

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO



JETLIST : APLICATIVO PARA AGENDAMENTO DE LAVA JATOS A DOMICÍLIO

ISAÍAS DE CARVALHO NOVAIS

GOIÂNIA
2022

ISAÍAS DE CARVALHO NOVAIS

JETLIST : APLICATIVO PARA AGENDAMENTO DE LAVA JATOS A DOMICÍLIO

Trabalho de Conclusão de Curso apresentado à Escola de Ciências Exatas e da Computação, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Me. Fernando Gonçalves Abadia

GOIÂNIA

2022

ISAÍAS DE CARVALHO NOVAIS

JETLIST : APLICATIVO PARA AGENDAMENTO DE LAVA JATOS A DOMICÍLIO

Este Trabalho de Conclusão de Curso julgado adequado para obtenção o título de Bacharel em Engenharia de Computação, e aprovado em sua forma final pela Escola Politécnica, da Pontifícia Universidade Católica de Goiás em 07 / 12 / 2022.

Profa. Ma. Ludimilla Reis Pinheiro dos Santos
Coordenadora de Trabalho de Conclusão de Curso

Bancada examinadora:

Orientador: Prof. Me. Fernando Gonçalves Abadia

Prof. Me. Rafael Leal Martins

Prof. Eng. Welder Silva Marques

Prof. Me. Gustavo Siqueira Vinhal

GOIÂNIA

2022

Dedico este trabalho a todos os meus familiares, principalmente aos meus pais Odair Santos Carvalho e Vitória Régia do P. N. Carvalho, a minha noiva Juliana Gomes Fernandes e ao meu irmão Isaque de Carvalho Novais, que sempre deram forças e sabedoria durante esta grande jornada.

AGRADECIMENTOS

Primeiramente a Deus que permitiu que tudo isso acontecesse, ao longo de minha vida, e não somente nestes anos como universitário, mas que em todos os momentos é o maior mestre que alguém pode conhecer.

Aos meus Pais que sempre me apoiaram e me deram forças, e sempre estiveram comigo.

A minha noiva que sempre esteve ao meu lado me ajudando, sempre positiva em todas ocasiões e sempre me impulsionando para não desistir dos meus sonhos.

Ao meu orientador Prof. Me. Fernando, minha enorme gratidão. Por todo incentivo, apoio, paciência, ao longo dessa jornada. Obrigado por não desistir desse desafio.

Sou grato a todo o corpo docente, à direção e administração desta universidade.

A todos os meus colegas de turma, que juntos tivemos a oportunidade de crescer pessoal e profissionalmente.

Aos meus amigos que sempre me ajudaram, me incentivando e podendo contribuir para meu crescimento, meus sinceros agradecimentos.

E a todos que direta ou indiretamente me ajudaram neste trabalho.

“E o meu Deus, segundo a sua riqueza em glória, há de suprir, em Cristo Jesus, cada uma de vossas necessidades.”

Filipenses 4:19

RESUMO

O projeto de pesquisa deste trabalho tem como objetivo desenvolver um protótipo de um aplicativo móvel multiplataforma para agendamento de um serviço de limpeza de carros (lavagem de carros) via autenticação com redes sociais ou autenticação com APIs do Google. O uso do aplicativo é conhecido por trazer benefícios como simplicidade e comodidade, e os objetivos deste trabalho são adaptados às necessidades dos profissionais autônomos, bem como de qualquer provedor para atender seus processos, necessidades e arranjos por parte dos clientes para viabilizar o cuidado em fazê-lo de forma simples e prática. Para o desenvolvimento, foi realizado um estudo bibliográfico, uma breve introdução à importância do aplicativo e histórico de uso, e foram explicadas as ferramentas de desenvolvimento. Com base nos resultados positivos de ambas as pesquisas indicando tanto a necessidade quanto a viabilidade do projeto foi dado início ao desenvolvimento do protótipo, utilizando React Native. Na etapa final são apresentadas as telas do protótipo e o descritivo sobre suas funcionalidades e explanado as conclusões sobre o desenvolvimento do projeto.

Palavras-Chave: Móvel; React Native; Domicílio;

ABSTRACT

The research project aims to develop a cross-platform mobile programming application for a car cleaning service (car wash) via authentication with social networks or authentication with Google APIs. The application is known for bringing benefits such as simplicity and convenience, and the work objectives are adapted to the owner's professionals as well as the provider to meet their processes, needs and arrangements for any care on the part of the customers to make it feasible in a simple way. and practice. For the development, a bibliographic study was carried out, a brief introduction to the application introduction and usage history, and were explained as development tools. Based on the positive data, results indicating both the necessity and feasibility of project development were started for both, using React Native. In the final phase are screens of project development and features explained as execution on project development.

Keywords: Mobile; React Native; Residence;

LISTA DE FIGURAS

Figura 1 - Elementos de um componente <i>React Native</i> definidos pelo <i>Framework</i>	19
Figura 2 - Comportamento da aplicação <i>React Native</i> em relação ao <i>React</i>	20
Figura 3 - Serviços disponibilizados pelo <i>FireBase</i>	23
Figura 4 - Tabela de Colorimetria cores quentes e frias	25
Figura 5 - Paleta de cores utilizado no aplicativo	26
Figura 6 - Visual Studio Code (VSCode).....	29
Figura 7 - Figma	30
Figura 8 - Tela de início do Android Studio	31
Figura 9 - Fluxograma de login	34
Figura 10 - Fluxograma de agendamento	35
Figura 11 - Fluxograma de mensagens e premiação	36
Figura 12 - Tela de Splash.....	37
Figura 13 - Tela de Início	38
Figura 14 - Tela de confirmação do Discord	39
Figura 15 - Processo de autenticação OAuth 2.....	40
Figura 16 - Tela home.....	41
Figura 17 - Listagem de Fornecedores	43
Figura 18 - Tela de Agendamento.....	44

LISTA DE SIGLAS

ECMA	Associação Europeia dos Fabricantes de Computadores
DOM	Document Object Model
VDOM	Virtual Document Object Model
JSX	JavaScript XML
HTML	HyperText Markup Language
API	Application Programming Interface
IDE	Integrated Development Environment
NPM	Node Package Manager
BaaS	Backend as a Service
SDK	Software Development Kit
UI	User Interface
RDBMS	Relational Database Management System
SQL	Structured Query Language
SGBD	Data Base Management System
JSON	Javascript Object Notation

SUMÁRIO

INTRODUÇÃO	13
1.1 Objetivos	14
1.1.2 Objetivos Gerais	14
1.1.3 Objetivos Específicos	14
1.2 Justificativa	14
1.3 Resultados Esperados	15
1.4 Organização do Trabalho	15
2 FUNDAMENTAÇÃO TEÓRICA	17
2.1 Javascript	17
2.2 React	18
2.3 React Native	19
2.3.1 Expo	20
2.3.2 Node.js	21
2.3.3 NPM	21
2.4 Firebase	22
2.4.1 Firebase Authentication	23
2.5 MySQL	24
2.6 Teoria das Cores	24
2.7 Cromoterapia	25
3 MATERIAIS E PROCEDIMENTOS METODOLÓGICOS	27
3.1 Métodos	27
3.2 Materiais	28
3.2.1 Visual Studio Code	28
3.2.2 Figma	29
3.2.3 Android Studio	30
3.2.4 Computador Utilizado	31
4 DESENVOLVIMENTO	33
4.1 Prototipação	33
4.2 Linguagem Utilizada	33
4.3 Fluxograma	34
5 RESULTADOS ESPERADOS	37
5.1 Tela de Splash	37
5.2 Tela Inicial	38
5.3 Tela Home	40
5.4 Tela de Agendamento	42
CONSIDERAÇÕES FINAIS	45

5.1 Trabalhos Futuros	46
REFERÊNCIAS	47

INTRODUÇÃO

Na segunda metade do século XX, mais precisamente entre as décadas de 1950 e 1970, deu-se início a Era Digital, período advindo dos desígnios e dos avanços tecnológicos da Terceira Revolução Industrial. Com isso, as relações humanas sofreram mudanças, tanto sociais quanto culturais, onde uma sociedade cada vez mais se conecta a um ambiente virtual. Um dos grandes fatores dessa mudança cultural/social, é a disseminação e popularização da internet via smartphones.

Segundo (Mawston, 2021), em 1994 existiam pouco mais de 30 mil smartphones no mundo, já em 2012 eram mais de 1 bilhão de aparelhos móveis. Até o ano passado, de 2021, a marca ultrapassou os 3,95 bilhões. Implicando um aumento exponencial, fazendo com que mais de 50% da população mundial tenha acesso a algum dispositivo móvel, marca histórica que levou apenas 27 anos para ser alcançada.

Pedir uma refeição, chamar um táxi, checar os e-mails, verificar o extrato bancário, navegar na internet, compartilhar arquivos, acessar as redes sociais, comprar, vender, procurar por produtos ou serviços, ouvir músicas, assistir vídeos, jogar, estão entre milhares de possibilidades de serviços disponíveis na palma da mão. Esse número infinito de possibilidades traz grandes oportunidades para o desenvolvimento de aplicativos móveis, focando na facilidade e comodidade de seus usuários (SANTOS, 2018).

O presente trabalho centra-se em propostas para o desenvolvimento de um protótipo de uma aplicação móvel destinada a facilitar a procura e marcação de serviços na área das lavagens de carros à domicílio, tanto para quem procura estes serviços como para os profissionais ou instituições que os prestam.

A relevância dos temas supracitados torna-se válida quando se observa que atualmente as pessoas não estão mais distantes de seus dispositivos móveis e os utilizam em seu cotidiano para realizar as diversas atividades descritas acima, tornando esses dispositivos praticamente inseparáveis de seus usuários (LECHETA, 2016).

Diante deste contexto, este projeto visa responder a seguinte questão de pesquisa: **Como um aplicativo móvel pode ajudar a auxiliar o agendamento de lava jatos a domicílio ?**

1.1 Objetivos

Esta seção tem por finalidade apresentar e descrever os objetivos gerais e específicos deste trabalho.

1.1.2 Objetivos Gerais

Desenvolver um aplicativo para Android/IOS que auxilie o agendamento para lavagem de carros a domicílio, por meio da seleção dos fornecedores disponíveis mais próximos ou que atendam a região desejada.

1.1.3 Objetivos Específicos

- Realizar levantamento bibliográfico das principais técnicas, padrões de projetos, relacionadas ao desenvolvimento da aplicação.
- Implementar um aplicativo Android/IOS utilizando *React Native*.
- Alimentar o banco de dados com as informações dos clientes e dos fornecedores.
- Desenvolver uma aplicação para facilitar o agendamento dos lava jatos a domicílio.
- Avaliar os resultados obtidos através da abordagem especificada.

1.2 Justificativa

O uso de aplicativos móveis está aumentando a cada dia, e sua facilidade de uso e conveniência incentiva as pessoas a utilizarem essas ferramentas, devido ao isolamento social causado pelo vírus SARS Covid-19 na pandemia iniciada em 2020, a atuação do serviço em casa pode precisar ser explorada. Novos cenários, incluindo expansão no uso, onde, por exemplo, um cliente não conhece sua cidade e não quer se mudar, mas quer um serviço focado em estética automotiva, o namoro em casa pode beneficiar tanto o cliente quanto o provedor. Essas ideias são consistentes com as dos autores abaixo, que explicam:

Aplicativos exercem um papel relevante em vários momentos do dia, como a busca de informações, conexão em redes sociais e a procura por produtos.

Eles representam espaços privilegiados para que as marcas criem relações efetivas e produtivas. A publicidade nos aplicativos móveis possui a função de complementar a experiência off-line (ofertas especiais em lojas, por exemplo), direcionar o e-commerce, ou contribuir com a comunicação de qualidade da marca (BORGES, 2019, p. 105).

