebase.
  - **authDomain**: Domínio utilizado para autenticação.
  - **projectId**: ID único do projeto no Firebase.
  - **storageBucket**: URL do Google Cloud Storage onde os arquivos (como imagens) são armazenados.
  - **messagingSenderId**: ID do remetente para o Firebase Cloud Messaging.
  - **appId**: ID único do app.
  - **measurementId**: ID usado para Google Analytics.
- 
- Inicialização do App:

```
const app = initializeApp(firebaseConfig);
```

Aqui, a função `initializeApp` é chamada com a configuração do Firebase, iniciando assim a conexão com o Firebase.

- Inicialização dos serviços:

```
export const auth = getAuth(app);
```

```
export const db = getFirestore(app);
```

- **getAuth(app)** inicia o serviço de autenticação e o torna disponível para outras partes do código como `auth`.
- **getFirestore(app)** inicializa o Firestore, permitindo operações de banco de dados, e o torna disponível como `db`. (FIREBASE, 2023)

Figura 6- Arquivo firebaseConfig

```
src > services >  firebaseConfig.ts > ...  
victor ciechovicz, 7 months ago | 2 authors (MarcosStevanini and others)  
1 import { initializeApp } from "firebase/app";  
2 import { getAuth } from "firebase/auth";  
3 import { getFirestore } from "firebase/firestore";  
4  
5 // import { getAnalytics } from "firebase/analytics";  
6  
7  
8 const firebaseConfig = {  
9   apiKey: "AIzaSyB6JZ4wfb_MZNT0QZ928_7UWJV0ug5-8ps",  
10  authDomain: "aprota66-630a7.firebaseio.com",  
11  projectId: "aprota66-630a7",  
12  storageBucket: "aprota66-630a7.appspot.com",  
13  messagingSenderId: "355377107221",  
14  appId: "1:355377107221:web:af5ee65e5ac5833a7c991c",  
15  measurementId: "G-3RPWJ482JQ"  
16 };  
17  
18 const app = initializeApp(firebaseConfig);  
19 // const analytics = getAnalytics(app);  
20 export const auth = getAuth(app);  
21  
22 export const db = getFirestore(app);  
23
```

Fonte: Elaborado pelo autor

Após estas etapas, seu aplicativo React Native estará configurado para usar o Firebase.

### 6.3.2.3 Exemplo de uso Firestore

Figura 7 - Exemplo de uso Firestore

```
useEffect(() => {
  setIsLoading(true)

  const subscribe = firestore()
    .collection<T.Apoc597Props>('audios')
    .where('estudo', '==', 'Estudo 597')
    .onSnapshot(snapshot => {
      const data = snapshot.docs.map(doc => {
        const {
          titulo,
          livro,
          capitulo,
          descricao,
          playlist,
          estudo,
          imagBookItem,
          imagBookPlayer,
          tema,
          time,
          url
        } = doc.data()

        const isFavorite = favoriteAudioIds.includes(doc.id)

        return {
          id: doc.id,
          titulo,
          livro,
          capitulo,
          descricao,
          playlist,
          estudo,
          imagBookItem,
          imagBookPlayer,
          tema,
          time,
          url,
          isFavorite
        }
      })
      setAudio(data)
      setIsLoading(false)
    })
  return subscribe
}, [favoriteAudioIds])
```

Fonte: Elaborado pelo autor

Na Figura 7, vemos um exemplo de uso do Firestore. Ao renderizar a tela, o código busca na collection 'áudio', que contém objetos representando cada áudio com atributos como título, livro, capítulo, entre outros. Utilizando a função "where", filtra-se os áudios associados ao "Estudo 597". Após a busca, o resultado é processado e armazenado em um estado para uso na tela.

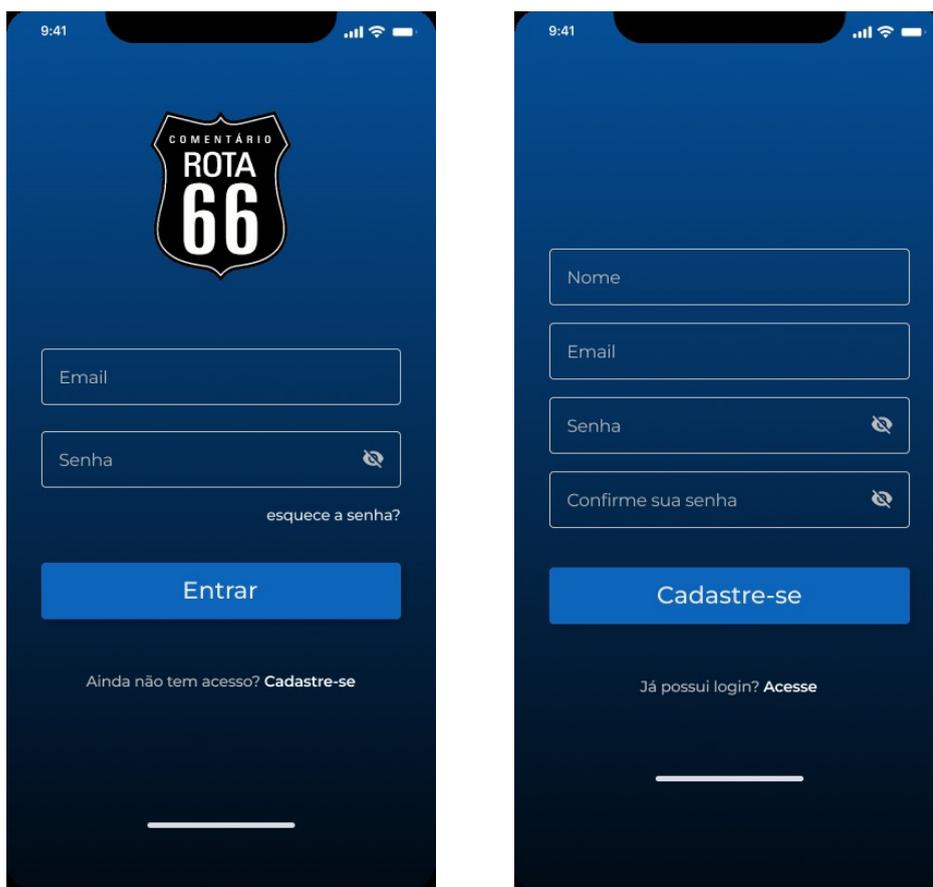
## 6.4 Resultados

O produto final deste trabalho foi um protótipo de aplicativo. Como já mencionado o aplicativo é dividido em duas áreas: I) pública, que são as telas de acesso a todos que desejarem instalar o aplicativo. II) autenticadas, onde as telas são visíveis somente após o login, ou seja, estão disponíveis somente aos usuários na plataforma.

### 6.4.1 Telas Públicas

As telas públicas, compõem a parte pública do sistema, onde o usuário tem acesso às funcionalidades sem nenhum tipo de restrição. A Figura 8a apresenta o resultado da tela inicial de login. Nela o usuário poderá realizar login caso já tenha um e-mail cadastrado, caso o mesmo não tenha ainda pode optar por realizar o cadastro no sistema através da Figura 8b.

Figura 8 - Telas públicas



(a) Tela de Login

(b) Tela de Cadastro

Fonte: Elaborado pelo autor

Figura 9 - Tela recuperação de senha



Fonte: elaborado pelo autor

Caso o usuário já tenha cadastro mas o mesmo não lembre da senha, é possível também realizar a recuperação de senha conforme apresentado na Figura 9, que é gerenciada toda pelo Firebase através de email, o usuário recebe um email e consegue redefinir sua senha.

### 6.4.2 Telas Autenticadas

Após ser realizado login o usuário será redirecionado para as telas autenticadas. caso seja o primeiro acesso deste usuário, ele será levado para uma tela (Figura 8) para escolha do nome que deseja ser chamado, este nome irá ser vinculado com o resto das informações do cadastro e também será apresentado na home inicial.

Figura 10 - Tela escolha nome



Fonte: elaborado pelo autor

Após ser realizado login e escolha do nome, o usuário é redirecionado para tela home (Figura 11). Nesta tela temos algumas opções para o redirecionamento do usuário a outras telas.

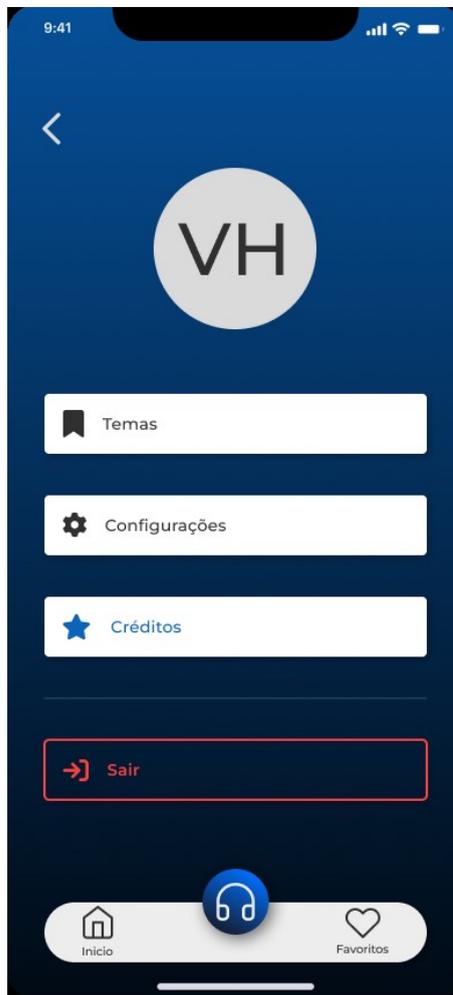
Figura 11 - Tela Home



Fonte: elaborado pelo autor

No canto superior direito temos o ícone que redireciona para a tela de usuário (Figura 12). Descendo um pouco mais temos um componente que lista alguns livros “não todos” para acesso rápido a tela do livro específico (Figura 13). No componente de “Ver todos” redireciona para a tela geral de livros (Figura 14). Na seção de temas, são distribuídos alguns temas pré selecionados de acordo com o conteúdo do áudio e leva para a listagens dos áudios referente a este tema (Figura 15). No menu tab abaixo temos as opções de navegação que serão vistas em todas as telas, dando a possibilidade do usuário navegar para a tela home (Figura 11), tela de livros (Figura 14) ou para tela de áudios favoritos (Figura 16).

Figura 12 - Tela do usuário



Fonte: elaborado pelo autor

Nesta tela temos algumas opções para redirecionar o usuário a outras telas, como tela de temas (Figura 19) tela de configurações (Figura 20) tela de créditos (Figura 21). Junto a esta tela temos a opções de sair da sessão autenticada com este botão de “Sair”.

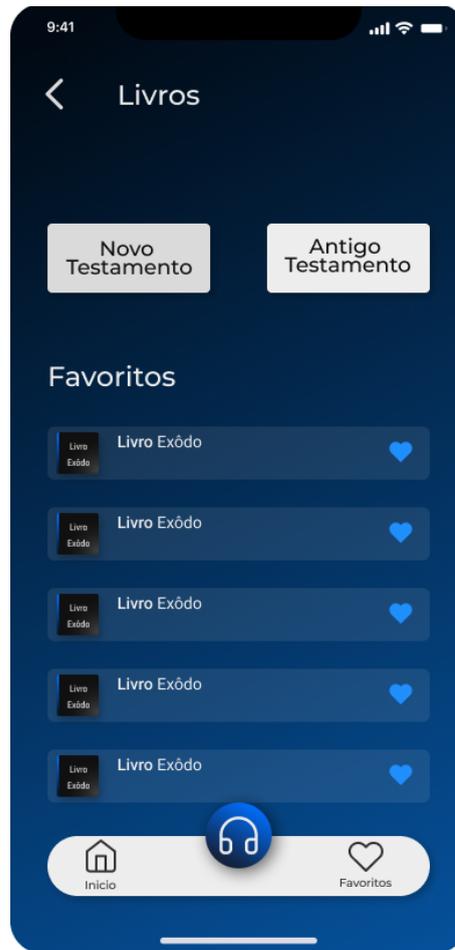
Figura 13 - Tela livro específico



Fonte: elaborado pelo autor

Nesta tela de livro específico, temos o exemplo do livro de Gênesis, esta listagem está organizada de acordo com o número de estudos relacionados a este livro. Conforme a figura mostra o livro possui 32 estudos, cada estudo pode conter mais de um áudio.

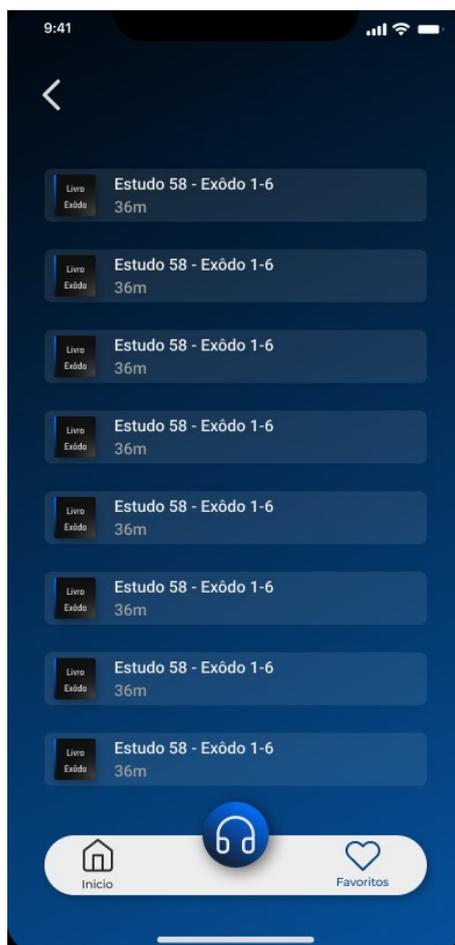
Figura 14 - Tela de livros



Fonte: elaborado pelo autor

Nesta tela temos o afunilamento para a organização em dois grandes grupos na qual os livros estão distribuídos: “Novo Testamento” e “Antigo Testamento” . Ao clicar em alguma das opções o usuário é redirecionado para a tela (Figura 17). Abaixo temos uma listagem dos livros favoritos de acordo com o gosto do usuário, ao clicar em alguns deles o usuário é redirecionado para tela de áudios daquele livro específico (Figura 13).