Automação de processos manuais, sempre proporciona uma melhora significativa, neste caso a principal afetada é a forma de gerenciamento das agendas, horários, disponibilidade, possíveis atrasos, avisos e divulgação dos serviços dos prestadores (profissionais e estabelecimentos), assim como praticidade e comodidade para o cliente.

1.3 Resultados Esperados

Os resultados da pesquisa foram determinantes para entender o cenário real de utilização de aplicativos, plataformas, e o desejo dessas pessoas em utilizar aplicações móveis para realizar a pesquisa de estética automotiva através de recursos que facilitem e ofereçam mais comodidade às suas atividades e processos do cotidiano. Com isso possibilitando ao fornecedor agregar praticidade e comodidade aos seus clientes.

1.4 Organização do Trabalho

Este trabalho foi dividido em 5 capítulos, estruturados da seguinte maneira. No presente capítulo foi apresentado a introdução do trabalho, identificando o tema sobre o agendamento de lava jato a domicílio, e mostrando a ideia inicial do desenvolvimento do aplicativo.

No capítulo 2 será mostrado a fundamentação teórica no qual é definido a abordagem e os conceitos teóricos que fundamentam a proposta deste trabalho de conclusão de curso.

No capítulo 3, materiais e métodos, será informado de qual maneira e com quais materiais seria possível para se concluir os objetivos deste trabalho.

No capítulo 4, desenvolvimento, será mostrado a criação dos protótipos de tela, incluindo imagens das telas, testes e ambiente de trabalho.

No capítulo 5, considerações finais, onde foram feitas as últimas considerações sobre o trabalho, abordando a sua relação com os objetivos gerais e

específicos, mostrando também as propostas de trabalhos futuros, nas quais são expostas opiniões sobre o que poderá ser feito de melhoria neste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo introduzir a fundamentação teórica necessária concebido e elaborado para o presente trabalho, visando localizar e compreender os conceitos e definições usados no processo de desenvolvimento aplicativo Android/IOS no qual se baseia o objetivo deste trabalho.

2.1 Javascript

O *JavaScript* é uma linguagem de programação interpretada criada em meados da década de 90 pela Netscape Communications. Seu objetivo era atender à demanda crescente por sites mais interativos e dinâmicos, já que, até então, as páginas HTML continham pouca ou nenhuma interatividade além dos hyperlinks (MELO, 2021).

O *JavaScript* permite controlar os elementos de uma página em tempo real, sem necessariamente ter que receber os dados ou uma resposta do servidor. Por exemplo, é possível atualizar o conteúdo de uma página web sem precisar recarregá-la por completo ao preencher um formulário (SHUTE, 2019).

“Algumas páginas apresentam informações estáticas e podem ser chamadas de documentos. (A apresentação dessa informação estática pode ser bastante dinâmica por causa do JavaScript, mas a informação em si é estática). Outras páginas da web parecem mais com aplicativos do que com documentos. Essas páginas podem carregar dinamicamente novas informações conforme necessário, elas podem ser gráficas em vez de textuais e podem operar off-line e salvar dados localmente para que possam restaurar seu estado quando você visitá-los novamente.” (FLANAGAN,2012, p. 307).

Seus interpretadores estão presentes nos navegadores mais modernos dos dispositivos móveis, computadores ou até console de jogos (FLANAGAN, 2012). Por estar presente nos navegadores mais comuns, o *JavaScript* não requer nenhuma instalação separada de um plugin ao lado do cliente para ser executado. Além disso, a linguagem nos oferece uma variedade de frameworks e bibliotecas para diversos dispositivos.

É uma linguagem multiparadigma (programação funcional e imperativa), versátil, com tipagem dinâmica (não é necessário definir os tipos das variáveis ao declará-las), sintaxe acessível, e recursos avançados como orientação a objetos e

APIs para trabalhar com textos, matrizes, datas e expressões regulares. Atualmente, segue as especificações *ECMAScript*, junção entre ECMA (Associação Europeia dos Fabricantes de Computadores) juntamente com *JavaScript*, que visam garantir o suporte e o funcionamento da linguagem entre os diferentes navegadores (MOZILLA, 2022).

2.2 React

O *React* é uma biblioteca *JavaScript* de código aberto, cuja função principal é tornar a experiência do usuário mais eficiente criando interfaces de usuário. A ferramenta foi utilizada para o desenvolvimento *web* do projeto e foi escolhida pela facilidade no desenvolvimento de componentes modernos, pela combinação entre eles, atualizações frequentes e pela facilidade de encontrar materiais para consulta.

Ela é baseada em componentes escritos em *JavaScript* e faz a atualização e a renderização apenas dos componentes necessários em tela. Esses componentes poderão, ainda, ser reutilizados em outros projetos ou até mesmo dentro de outros componentes.

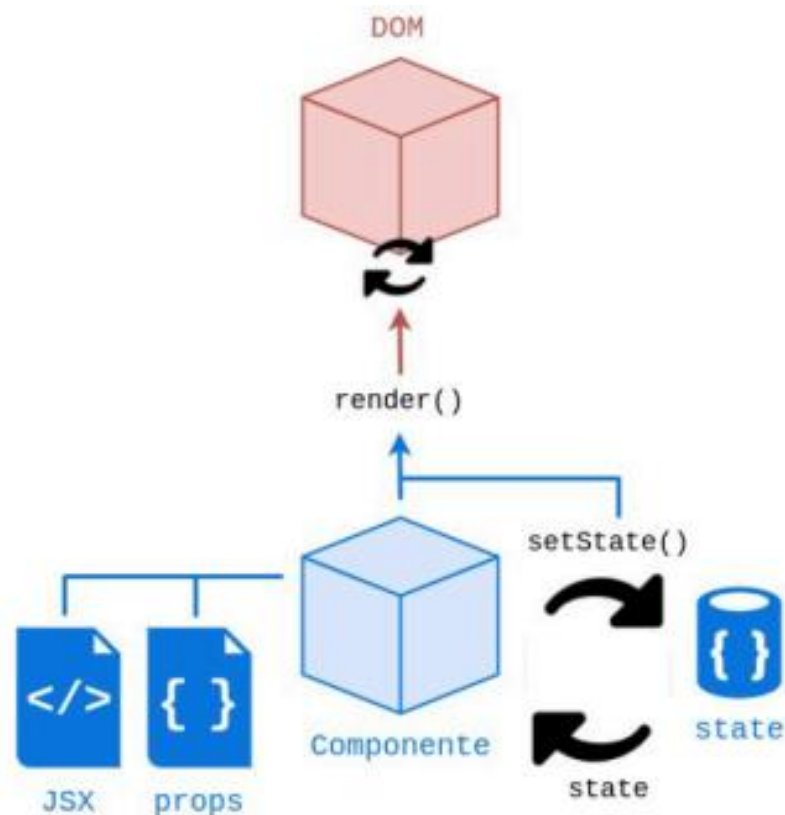
Durante a criação de interfaces com o *React*, podemos implementar diversos algoritmos e tecnologias para desempenho e reutilização da renderização de componentes de tela. Diferente de outras linguagens, ele trabalha com o DOM através do seu próprio Virtual DOM, um dos seus grandes benefícios junto com a atualização do estado de componentes. Ou seja, quando precisamos alterar um estado de um componente, a alteração é declarada no VDOM. E para obter um maior desempenho de renderização, o VDOM é mantido em memória sincronizado com a imagem do DOM real, atualizando apenas os componentes que mudaram de estado (PRUSTY, 2016).

A biblioteca também possui uma sintaxe própria chamada JSX, semelhante ao XML, que combina *JavaScript* e HTML. Ela surgiu como uma alternativa para escrevermos nosso *template* servindo para facilitar o uso da API do componente (HORTON; VICE, 2016).

2.3 React Native

React Native é uma biblioteca para construção de aplicativos móveis com base no framework originalmente definido pela biblioteca *React* (BANKS, 2017). Portanto, a especificação básica do *React Native* é a mesma que a especificação básica definida no *React*, ou seja, as aplicações são escritas declarativamente, seja usando a especificação funcional Javascript ou usando a especificação declarativa JSX 5. A estrutura para construir o aplicativo permanece a mesma. A Figura 1 mostra uma representação básica desse framework, destacando seus elementos. Vale ressaltar que com o *React Native*, as atualizações dos componentes são feitas diretamente na plataforma de execução.

Figura 1 - Elementos de um componente *React Native* definidos pelo *Framework*.



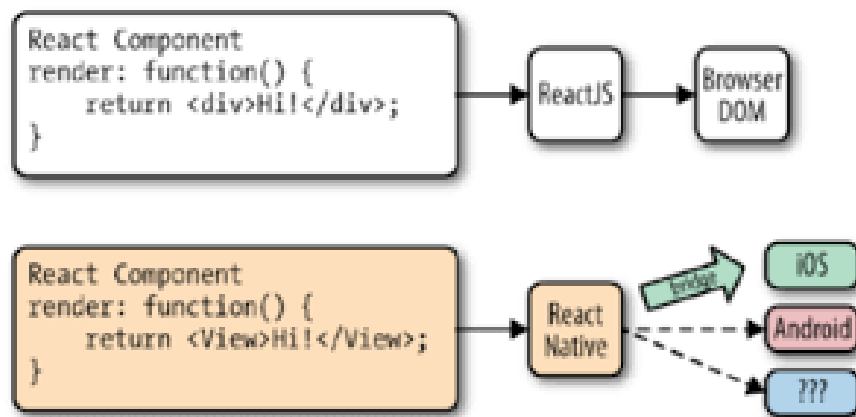
Fonte: Pontes,2018

Portanto, mesmo que seja necessário algum código específico da plataforma, partes do código podem ser compartilhadas entre plataformas. Uma estrutura de origem que permite que a comunidade de programação contribua para o seu desenvolvimento (Asp, 2018).

O *React Native* não usa *WebView*, o que significa que pode esperar um desempenho maior, pois pode usar os recursos nativos do dispositivo. O *React Native* roda em *JavaScript* como o *React*, e as ferramentas em que se baseia são ferramentas de desenvolvimento criadas pela equipe do Facebook (Eisenman, 2017).

No contexto acima, de acordo com Occhino (2015), engenheiro-chefe da equipe *React* do Facebook, o lema: "Aprenda uma vez, escreva em qualquer lugar" significa que os desenvolvedores devem usar o *ReactJS* para desenvolver experiências na web, e o mesmo vale para dispositivos móveis (Figura 2) .

Figura 2 - Comportamento da aplicação *React Native* em relação ao *React*



Fonte: EISENMAN, 2017.

Para desenvolver um aplicativo em *React Native*, nenhum IDE específico é necessário. A maneira mais fácil é usar o NPM (*Node Package Manager*), uma ferramenta de gerenciamento de pacotes *JavaScript* que permite exportar módulos e desenvolvê-los em um editor de texto ou IDE habilitado para *JavaScript* (Hansson, 2016; Vidhall, 2016).

2.3.1 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, 2021).

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).

2.3.2 Node.js

Node.js é um ambiente de execução Javascript do lado do servidor. Ou seja, através dele é possível criar aplicações Javascript para serem executadas no servidor, sem a necessidade de um browser para executá-las. (NODEJS, 2021).

Ele possui uma execução single-thread que não exige resposta a cada requisição, ele possibilita executar várias de conexões simultâneas, porque não aguarda o processamento da resposta, ou seja, é altamente escalável. (SOUZA, 2020).

Node.js é uma biblioteca usada do lado do servidor para diversas finalidades, principalmente para criar API baseadas em JSON 4 (*Javascript Object Notation*) para conexões com banco de dados (SOUZA, 2020). Permitindo que os dados sejam enviados para o *front-end* não havendo necessidade de diversas conexões que padronizam as informações facilitando que as informações cheguem ao usuário.(SOUZA, 2020).

Para a instalação do Node.Js e suas dependências e pacotes, necessita que se instale antes o gerenciador de pacotes NPM (*Node Package Manager*).

2.3.3 NPM

NPM que significa *Node Package Manager* é um gerenciador de pacotes que faz parte do Node.js. Ele permite instalar, desinstalar e atualizar dependências por meio de instrução na linha de comando. Sempre que é criado um projeto por meio dele, é adicionado um arquivo chamado *package.json*, que contém a relação dos pacotes instalados. (SOUZA, 2020).

Sendo assim, quando for necessário realizar alguma alteração, o NPM verifica o arquivo *package.json* e faz as atualizações necessárias de forma simples e rápida. Graças a isso mantém-se a organização do projeto e de suas dependências, além de evitar erros de configurações ao fazer a instalação de pacotes de forma manual. (SOUZA, 2020).

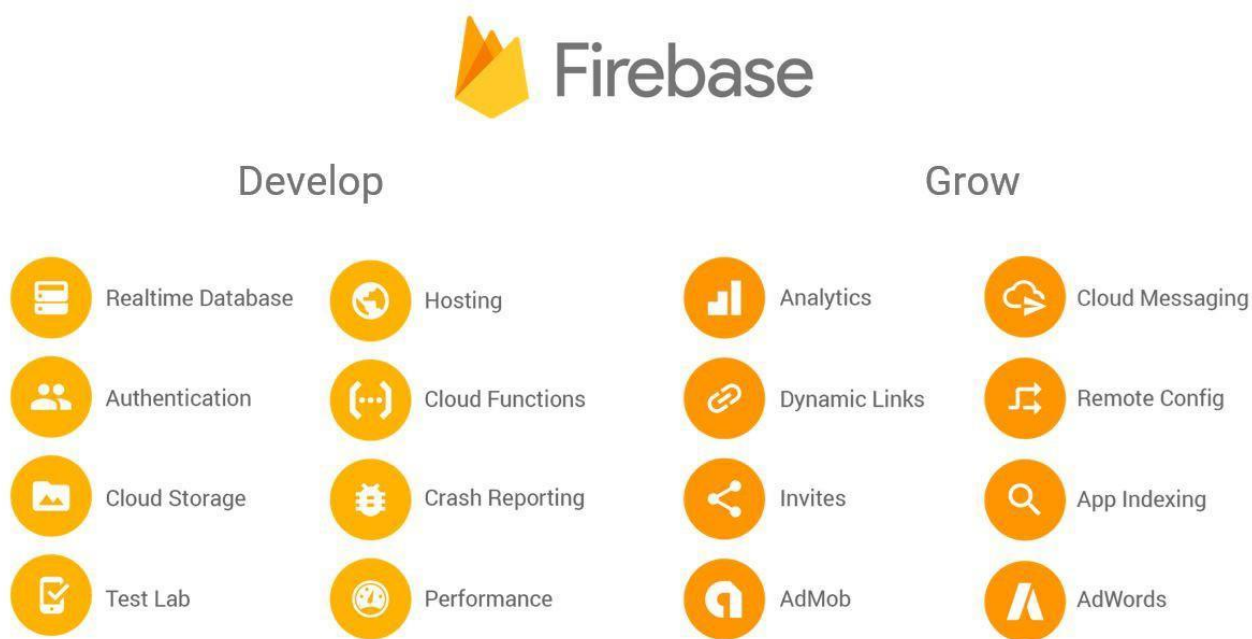
2.4 Firebase

Firebase é uma plataforma de desenvolvimento de aplicativos digitais lançada pelo Google. Com ele, é muito simples, rápido e fácil criar e estender aplicativos para Android, iOS e web (SILVA, 2021).

Além disso, seus recursos permitem melhorar o desempenho e o desempenho de seus aplicativos, torná-los mais seguros e fornecer uma experiência de usuário mais rica e completa. Com seus inúmeros recursos gratuitos, é usado como técnica de marketing digital para aumentar a base de usuários e gerar maiores benefícios econômicos.

Nesse sentido, o *Firebase* é considerado um *Backend-as-a-Service* (BaaS), um modelo de serviço que fornece toda a infraestrutura, com foco no funcionamento interno do software, como sistemas, bancos de dados, envio e recebimento de informações, armazenamento, entre outros.

Isso quer dizer que o profissional não precisará desenvolver todo o sistema de forma manual, uma vez que o *Firebase* oferece esse serviço de forma mais automatizada (CONTENT, 2019).

Figura 3 - Serviços disponibilizados pelo *FireBase*

Fonte: SINGH, 2018

2.4.1 Firebase Authentication

O *Firestore Authentication* fornece serviços de *back-end*, *Kits* de Desenvolvimento de *Software* (SDKs) fáceis de usar e bibliotecas de interface de usuário (IU) prontas para autenticar usuários no seu aplicativo. Ele oferece suporte à autenticação usando senhas, números de telefone, provedores de identidade federados conhecidos, entre outros. O SDK do *Firestore Authentication* disponibiliza métodos para criar e gerenciar usuários que utilizam os próprios endereços de e-mail e senhas para fazer login com as Contas do Google, Facebook, Twitter e GitHub (FIREBASE, 2020).

O *Firestore Authentication* é estreitamente integrado a outros serviços do *Firestore* e aproveita os padrões do setor, como OAuth 2.0 e OpenID *Connect*, para que possa ser facilmente integrado ao seu *back-end* personalizado. Com isso, será utilizado o *Firestore Authentication* para obter e validar as informações do usuário no seu *login* do JetList.

2.5 MySQL

O MySQL é um sistema de gerenciamento de banco de dados relacional de código aberto (RDBMS) suportado pela Oracle e baseado em linguagem de consulta estruturada (SQL). O sistema funciona em praticamente todas as plataformas, incluindo Linux, UNIX e Windows.

O sistema é um componente importante de um conjunto de ferramentas com código aberto chamada LAMP, que usa Linux como sistema operacional, Apache como servidor *web*, MySQL como sistema de gerenciamento de banco de dados relacional e PHP, Perl ou Python como linguagem de script orientada a objetos (TECMUNDO,2021).

Embora possa ser usado em uma ampla variedade de aplicativos, o gerenciador de banco de dados é mais frequentemente associado a aplicativos da *web* e à publicação *online*. Assim, por trás de muitos dos principais *sites* do mundo, a ferramenta usada é o RDBMS, assim como de inúmeros aplicativos corporativos e de consumo baseados na *web*, como Facebook, Twitter e YouTube.

Originalmente concebido em 1995 pela empresa sueca MySQL AB, o programa foi adquirido pela Sun Microsystems em 2008, que, por sua vez, passou para o controle da Oracle em 2010. Os desenvolvedores podem usar o *software* de forma gratuita, mas as empresas devem obter uma licença comercial.

O SGBD (Servidor e Gerenciador de Banco de Dados) escolhido para este trabalho em questão foi o MySQL, devido a vários fatores, entre eles a vantagem de sua licença de uso ser de código aberto, sua confiabilidade e sua capacidade de processamento sendo bem similar às de versões de SGBDs pagos. “Por possuir essas características, a ferramenta é indicada para uso em aplicações em todas as áreas de negócio, independentemente do tamanho da sua aplicação” (MILANI, 2007).

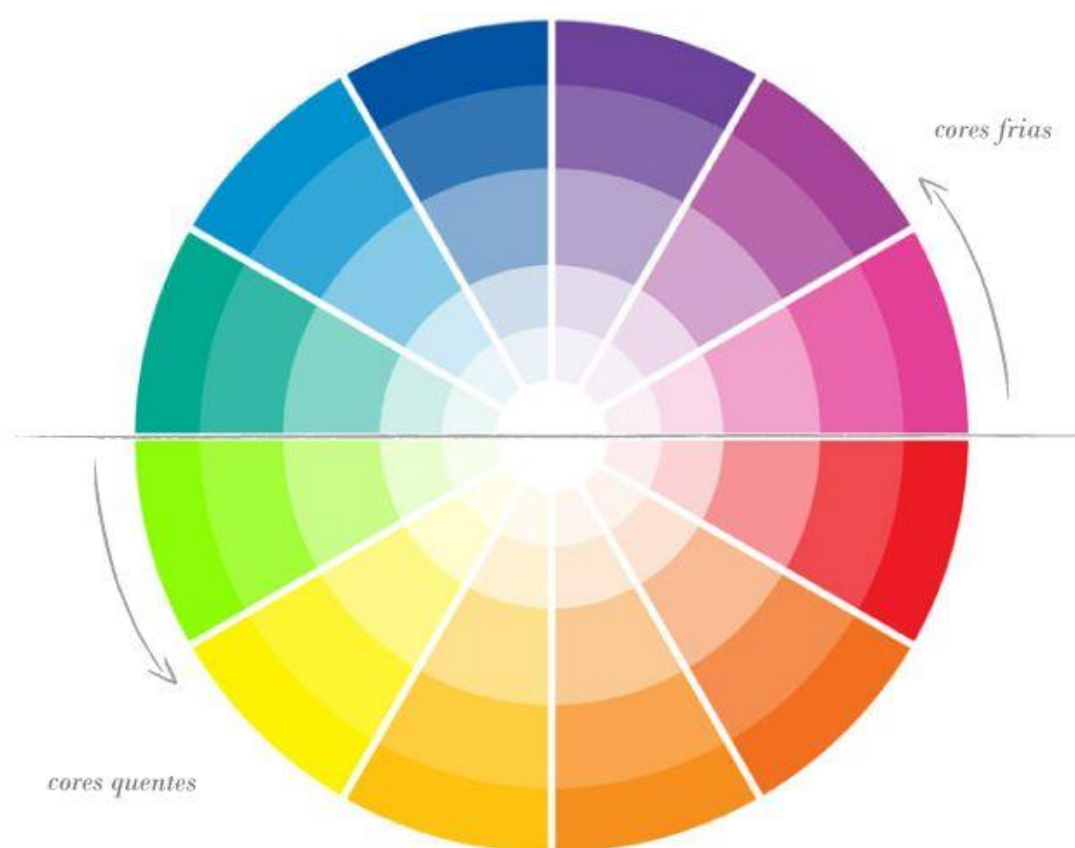
2.6 Teoria das Cores

A Teoria das Cores é o estudo das cores, desde como ela é interpretada pelo nosso cérebro até a utilização em peças gráficas para a comunicação visual. Na teoria das cores podemos entender como a cor age no ser humano e como podemos utilizar isso a nosso favor, manipulando as cores para passar uma determinada mensagem ou transmitir determinada sensação.

Sabendo disso, utilizaremos de forma favorável, o conhecimento da teoria das cores em nosso projeto, utilizando as cores para passar uma determinada mensagem ou transmitir sensações. As cores são divididas em duas temperaturas, são elas cores quentes e cores frias. As cores quentes são as que o amarelo e o vermelho estão presentes em sua maioria (Figura 4) .

São assim denominadas pois geram uma sensação de calor, sendo elas associadas ao sol e ao fogo. As cores frias são as que o verde e o azul predominam (Figura 4). Relacionadas ao gelo, água e frio, geram sensações tranquilas, calmas e frescas (ARTY, 2018).

Figura 4 - Tabela de Colorimetria cores quentes e frias



Fonte: O Autor.