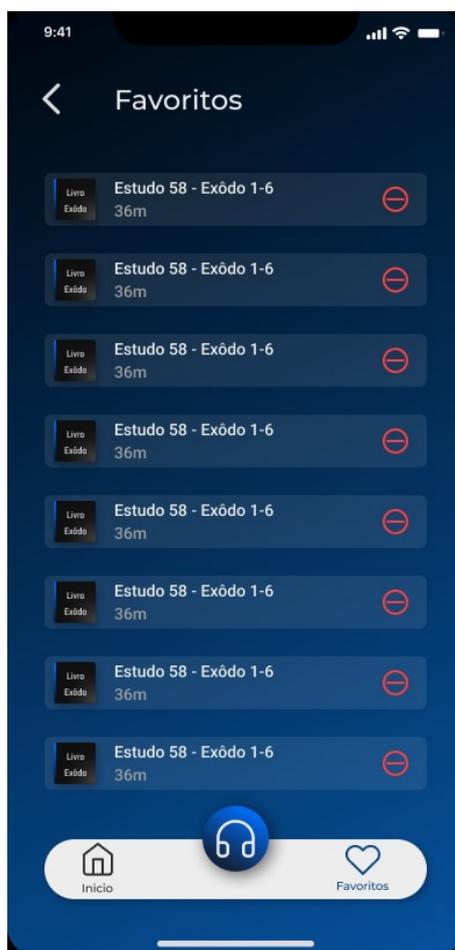
Figura 15 - Tela de tema específico



Fonte: elaborado pelo autor

Nesta tela temos as listagens de áudios de acordo com o tema a que o áudio pertence, esta definição de qual áudio o tema vai pertencer é feita pelo Administrador do banco de dados Firebase. Se o usuário clicar em qualquer áudio ele é redirecionado para o player e pode escutar o áudio (Figura 18).

Figura 16 - Tela favoritos



Fonte: elaborado pelo autor

Nesta tela estão listados os áudios favoritos de que foram favoritados pelo usuário, ele tem a opção de desfavoritar o áudio que desejar. Ao clicar em algum áudio que está favoritado o usuário é redirecionado para tela de player (Figura 18).

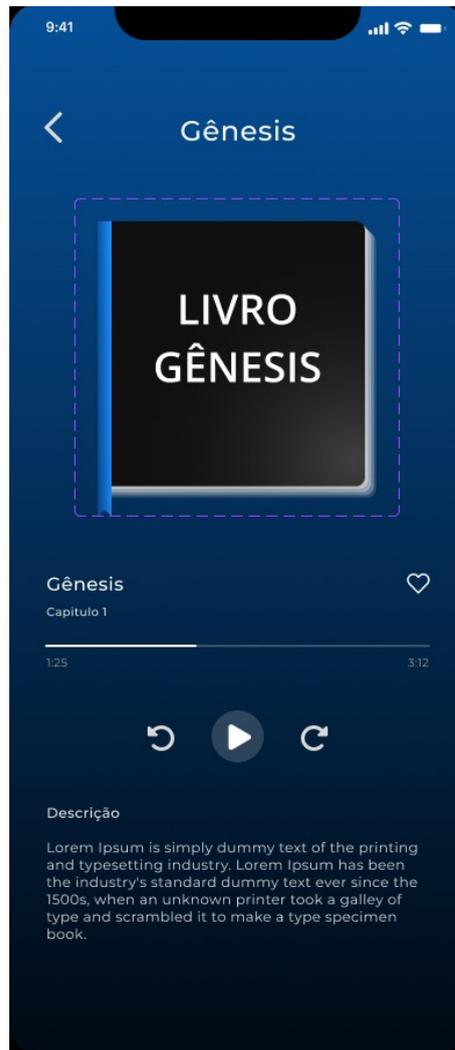
Figura 17 - Telas de livros



Fonte: elaborado pelo autor

Esta tela de livros (Figura 17) está listada em todos os livros da bíblia de acordo com o Testamento. Se o usuário clicar no livro que desejar ele é redirecionado para a tela de listagens dos áudios daquele livro específico (Figura 13).

Figura 18 - Tela player



Fonte: elaborado pelo autor

Nesta tela temos o player de áudios, nesta tela também podemos favoritar o áudio escolhido e realizar as ações de pausa, play e avançar ou retroceder o áudio 5 segundos.

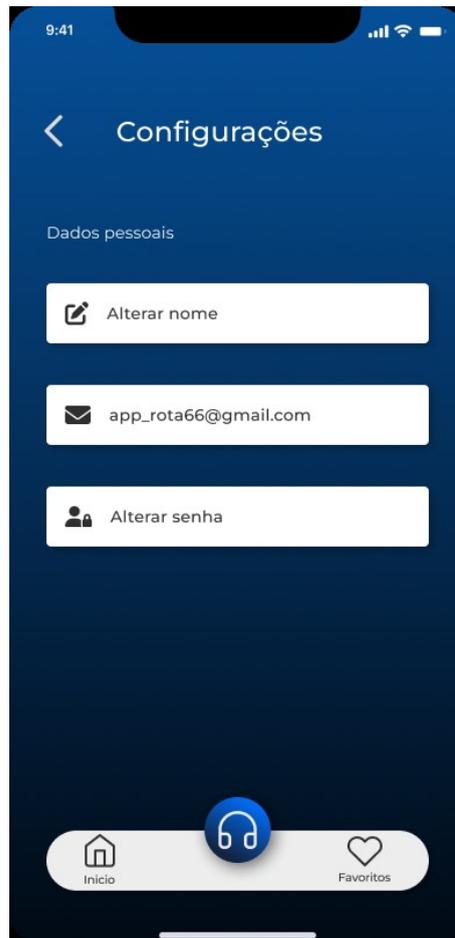
Figura 19 - Tela temas



Fonte: elaborado pelo autor

Nesta tela temos a distribuição dos temas que seguem a mesma funcionalidade da tela home (Figura 11), mas nesse caso estão segregadas em uma tela separada e exclusiva, caso seja necessário adicionar mais opções de temas futuramente, além dos existentes no protótipo.

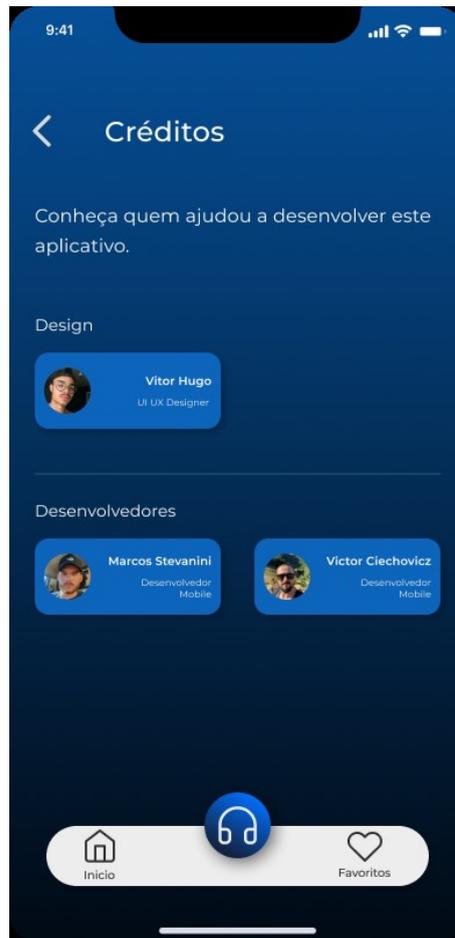
Figura 20 - Tela configurações



Fonte: elaborado pelo autor

Nesta tela temos alguns itens de configuração na qual o usuário pode realizar a alteração, como seu nome e senha, desta forma tornando mais prático algumas situações na qual o mesmo pode realizar sem ter que sair da aplicação e entrar novamente para trocar sua senha.