2.7 Cromoterapia

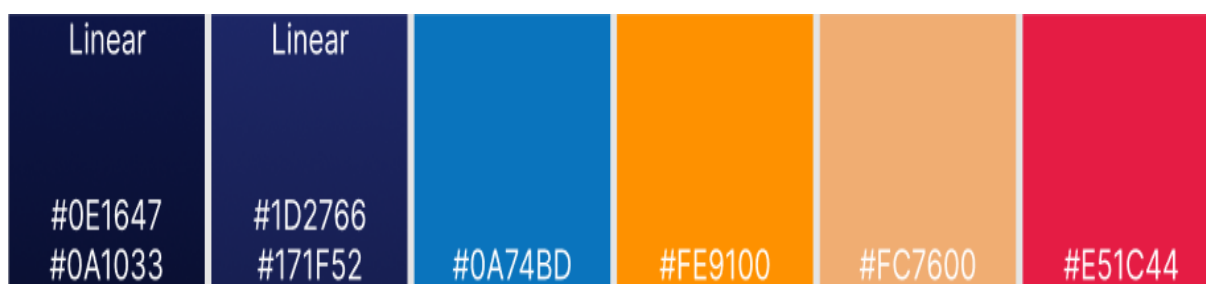
A cor tem sido estudada em vários campos, como arquitetura, marketing e design. Quando o usuário faz a escolha certa, eles beneficiam o usuário e podem

melhorar a leitura em cerca de 40%, o aprendizado em 55% a 68% e a compreensão em 73% (CARVALHO, 2013). No campo da medicina alternativa, às pesquisas que visam ajudar no tratamento de doenças por meio do uso da cor são conhecidas como cromoterapia, que, assim como a medicina tradicional, busca a saúde física e mental. Essa técnica é utilizada desde a antiguidade por grandes civilizações como Pérsia, Egito e China (VALNET, 2019).

Os seres humanos são afetados por três aspectos fundamentais: físico, cognitivo e psicológico. A adequada correlação desses fatores permite projetar ambientes seguros, confortáveis e produtivos (KOTH, 2013). A combinação de luz natural e luz artificial pode trazer boa sensação aos pacientes, reduzir a dor, Como importante ferramenta de apoio à saúde (BIANCHI; DAVID; SUETA, 2017), e em geral, pode gerar estados mentais nos indivíduos que respondem aos seus desejos ou ações (MANAIA, 2013).

Levando em conta os conceitos apresentados no tópico atual e no anterior, a paleta de cores definida para o aplicativo, busca trazer uma sensação tranquila e fresca utilizando cores predominantes frias (Figura 5).

Figura 5 - Paleta de cores utilizado no aplicativo



Fonte: O Autor.

3 MATERIAIS E PROCEDIMENTOS METODOLÓGICOS

Este capítulo tem por finalidade apresentar os procedimentos de soluções para este trabalho na qual é realizado no Trabalho de Conclusão de Curso II. Com isso, serão apresentadas as ferramentas que foram utilizadas para implementar o aplicativo nos sistemas Android/IOS, tendo como base os conceitos e definições apresentados no capítulo anterior.

3.1 Métodos

Para Wazlawick (2020), a pesquisa experimental se dá pela manipulação de uma parte da realidade do pesquisador, como por exemplo, esse trabalho que busca auxiliar uma melhor forma de agendamento simples e eficácia, entre o fornecedor e cliente.

“A pesquisa documental, por outro lado, consiste na análise de documentos ou dados que não foram ainda sistematizados e publicados.” (WAZLAWICK, 2020, p.23.).

Para Gil (2019), uma pesquisa experimental depende de um objeto de estudo, variáveis que podem manipulá-lo, formas de controle das variáveis e formas de observação dos efeitos que as variáveis produzem no objeto.

Segundo os procedimentos técnicos esta pesquisa é bibliográfica, na qual terá uma investigação para desenvolver os devidos experimentos. Como um trabalho de pesquisa bibliográfica, este projeto é baseado em monografias, artigos, teses (WAZLAWICK, 2020).

“A pesquisa bibliográfica é um passo fundamental e prévio para qualquer trabalho científico, mas ela em si não produz qualquer conhecimento novo. Apenas supre o pesquisador de informações públicas que ele ainda não possuía.” (WAZLAWICK, 2020, p.23.).

Wazlawinck (2020), sugere que a pesquisa bibliográfica deve seguir os seguintes passos:

1. Listar os títulos de periódicos e eventos relevantes para o tema de pesquisa e os títulos de periódicos gerais em computação que eventualmente possam ter algum artigo na área do tema de pesquisa.
2. Pesquisar artigos publicados a pelo menos cinco anos atrás.
3. Selecionar da lista, títulos relacionados ao tema a ser pesquisado.

4. Ler o abstract dos artigos selecionados e classificá-los em relevância “alta”, “média” ou “baixa”.
5. Ler artigos com alta relevância e fazer resumos com os principais assuntos aprendidos sobre o tema. Anotar títulos que podem ser mencionados na bibliografia mesmo que tenham mais de cinco anos.
6. Se necessário, ler artigos de relevância média ou baixa, porém sempre priorizar os artigos com alta relevância.
7. Aluno decide se já possui material suficiente para elaborar uma pesquisa consistente.
8. Se precisar aumentar a pesquisa lendo artigos mais antigos (Expandindo o passo 2) ou periódicos com grau de relevância menor (Expandindo o passo 1).

Após ter sido realizado estes procedimentos, com os dados levantados e mapeados, o desenvolvimento do aplicativo foi realizado.

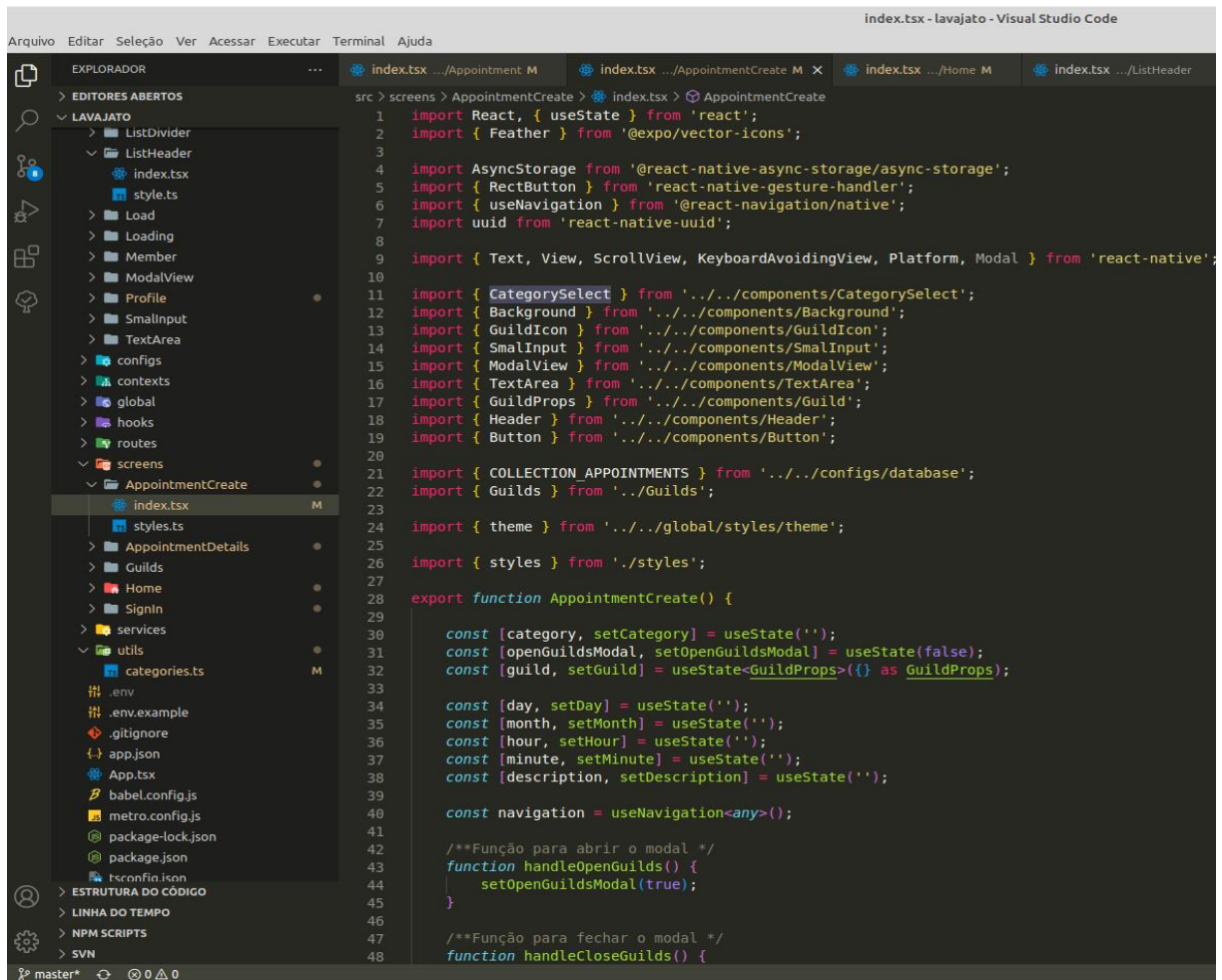
3.2 Materiais

3.2.1 Visual Studio Code

Em 2015 foi lançado pela Microsoft um editor de código destinado ao desenvolvimento de aplicações *web* chamado de *Visual Studio Code*, ou simplesmente VSCode. Anunciada durante o Build, evento voltado a desenvolvedores que ocorre nos Estados Unidos anualmente, trata-se de uma ferramenta leve e multiplataforma que está disponível tanto para Windows, quanto para Mac OS e Linux e atende a uma gama enorme de projetos, não apenas ASP.NET, como também Node.js.

Convém mencionar, ainda, que essa ferramenta conta com suporte a muitas e diversas linguagens de programação e é *open source*. Isso quer dizer que o seu código é aberto e qualquer pessoa pode contribuir com melhorias para o seu desenvolvimento.

Figura 6 - Visual Studio Code (VSCode)



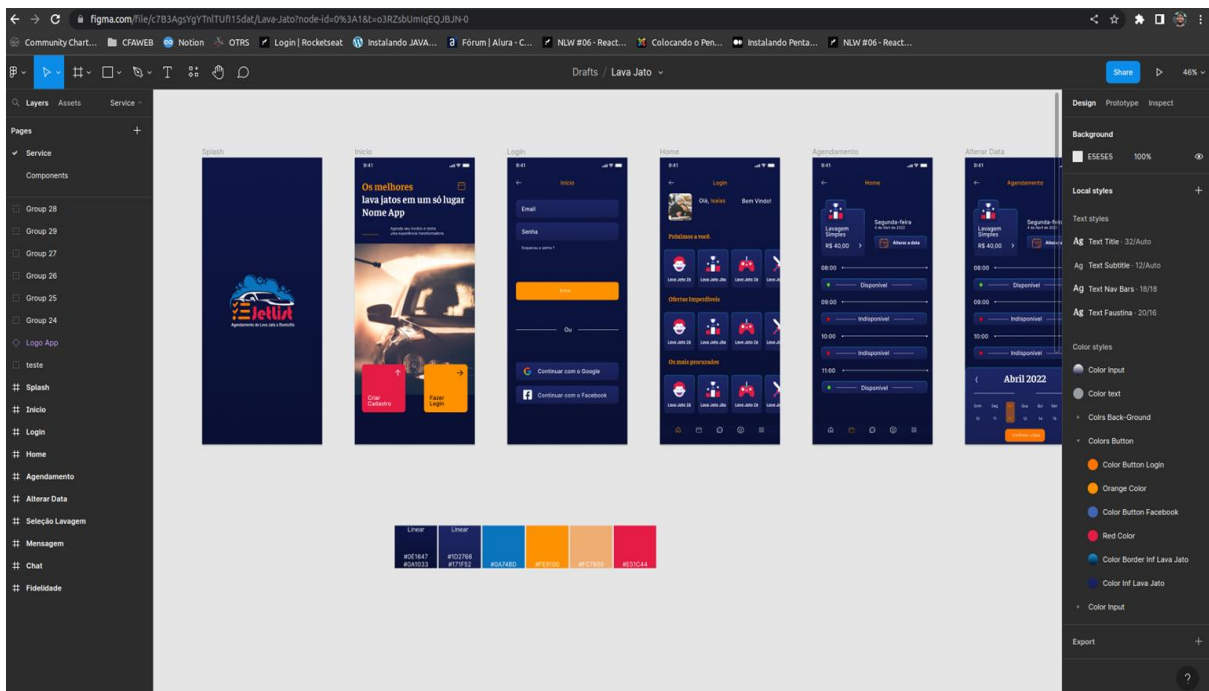
Fonte: O Autor

3.2.2 Figma

Figma é um editor online de *design* gráfico focado na criação de interfaces gráficas e experiência de usuário com ênfase colaborativa. A ferramenta permite que uma equipe de designers tenha acesso simultâneo e trabalhe em cima de um mesmo projeto ao mesmo tempo, facilitando a vida de times com profissionais em trabalho remoto em diversas localidades diferentes.