Figura 21 - Tela créditos



Fonte: elaborado pelo autor

Nesta tela temos os créditos para as pessoas que participaram de alguma forma no desenvolvimento do aplicativo, dando acesso a algumas redes sociais dos mesmo, caso alguém queira algum contato futuro.

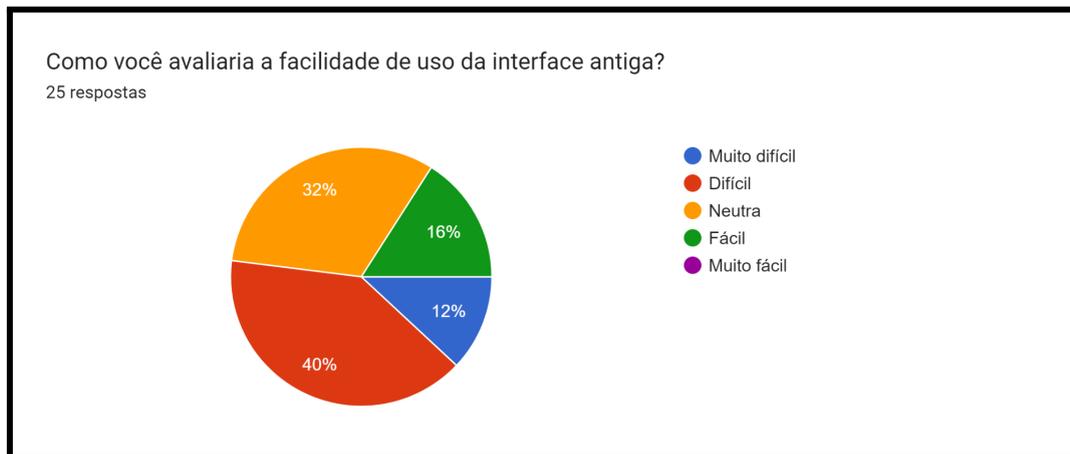


## 6.5 Coleta de Dados

O levantamento de dados sobre a experiência do usuário atual foi realizado através de entrevistas e questionários estruturados através de formulários comparativos entre a interface atual e a melhorada. Além disso, foram analisadas métricas de interação e psicologia das cores para compreender o comportamento do usuário dentro do aplicativo.

### 6.5.1 Avaliação Geral

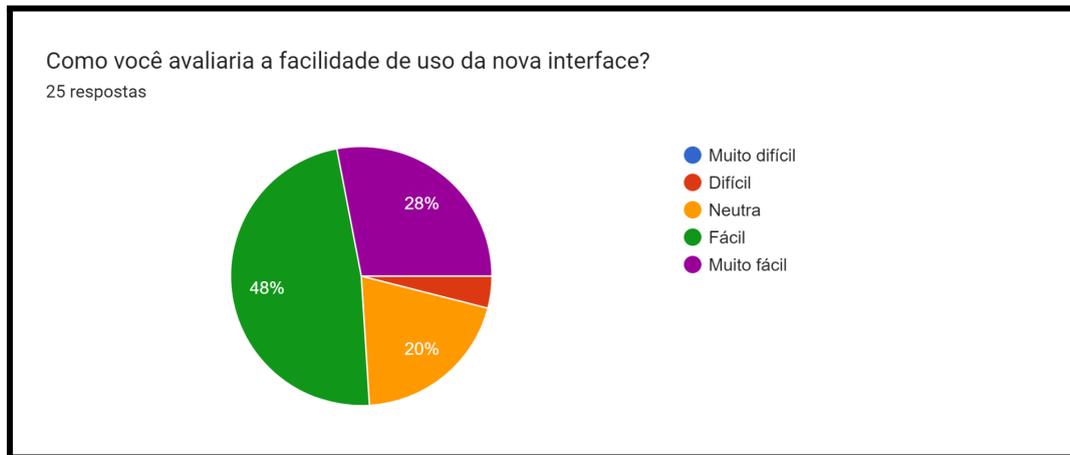
Figura 23 - Resultado Teste Usabilidade



Fonte: Elaborado pelo autor

O primeiro gráfico Figura 23, mostra a avaliação da facilidade de uso da interface antiga. As porcentagens indicam que 40% dos usuários acharam a interface antiga "Fácil", enquanto 32% a consideraram "Muito difícil".

Figura 24 - Resultado Teste Usabilidade



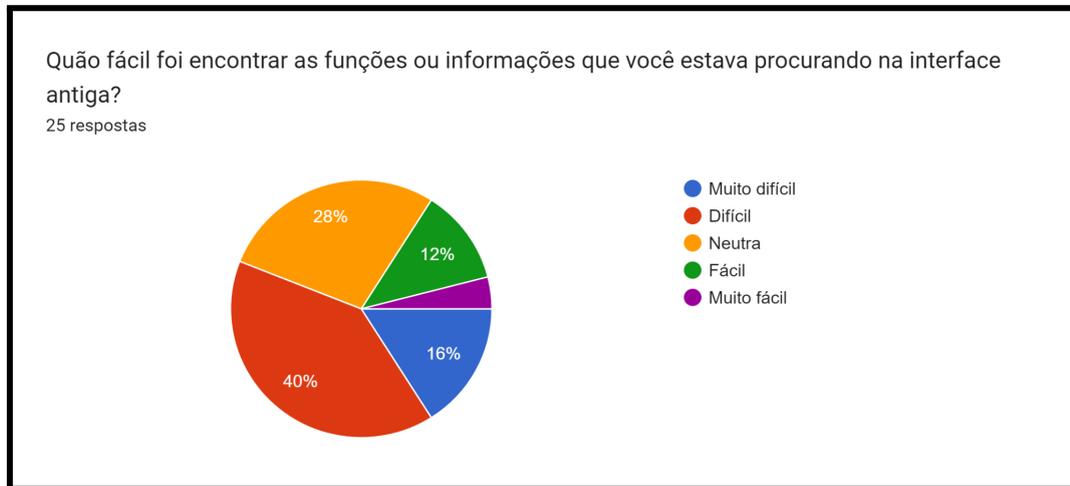
Fonte: Elaborado pelo autor

O segundo gráfico, Figura 24, mostra a avaliação da facilidade de uso da nova interface. Aqui, uma maior porcentagem (48%) dos usuários achou a nova interface "Fácil", e apenas 8% a acharam "Muito difícil".

Comparando os dois gráficos, podemos ver uma melhoria perceptível na usabilidade da nova interface em comparação com a antiga, com uma diminuição nos usuários que acharam a interface "Muito difícil" e um aumento naqueles que acharam "Fácil". Isso sugere que as mudanças implementadas na nova interface tiveram um impacto positivo na experiência do usuário.

## 6.5.2 Localização de Recursos

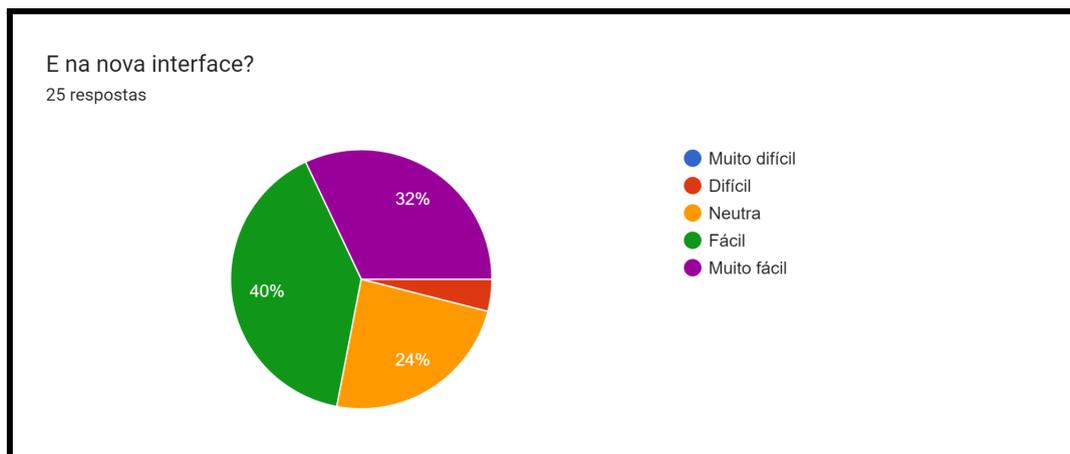
Figura 25 - Resultado Teste Usabilidade



Fonte: Elaborado pelo autor

O gráfico Figura 25 mostra a percepção dos usuários sobre a facilidade de encontrar recursos na interface antiga. A maioria (40%) achou fácil, enquanto uma parcela significativa (28%) achou muito difícil.

Figura 26 - Resultado Teste Usabilidade



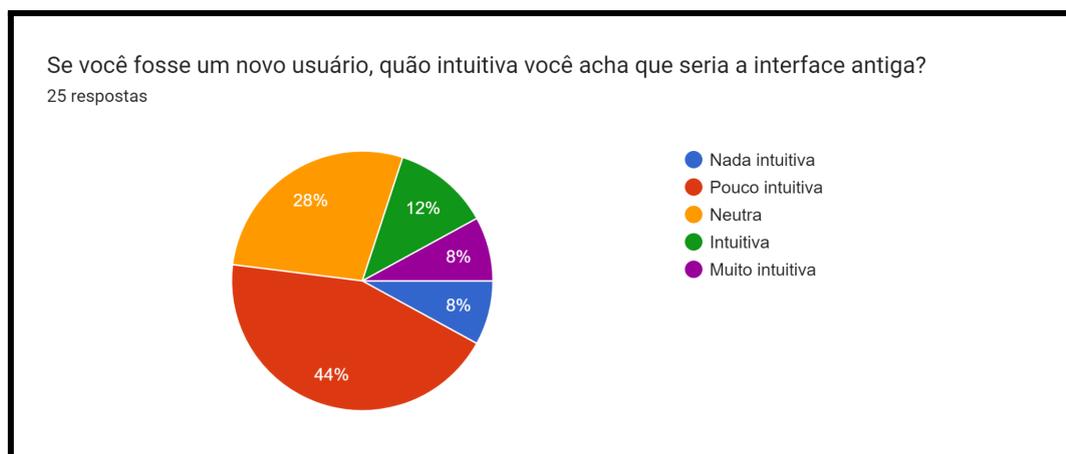
Fonte: Elaborado pelo autor

O gráfico Figura 26 mostra a percepção dos usuários sobre a facilidade de encontrar recursos na nova interface. Aqui, a maior parte dos usuários (40%) também achou fácil, mas menos usuários (24% comparado a 28% na interface antiga) acharam muito difícil.

Comparando os dois gráficos, é possível observar uma tendência semelhante à dos gráficos anteriores, com a nova interface sendo avaliada como mais fácil de usar do que a antiga. Menos usuários acharam a nova interface "Muito difícil" em comparação com a antiga, o que pode indicar que a nova interface oferece uma melhor localização de recursos ou é mais intuitiva.

### 6.5.3 Aprendizado e Curva de Aprendizado

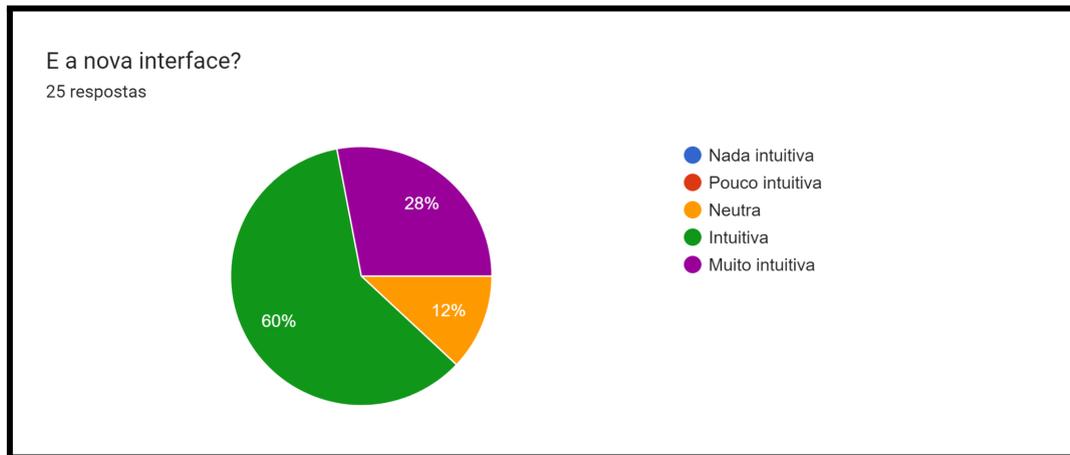
Figura 27 - Resultado Teste Usabilidade



Fonte: Elaborado pelo autor

O gráfico Figura 27 pergunta aos usuários quão intuitiva eles acharam a interface antiga. A maioria (44%) achou que a interface antiga era "Pouco intuitiva". Um percentual significativo (28%) achou "Nada intuitiva", e um pequeno grupo (12%) achou "Intuitiva". Ninguém marcou a opção "Muito intuitiva".