Na figura 7 é mostrado a tela do programa, onde ocorreu a criação do layout da aplicação desenvolvida.

Figura 7 - Figma



Fonte: O Autor

3.2.3 Android Studio

O Android Studio é uma IDE que foi criada pela Google. Esse ambiente de desenvolvimento possui várias funcionalidades, tais como preenchimento automático de comandos, auto indentação, navegador de pasta, simulação de dispositivo, dentre outras funcionalidades (HELLMANN, 2016).

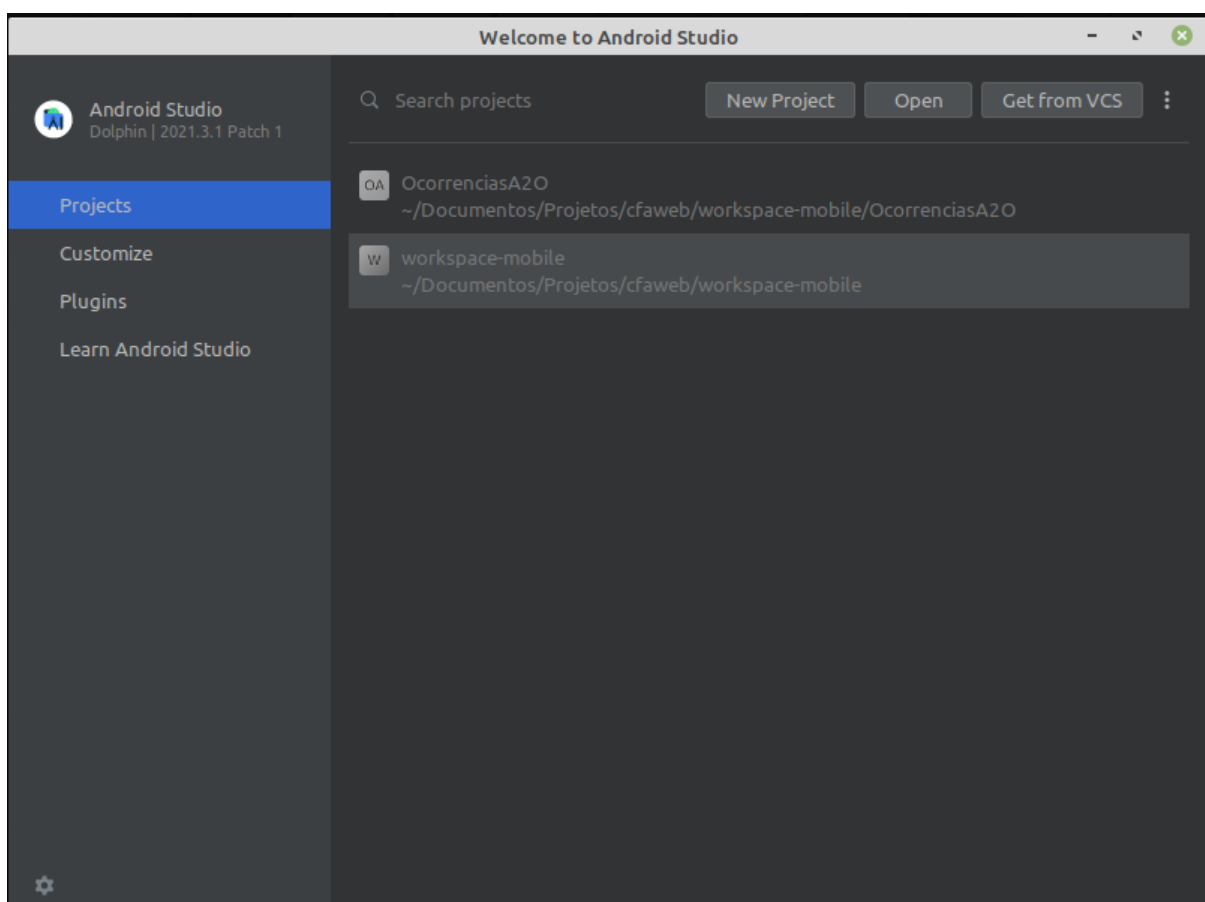
O Android Studio oferece um ambiente de desenvolvimento robusto, com compatibilidade de simulação a vários dispositivos móveis, e com esta proposta foi utilizado as funcionalidades de seus emuladores, para ser desenvolvido o aplicativo.

De acordo com o guia de desenvolvimento disponibilizado pelos próprios desenvolvedores da IDE, o Android Studio é o ambiente de desenvolvimento integrado oficial para o desenvolvimento de apps Android e é baseado no IntelliJ IDEA. Além do editor de código e das ferramentas de desenvolvedor avançadas do próprio IntelliJ, o Android Studio oferece ainda mais recursos para aumentar sua produtividade na criação de apps Android, tais como:

- Um sistema de compilação flexível baseado em Gradle;
- Um emulador rápido com inúmeros recursos (principal foco de sudo);

- Um ambiente unificado que possibilita o desenvolvimento para todos os dispositivos Android;
- A aplicação de mudanças para enviar mudanças de código e recursos ao aplicativo em execução sem reiniciar o app;
- Frameworks e ferramentas de teste cheios de possibilidades;

Figura 8 - Tela de início do Android Studio



Fonte: O Autor

3.2.4 Computador Utilizado

Em relação ao computador utilizado, será apresentado suas características para produção do projeto. Para isso, foi utilizado o sistema operacional (SO) Linux Mint, na sua versão 19.3 Cinnamon, um SO de código aberto baseado no Debian e tendo o GNOME como ambiente de desktop. A tabela 1 mostra os detalhes.

Tabela 1: Características do computador utilizado

MARCA	DELL
MODELO	DELL INSPIRON 15 - 557
SISTEMA OPERACIONAL	LINUX MINT 19
PROCESSADOR	Intel© Core™ i7-6500U CPU @ 2.50GHz x 2
MEMÓRIA RAM	16 GB
TIPO DE SISTEMA	64 BITS
PLACA DE VÍDEO	Intel Corporation Skylake GT2 [HD Graphics 520]
KERNEL DO LINUX	5.4.0-126-generic
DISCOS RÍGIDOS	240.4 GB

Fonte: O autor.

4 DESENVOLVIMENTO

Neste capítulo será apresentado o funcionamento do aplicativo, ilustrado através da navegação de suas diferentes telas, funcionalidades e prototipagem por meio de demonstrações e de conceitos citados anteriormente, de forma clara e visualmente intuitiva. Como mencionado, a prototipagem foi realizada no Figma, possibilitando o vislumbre visual do aplicativo e suas características. As próximas subseções irão discutir esses assuntos.

4.1 Prototipação

A prototipação é a construção de um modelo representativo da aplicação. Com ela é possível identificar funcionalidades, componentes a serem usados e o fluxo que se irá seguir.

Com a visualização geral do aplicativo que a prototipação proporciona, torna-se mais fácil identificar problemas de usabilidade, fluxo determinado, experiência de usuário, componentes a serem criados entre outros.

Conforme mencionado no capítulo anterior, o desenvolvimento do protótipo se deu por meio do Figma, ferramenta com ênfase na prototipagem de interfaces 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 de serem desenvolvidas.

4.2 Linguagem Utilizada

Como apresentado, o React Native foi escolhido como a linguagem de desenvolvimento, por ser atual, versátil, com vários componentes nativos e a possibilidade de se ter apenas um único código que será executado em ambas as plataformas IOS e Android.

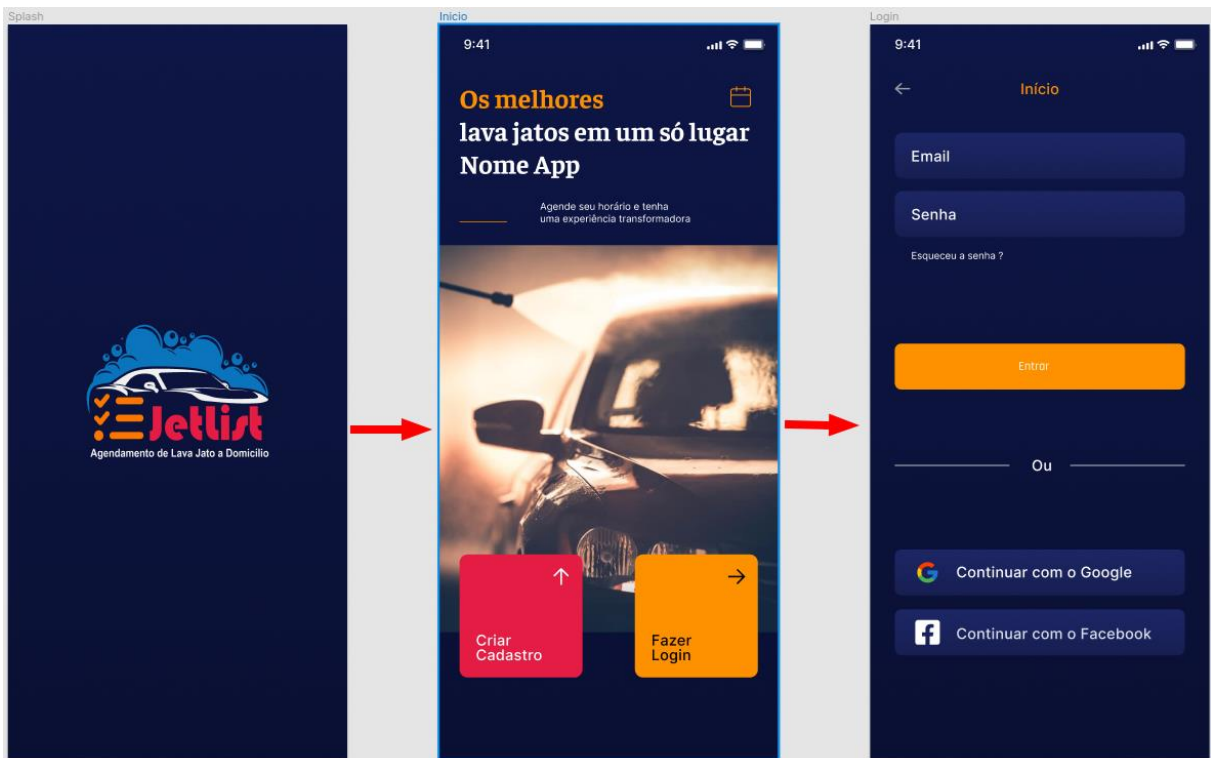
Em relação ao gerenciamento de pacotes, o NPM faz com bastante simplicidade. Para instalar ou desinstalar bibliotecas é bem simples. Fazendo com que, não seja necessário ficar alterando dependências. Após a instalação de um novo pacote, é só importar no projeto e começar a utilizá-lo.