Figura 28 - Resultado Teste Usabilidade



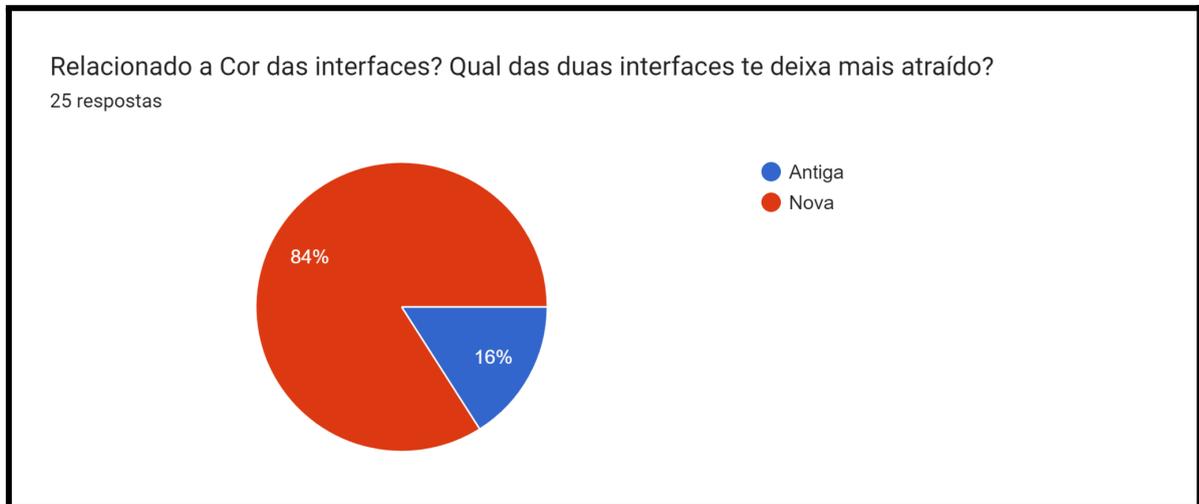
Fonte: Elaborado pelo autor

O gráfico Figura 28 pergunta aos usuários sobre a intuição na utilização da nova interface. Uma maioria esmagadora (60%) considerou a nova interface "Muito intuitiva". Uma minoria (20%) achou "Intuitiva", e apenas 12% acham "Pouco intuitiva". A opção "Nada intuitiva" não foi escolhida por nenhum dos usuários.

Esses gráficos indicam que a nova interface é percebida como significativamente mais intuitiva do que a antiga. Isso implica que os usuários encontram a nova interface mais fácil de aprender e usar, o que é um indicador positivo para o design e a ergonomia da interface.

## 6.5.4 Estética e Aparência

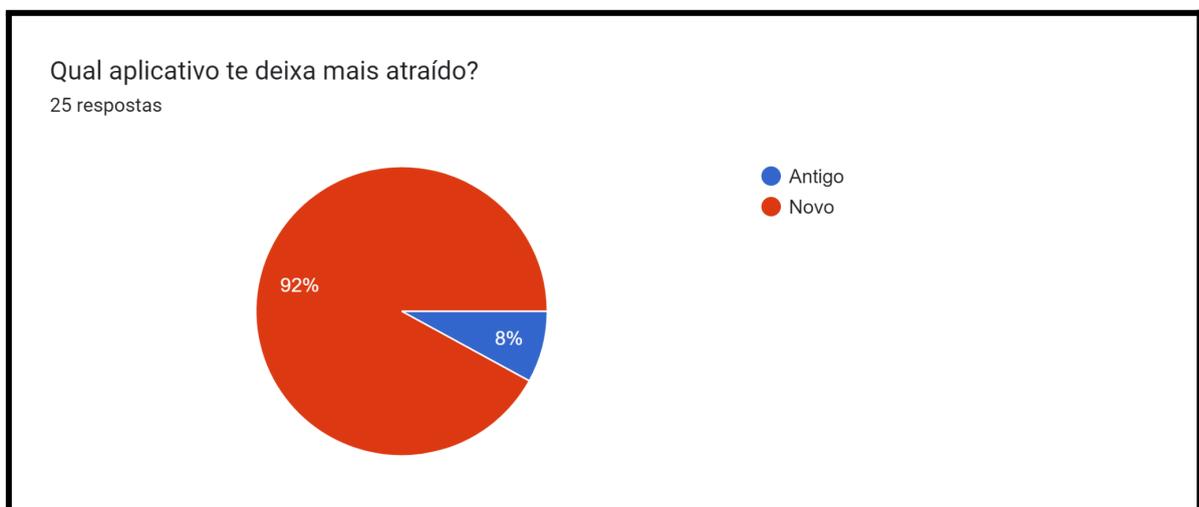
Figura 29 - Resultado Teste Usabilidade



Fonte: Elaborado pelo autor

A Figura 29 mostra a resposta para a pergunta sobre qual das duas interfaces (antiga ou nova) tem uma cor que mais atrai os usuários. A grande maioria, 84%, prefere a cor da nova interface, indicando que a atualização provavelmente adotou uma paleta de cores mais agradável aos usuários.

Figura 30 - Resultado Teste Usabilidade

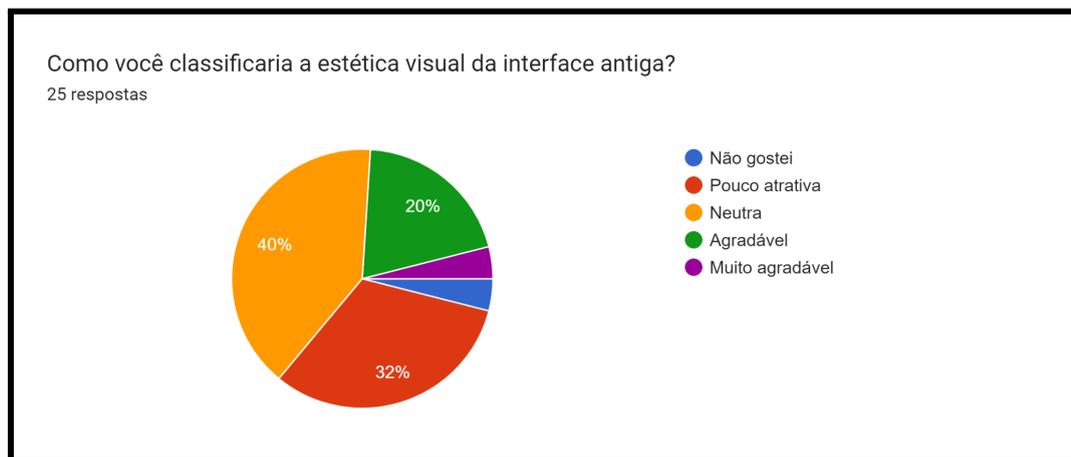


Fonte: Elaborado pelo autor

A Figura 30 refere-se à pergunta sobre qual versão do aplicativo (antiga ou nova) é mais atraente para os usuários. Um esmagador 92% dos usuários achou a nova versão mais atraente, sugerindo que a nova interface é visualmente mais agradável e possivelmente apresenta melhor design gráfico e layout.

Ambos os gráficos indicam uma forte preferência pela nova interface em termos de estética e apelo visual, o que pode ter um impacto positivo na satisfação do usuário e na adoção do aplicativo.