4.3 Fluxograma

A seguir será apresentado um fluxograma do protótipo realizado no Figma. Com a maturação do desenvolvimento foi-se percebendo a necessidade de componentização, pois vários itens eram repetidos em diferentes telas. Uma vantagem da prototipação, pois consegue-se enxergar com mais facilidade o processo para o desenvolvimento.

Lembrando que durante essa etapa, a escolha de cores, e tonalidade foi baseada no estudo de Cromoterapia, aqui apresentado. Levando em conta as cores frias para uma melhor experiência de usuário. Com isso todos os componentes criando seguem o mesmo padrão de cores e gradientes, que sempre começa de um tom mais claro na parte superior e vai escurecendo aos poucos nas partes inferiores. Conforme poderá ser observado na Figura 9.

Figura 9 - Fluxograma de login



Fonte: O Autor

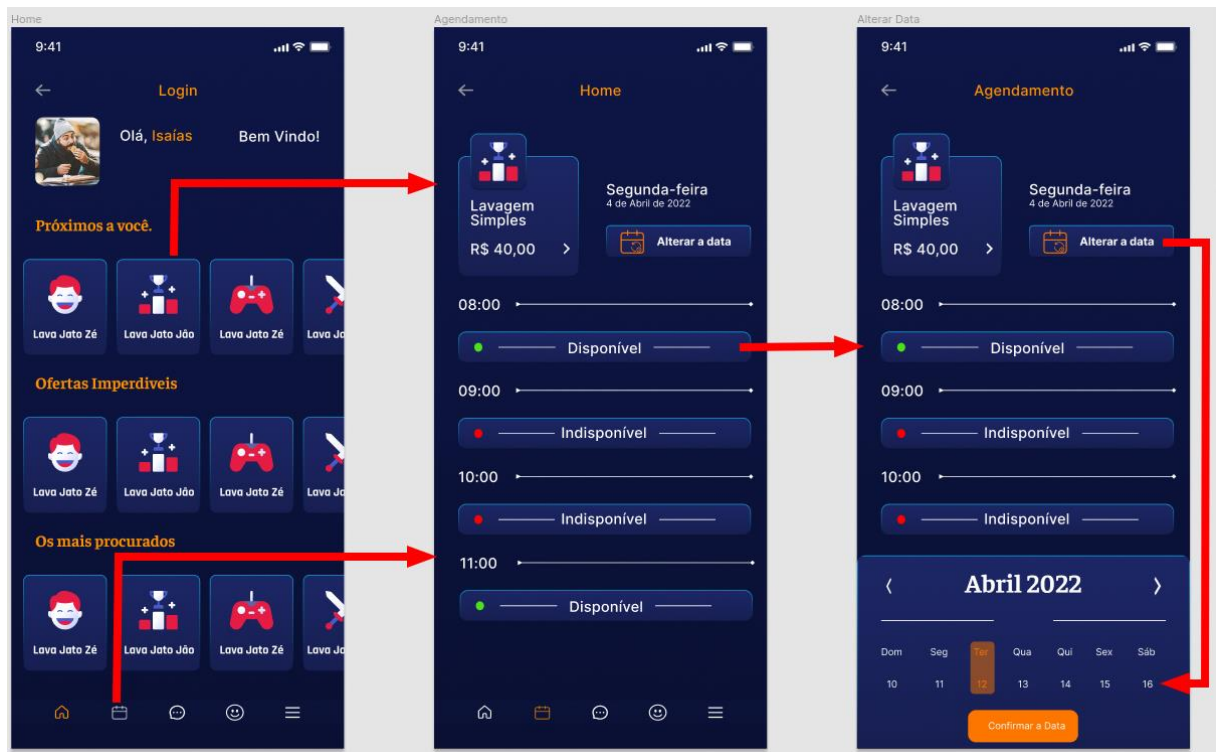
A primeira parte é o carregamento do aplicativo juntamente com a logo. Lembrando que a logo tem as mesmas tonalidades aplicadas nas telas (Figura 9). Isso foi pensado para a identidade visual ser uma só e não causar confusão no usuário.

Já a tela central e da direita dão continuidade ao fluxo que o usuário vai seguir, ao solicitar o *login*, será redirecionado para a tela da direita, tela de login. Onde existem as opções de logar com um *e-mail*, recuperar senha ou conectar junto com a conta do Google ou do Facebook.

Seguindo adiante, será listado os fornecedores divididos em categorias. A ideia dessa listagem foi inspirada no aplicativo Spotify, que possui o mesmo estilo de listas na horizontal.

No final de cada página foi adicionada uma barra chamada de *Footer* (Figura 10), onde são botões de navegação, caso o usuário deseje ir para uma determinada tela mais facilmente. Atentando-se que cada botão fosse intuitivo, evitando de causar dúvidas no usuário sobre qual seria a ação do botão.

Figura 10 - Fluxograma de agendamento

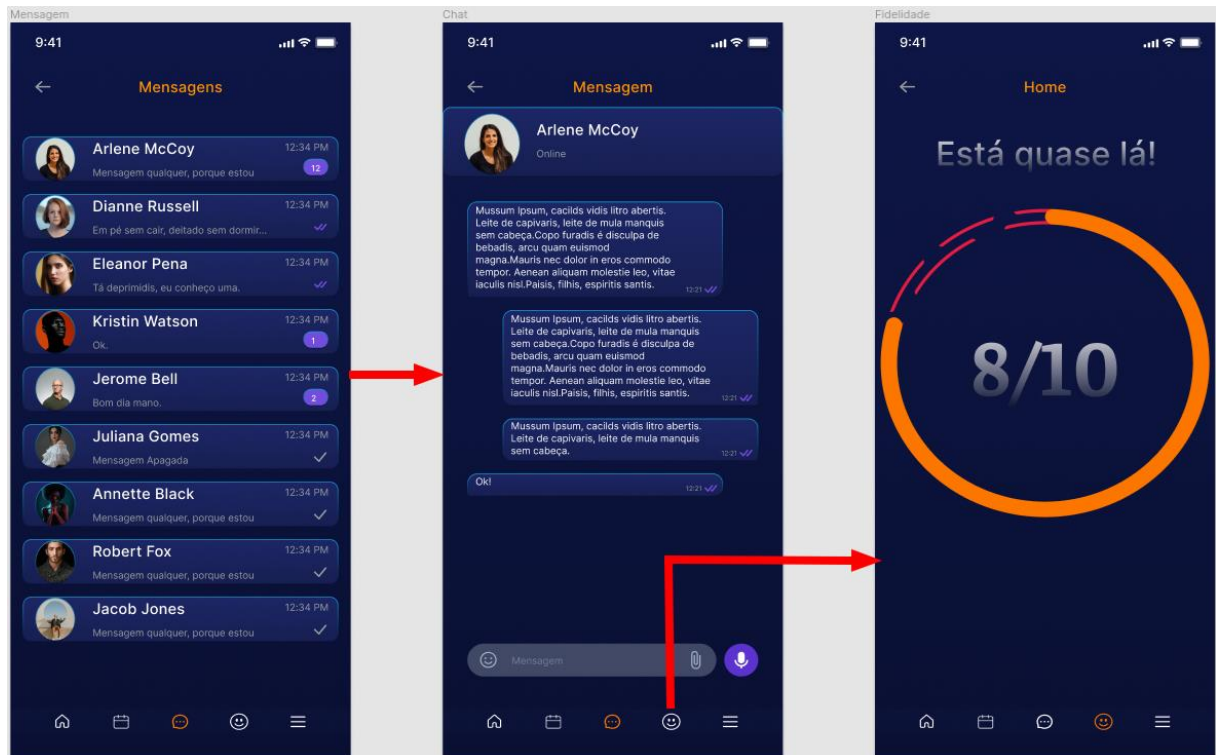


Fonte: O Autor

Seguindo, pois a ideia é que o aplicativo seja mais simples possível para executar sua proposta de agendamento. Foi criada a opção para troca de mensagens entre cliente e fornecedor, com o intuito de melhor comunicação entre ambas as partes, seja lá se o fornecedor for atrasar um pouco, ou se o cliente deseja falar alguma observação específica na hora da limpeza de seu veículo (Figura 11).

E por fim, um programa de bonificação para o cliente que o fornecedor pode ou não fornecer. E como seria? O cliente a cada vez que realizasse um agendamento com determinado fornecedor ganharia um ponto. E ao final, quando chegasse a marca estabelecida, ganharia uma lavagem de brinde. Uma forma de incentivar e fidelizar o cliente com o fornecedor (Figura 11).

Figura 11 - Fluxograma de mensagens e premiação



Fonte: O Autor

Com isso pode-se imaginar o que será necessário para o desenvolvimento do aplicativo, suas funcionalidades, regras de negócios e mais. O Figma permite com muita facilidade desenvolver os protótipos, fácil de inspecionar e desenvolver, pois todas as suas funcionalidades são bastantes intuitivas.

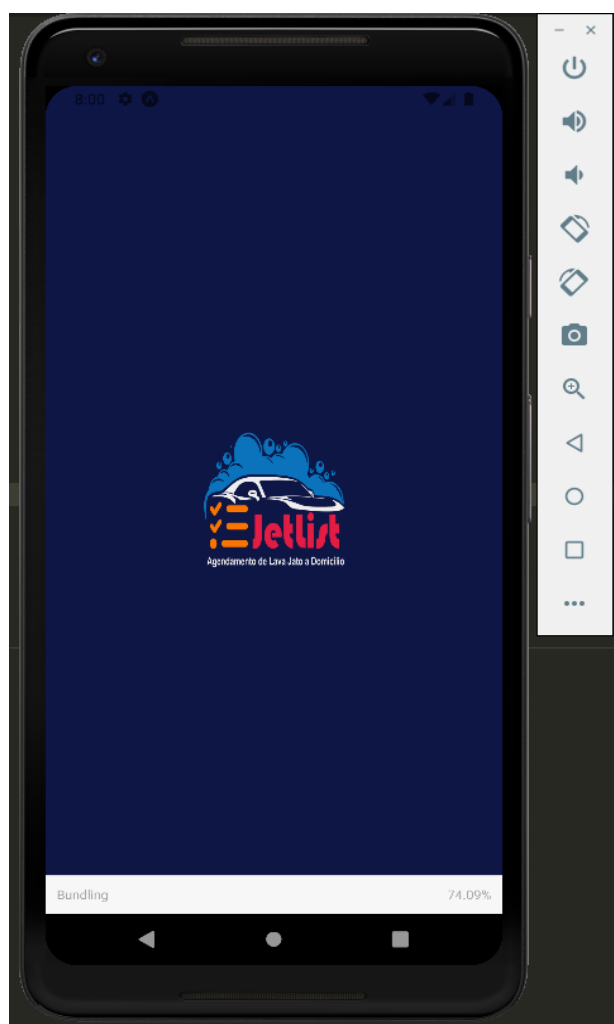
5 RESULTADOS ESPERADOS

Neste capítulo será apresentado o funcionamento do aplicativo, juntamente com suas funcionalidades e regras desenvolvidas.

5.1 Tela de Splash

A tela de *Splash* funciona como uma tela de carregamento antes do aplicativo ser executado. O intuito com essa tela é fazer o carregamento de fontes da google, que não são nativas das bibliotecas do React Native, caso o usuário não tenha em seu dispositivo para não ocorrer erros ao logar, ou não ter a experiência desejada, conforme demonstrado na Figura 12. Esse processo de baixar fontes é realizado somente na primeira vez de acesso.

Figura 12 - Tela de Splash

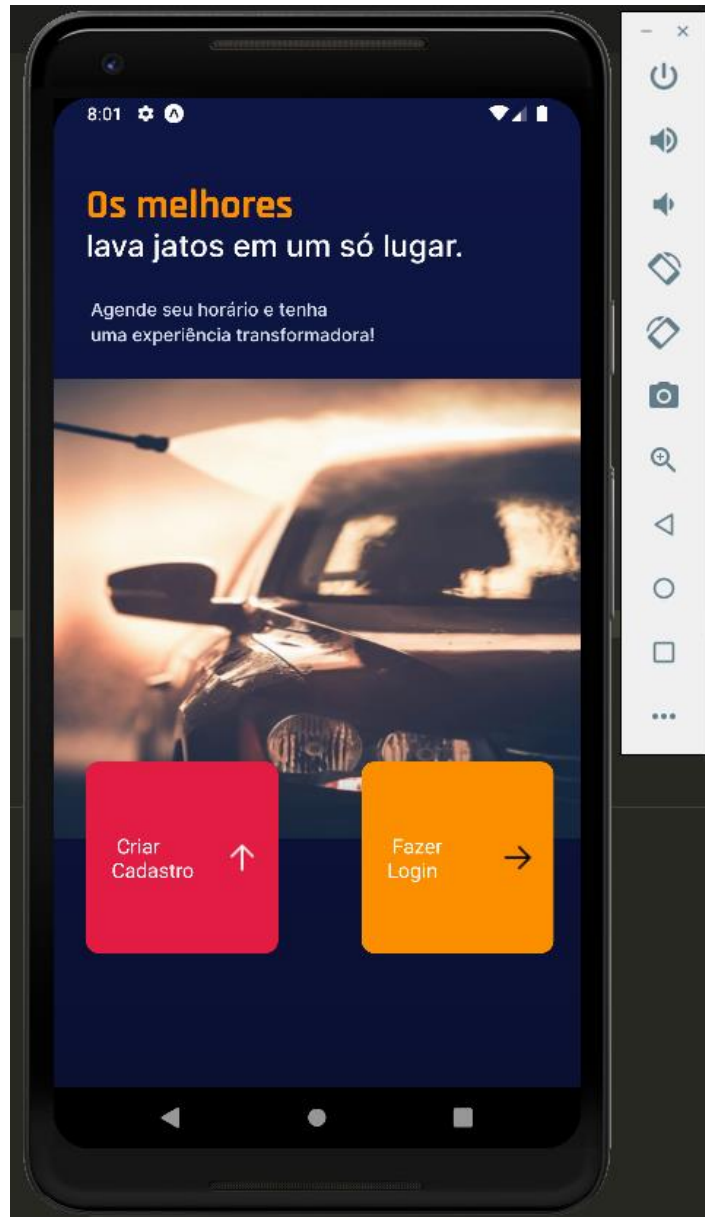


Fonte: O Autor

5.2 Tela Inicial

Esta é a tela de apresentação, ou também chamada de tela de início, para o usuário, onde será solicitado o fornecimento e a autorização de seus dados de *Login*. Conforme demonstrado na Figura 13.

Figura 13 - Tela de Início

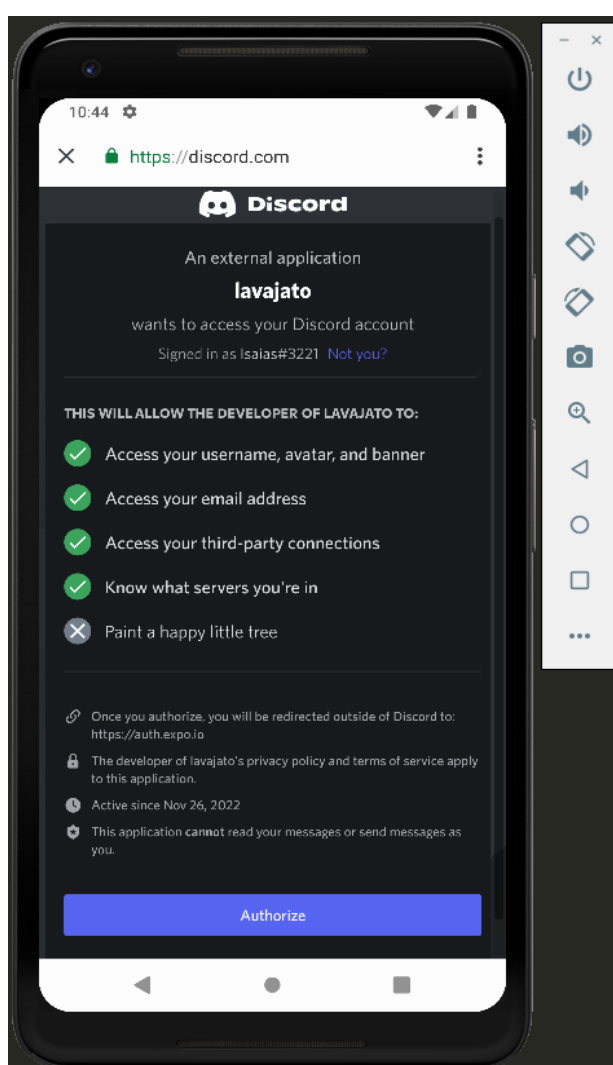


Fonte: O Autor

Conforme a imagem anterior, já pode ser observado nos títulos em destaque e em cores e tamanhos diferentes, o uso das fontes da biblioteca do google citada no tópico anterior, que foram carregadas após passar pela Splash.

OAuth 2 é um protocolo de autorização que permite que uma aplicação se autentique em outra. Para que isso aconteça, uma aplicação pede permissão de acesso para um usuário, sem que para isso ela tenha acesso a alguma senha dele. O usuário pode conceder ou não o acesso à aplicação (Figura 14). Depois da permissão ser aceita, caso o usuário precise alterar a senha de acesso, a permissão continuará válida para a aplicação e, caso necessário, a permissão dada à aplicação pode ser revogada a qualquer momento também (GUEDES, 2022).

Figura 14 - Tela de confirmação do Discord



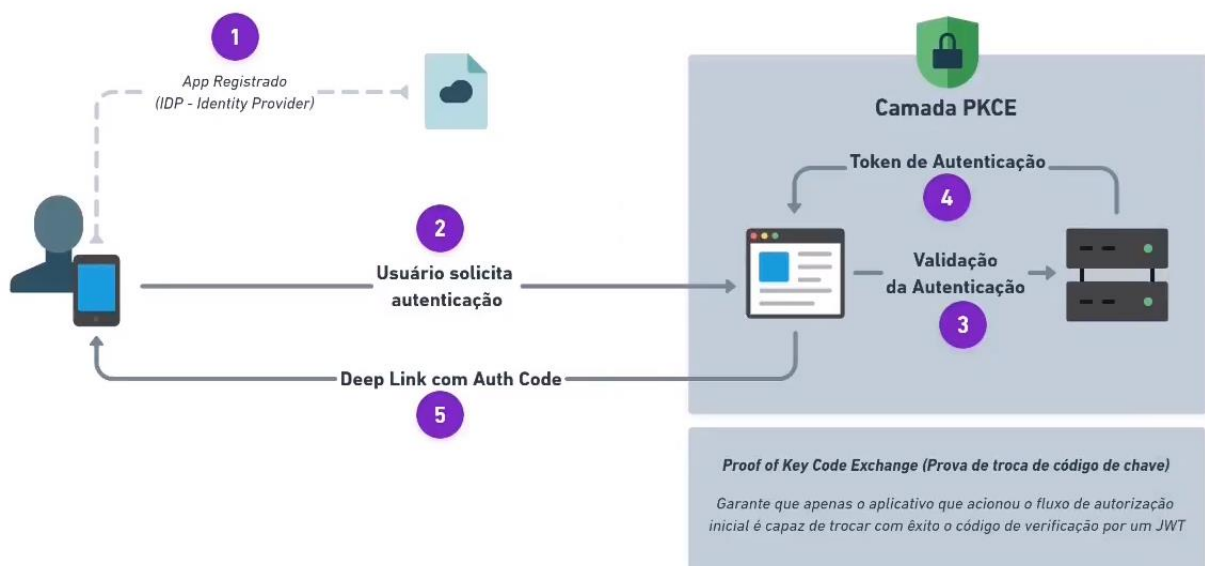
Fonte: O Autor

Esta autenticação pode ser feita por diversos aplicativos, Facebook, Google, Spotify entre outros. Primeiro é feito o registro no servidor do aplicativo escolhido, no caso Discord, para validar que quando for solicitado acesso, saber onde é a origem.

Posteriormente feito a requisição, será aberto o navegador onde se é possível informar seus dados pessoais, se já tiver feito essa verificação, retornará para a aplicação com os dados, caso contrário buscará no servidor de destino as permissões (Figura 15) e retornará para a aplicação.

Portanto, com o fim desse processo (Figura 15) , é recuperado um objeto contendo as suas informações, que será armazenada localmente em seu dispositivo. A partir disso, todo o restante da aplicação faz o uso dessas informações. Como por exemplo, sua foto, nome, servidores que participa, entre outros. E também sua foto, nome, servidores que participa, entre outros.

Figura 15 - Processo de autenticação OAuth 2



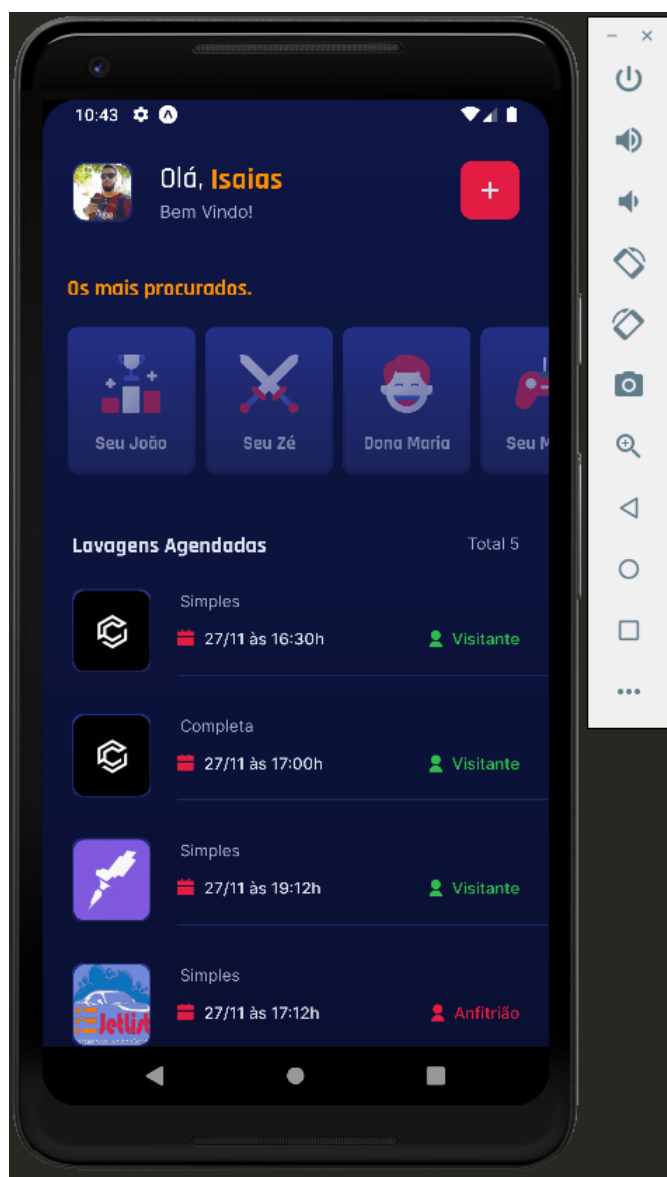
Fonte: O Autor

5.3 Tela Home

Na tela *home* é apresentada a lista de lava jatos, juntamente com seus respectivos agendamentos. Se não for selecionado nenhum lava jato da *CategorySelect*, componente criado para fazer listagem e ser reaproveitado no projeto, irá mostrar todos os agendamentos feitos nos lava jatos. Quando selecionado o lava jato desejado, mostrará somente os agendamentos referentes ao selecionado. Já a listagem de agendamentos é feita por outro componente, esse nativo do *React Native*, chamado de *FlatList*, que é indicado para listagem na vertical.