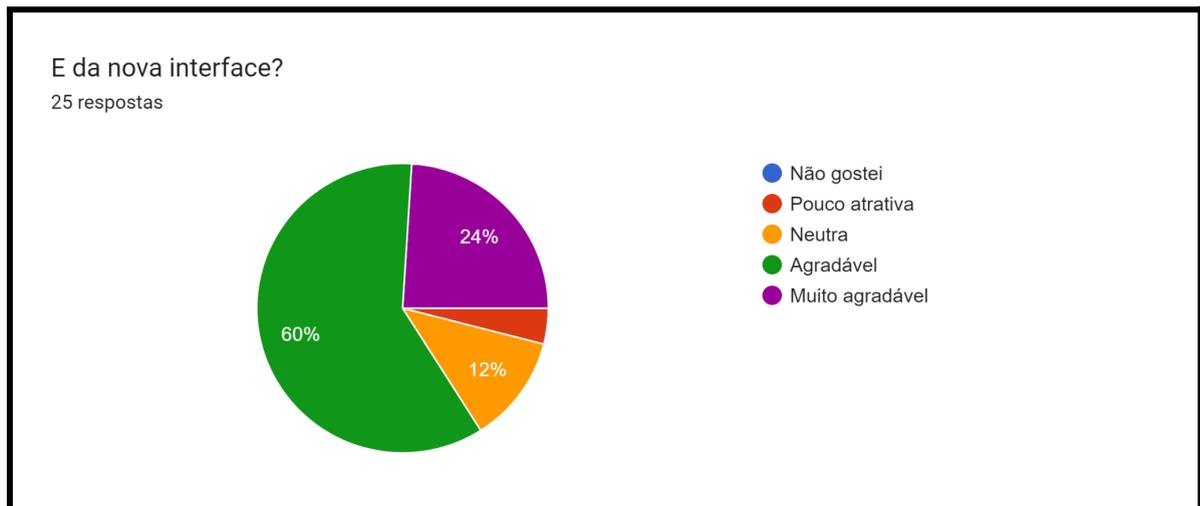
Figura 31 - Resultado Teste Usabilidade



Fonte: Elaborado pelo autor

A Figura 31 mostra as percepções dos usuários sobre a estética visual da interface antiga do aplicativo. As respostas estão divididas da seguinte maneira: 40% dos usuários acharam a estética visual da interface antiga "Pouco atraente", 32% a consideraram "Neutra", 20% disseram que era "Agradável", e um menor segmento de 8% a classificou como "Muito agradável".

Figura 32 - Resultado Teste Usabilidade



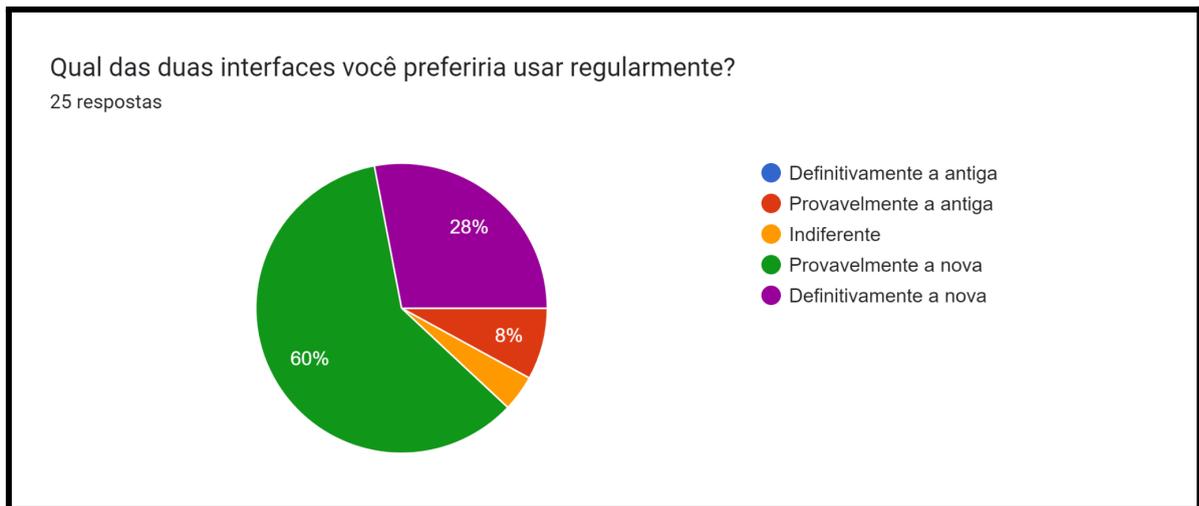
Fonte: Elaborado pelo autor

A Figura 32 reflete as opiniões dos usuários sobre a estética visual da nova interface. A maioria dos usuários, 60%, classificou a estética da nova interface como "Muito agradável", indicando uma clara melhoria em relação à interface antiga. Além disso, 24% acharam "Agradável" e 12% "Neutra". Apenas 4% dos usuários não gostaram da nova estética.

Esses gráficos indicam uma melhoria substancial na estética visual da nova interface em comparação com a antiga, com um aumento significativo no número de usuários que encontraram a nova interface visualmente agradável. Isso sugere que as atualizações de design implementadas foram bem-sucedidas em melhorar a percepção visual do aplicativo pelos usuários.

### 6.5.5 Preferência Geral

Figura 33 - Resultado Teste Usabilidade



Fonte: Elaborado pelo autor

O gráfico Figura 33 mostra as respostas dos usuários à pergunta sobre qual interface eles prefeririam usar regularmente. Uma grande maioria de 60% dos usuários prefere definitivamente a nova interface, enquanto 28% têm uma preferência leve pela nova interface, resultando em uma esmagadora preferência geral de 88% para a nova interface sobre a antiga. Apenas 8% dos usuários preferem a interface antiga, seja definitiva ou provavelmente, e ninguém é indiferente na escolha.

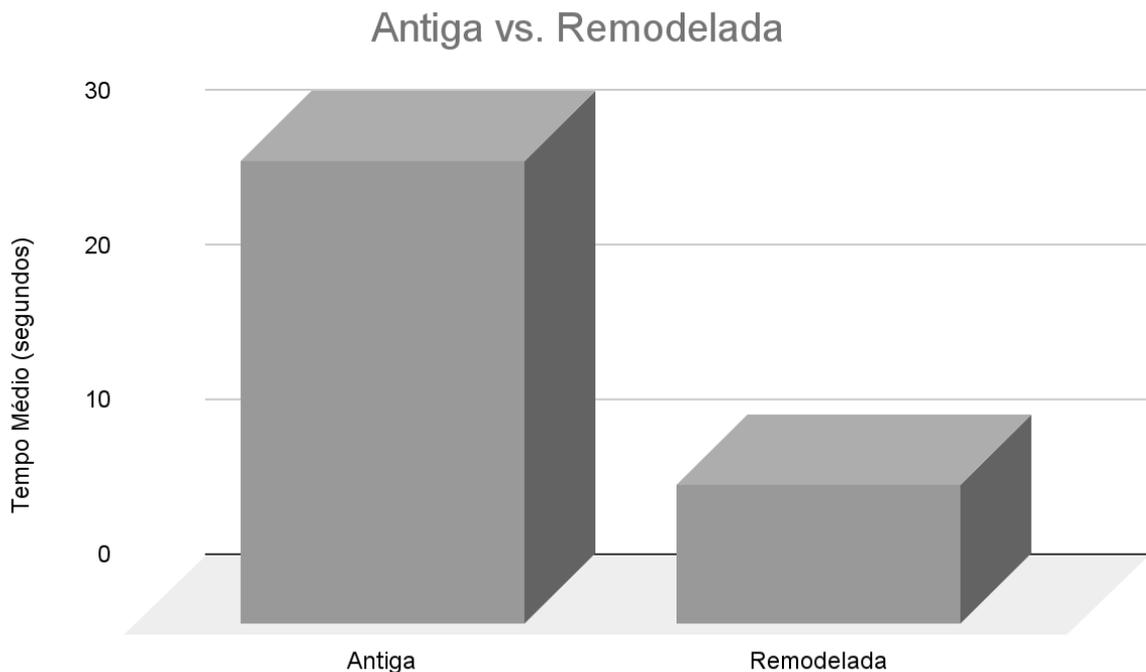
Este resultado sugere uma forte inclinação dos usuários para a nova interface, indicando que as atualizações feitas são bem recebidas e que a nova interface é preferida para uso regular. Isso pode ser devido a melhorias na usabilidade, na estética ou em outras áreas de design da interface do usuário.

### 6.5.6 Análise Tempo de Navegação

Com o objetivo de otimizar a experiência do usuário, implementamos uma reformulação completa na interface do aplicativo. Esta nova abordagem, focada na usabilidade e acessibilidade, transformou significativamente a interação do usuário com o aplicativo, tornando-a mais fluida e intuitiva.

Um exemplo concreto dessa melhoria pode ser observado no tempo necessário para realizar uma tarefa específica: Encontrar o estudo número 122. Na versão anterior do aplicativo, esta ação levava em média 30 segundos. Após a remodelação da interface, o mesmo processo passou a ser concluído em apenas 9 segundos. Isso representa uma redução de tempo de 70%, evidenciando a eficácia das mudanças implementadas.

Figura 34 - Gráfico comparativo Tempo Médio de Procura



Fonte: Elaborado pelo autor

## CONCLUSÃO

A partir do estudo apresentado, foi possível concluir com êxito que o projeto de reestruturação do aplicativo Rota 66 alcançou seus objetivos, especialmente com a prototipação bem-sucedida e as funções propostas operando corretamente em avaliações conduzidas pelo autor.

Um destaque relevante foi a escolha do React Native como ferramenta para o desenvolvimento. Isso se confirmou acertado, visto que o React Native é amplamente reconhecido por sua capacidade de criar aplicativos móveis que operam nativamente tanto em IOS quanto em Android.

A decisão de empregar o banco de dados não relacional Firebase mostrou-se estratégica, otimizando a arquitetura do aplicativo para um funcionamento eficaz em plataformas como mobile e web. A adaptabilidade e as funcionalidades oferecidas pelo Firebase simplificaram e enriqueceram o processo de desenvolvimento.

Olhando para o horizonte, algumas funcionalidades podem ser incorporadas para aprimorar o protótipo, visando sua eventual publicação:

- a) Painel Administrativo (web) para gestão dos áudios;
- b) Mecanismo aprimorado de busca de áudios, seja através de filtros ou inserção direta do nome do áudio;
- c) Integração multilíngue, ampliando a acessibilidade do aplicativo em mercados internacionais;
- d) Mecanismo de confirmação de cadastro do usuário, seja via e-mail ou SMS.

Para o autor, o desenvolvimento deste projeto foi uma jornada rica em aprendizado tecnológico. A seleção das práticas e ferramentas envolveu extensa pesquisa e testes. Apesar dos desafios e inconsistências encontrados ao longo do caminho, a determinação prevaleceu, e não houve obstáculos que comprometessem ou interrompessem o progresso.

## REFERÊNCIAS

- ABRAMOV, D. Overreacted – A blog by Dan Abramov. 2019. Disponível em: <https://overreacted.io/>. Acesso em: 9 set. 2023.
- AGARWAL, S. Mastering Firebase for Android Development. Packt Publishing, 2018.
- ALURA. React Native: o que é e tudo sobre o Framework. 2023. Disponível em: <https://www.alura.com.br/artigos/react-native>. Acesso em: 2 set. 2023.
- AXEL, R. Native React for Mobile Development. Packt Publishing, 2017.
- APPLE STORE. Apple Store. Disponível em: [https://pt.wikipedia.org/wiki/Apple\\_Store](https://pt.wikipedia.org/wiki/Apple_Store). Acesso em: 3 Nov. 2023.
- BIERMAN, G.; ABADI, M.; TORGERSEN, M. Understanding TypeScript. In: Proceedings of the European Conference on Object-Oriented Programming (ECOOP), 2014.
- BOCARD.T. (2021, Marco 29). O que são aplicativos?. Disponível em: <https://usemobile.com.br/aplicativo-movel/>. Acesso em: 22 set. 2023.
- BONAMENTE, M. Firebase for Mobile App Development. Apress, 2018.
- FIREBASE. Firebase Documentation. 2023. Disponível em: <https://firebase.google.com/>. Acesso em: 2 set. 2023.
- GOOGLE INC. Firebase: Build, improve, grow your app. 2016. Disponível em: <https://firebase.google.com/>. Acesso em: 2 set. 2023.
- GOOGLE PLAY. Google Play Store. Disponível em: [https://pt.wikipedia.org/wiki/Google\\_Play](https://pt.wikipedia.org/wiki/Google_Play). Acesso em: 03 nov. 2023.
- HANSSON, D. H. The Rails Way. Addison-Wesley, 2008.
- IBM. What is mobile application development?. 2023. Disponível em: <https://www.ibm.com/topics/mobile-application-development>. Acesso em: 2 set. 2023.
- MAJID, A. React Native Practical Guide. Apress, 2020.
- MDN WEB DOCS. JavaScript. 2023. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. Acesso em: 2 set. 2023.
- W3SCHOOLS. Web Tutorials. 2023. Disponível em: <https://www.w3schools.com/>. Acesso em: 29 set. 2023.
- MOURA, C. Z. Barbers: um protótipo de aplicativo mobile para mediar a busca entre clientes e profissionais barbeiros. Instituto Federal de Educação Ciência e Tecnologia Farroupilha, PANAMBI, 2020.
- NADER, A. React Native in Action. Manning Publications, 2020.
- Nielsen, J. (2005). Usability for the Web: Designing web sites that work. San Francisco: Morgan Kaufmann.

SAHNI, R. Learning React Native: Building Native Mobile Apps with JavaScript. O'Reilly Media, 2018.

SMITH, J. The Importance of User Experience in Mobile Apps. 2021.

STARCZEWSKI, K. Mastering React Native. Packt Publishing, 2019.

REACT. React Documentation. 2023. Disponível em: <https://react.dev/> Acesso em: 2 set. 2023.

REACT NATIVE. React Native Documentation. 2023. Disponível em: <https://reactnative.dev/>. Acesso em: 2 set. 2023.

ROTA 66 - APPLE STORE. Disponível em: <https://apps.apple.com/br/app/rota-66/id1521579195>. Acesso em: 4 dez. 2023.

TYPESCRIPT. TypeScript Documentation. 2023. Disponível em: <https://www.typescriptlang.org/docs/>. Acesso em: 2 set. 2023.

WIKIPEDIA. Mobile device. 2023. Disponível em: [https://pt.wikipedia.org/wiki/Aplicativo\\_m%C3%B3vel](https://pt.wikipedia.org/wiki/Aplicativo_m%C3%B3vel). Acesso em: 2 set. 2023.

WIKIPEDIA. Mobile operating system. 2023. Disponível em: [https://pt.wikipedia.org/wiki/Sistema\\_operacional\\_m%C3%B3vel](https://pt.wikipedia.org/wiki/Sistema_operacional_m%C3%B3vel). Acesso em: 2 set. 2023.

WILSON, S. Expo for Dummies. Wiley, 2019.

WULF, W. A. et al. Fundamental Structures of Computer Science. Addison-Wesley, 1981.