Há também o campo texto que mostra a quantidade de agendamentos, que foi montada dinamicamente. Esta busca é feita dentro do *ListHeader*, componente responsável por mostrar esses textos antes da *FlatList*, conforme visto na figura 16.

Figura 16 - Tela home



Fonte: O Autor

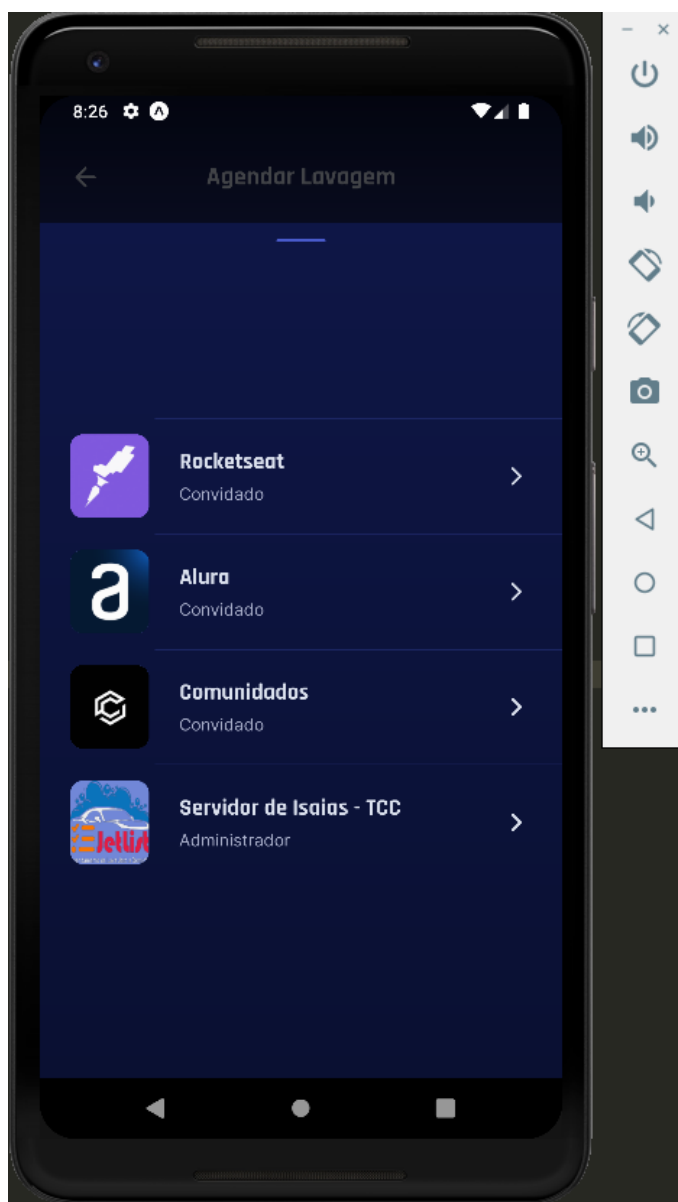
No menu superior, também visto na figura 16, pode-se observar o componente de Profile, que dentro dele tem o Avatar, outro componente reutilizável no restante da aplicação, e um outro componente, para agendar os lava jatos através do botão de mais.

5.4 Tela de Agendamento

No agendamento, é possível escolher o tipo de lavagem, o fornecedor, os horários e uma descrição. Quando selecionado o tipo de lavagem desejado, uma propriedade do *CategorySelect*, *hasCheckBox*, é passada como verdadeira, para que o usuário possa saber que o item foi selecionado. Sendo assim diferenciando da primeira listagem da *Home*.

Seguindo adiante, fica disponível a opção para selecionar um fornecedor de sua escolha. Esta ação dispara uma função para chamar o *ModalView* (Figura 17), também componente, que irá listar os fornecedores disponíveis. Nesta parte de listar os fornecedores, foi criada uma outra função que a partir dos dados do usuário obtidos pela validação do Discord, pega todos os servidores que o usuário faz parte, fazendo com isso uma lista dinâmica e real, aproveitando os conhecimentos obtidos para montar o mesmo.

Figura 17 - Listagem de Fornecedores

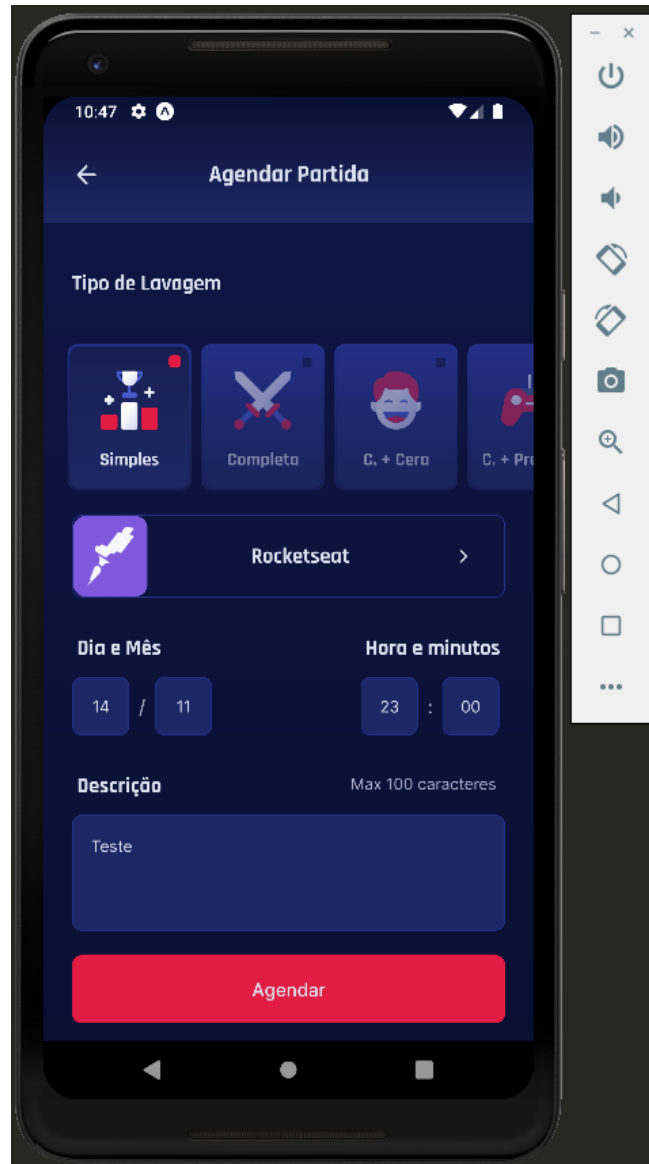


Fonte: O Autor

Por fim, antes de finalizar o agendamento, é necessário adicionar dia e mês, horas e minutos e uma descrição (Figura 18). Feito em cada campo, validações para aceitar números nos quatros primeiros campos, e *string* no campo de descrição, colocando limites de caracteres em cada opção também.

O botão de agendar é outro componente que vem sendo utilizado em quase todas as telas, porém especificamente por causa de seus formatos em telas distintas e não se adequar bem o estilo quando lhe é passado por parâmetro, foi criado mais de um componente de botão, para suas diversas funcionalidades.

Figura 18 - Tela de Agendamento



Fonte: O Autor

CONSIDERAÇÕES FINAIS

No início deste trabalho, conforme apresentado na introdução, identificou-se a crescente demanda do uso de aplicativos móveis e o quanto cada dia mais nossas necessidades e rotinas são afetadas por eles. Com o advento da pandemia, os serviços a domicílios cresceram, surgindo para um nicho, uma carência por aplicações voltadas para o agendamento de serviços de lavagem de carros a domicílio.

Portanto, o presente trabalho teve como objetivo analisar a viabilidade do desenvolvimento do protótipo de um aplicativo móvel para o agendamento de serviços estéticos automotivos, ou seja, lavagem de carros através do levantamento bibliográfico consultando livros, coleta de dados e pesquisa em documentos eletrônicos. Tema de suma importância diante da evolução das tecnologias e popularização dos dispositivos e aplicativos móveis.

Com a escolha da linguagem de programação e ferramentas escolhidas, deu-se início ao desenvolvimento do aplicativo. Por se tratar de um primeiro contato com o *React Native* e não o conhecer, obtive bastante dificuldades no início, principalmente com o CSS, pois não entendia como era feita a renderização de seus componentes através do *StyleSheet*.

Após quebrar essa barreira, partir para criação dos componentes aqui citados por diversas vezes. Entendi que no início seria muito trabalhoso, porém com o crescimento do projeto, o mesmo ficaria mais simples de refatorar, pois as chamadas em várias telas eram constantes. Sendo assim, segui as boas práticas de programação no projeto.

O versionamento foi feito no GitHub, e a cada módulo que era disponibilizado, após ser testado, era enviado para o servidor do Git, evitando assim a perda de dados.

Todavia, mesmo após a pesquisa minuciosa sobre esses componentes que até o presente momento se mostram ideias para serem utilizados no trabalho, pode haver alguma mudança.

Outro ponto de grande desafio foi a autenticação de usuário. Aprender como o *React Native* trabalha com contextos e fazer esse encapsulamento. Para isso criou-se duas classes separadas. Uma denominada *AuthContextProvider*, onde toda a lógica de validação e autenticação, dados de sessão está inserida, e outra de rotas chamado *AppRoutes*, que é responsável por distribuir em todas as telas do aplicativo os dados obtidos, fazendo com o que não se perca durante a navegação. Como ficou

separado em classes diferentes e os diversos componentes para melhor experiência de usuário, a manutenibilidade é mais simples, pois estão separados e comentados cada função.

Com isso, a partir do referencial teórico, é possível inferir que o presente trabalho conseguiu responder seu objetivo proposto de agendamentos.

5.1 Trabalhos Futuros

Ao finalizar o desenvolvimento do aplicativo encontram-se duas oportunidades de melhorias. A primeira seria uma opção de listar os fornecedores próximos, sendo assim necessário a utilização de geolocalização. A segunda, é poder listar os horários disponíveis de cada fornecedor, pois está sendo possível agendar mais de um agendamento para o mesmo horário. Ao realizar essa alteração evitaria esse problema aqui descrito.

REFERÊNCIAS

ARTY, Deivid. Chief of Design. **Teoria das Cores - Guia sobre teoria e harmonia das cores no Design**. Disponível em:

<https://chiefofdesign.com.br/teoria-das-cores/?utm_source=rss>. Acesso em: 20 out. 2022.

BANKS, A.; PORCELLO, E. **Learning React: Functional Web Development With React and Redux**. O'Reilly Media. 2017, 350 p.

BERNARDO, Fernando. **VS Code: guia completo para instalar, usar e customizar**.

Disponível em: <<https://blog.betrybe.com/ferramentas/vs-code-guia-completo/>>. Acesso em: 01 set. 2022.

BIANCHI, G.; DAVID, P. L. D.; SUETA, R. U. **Luz e cor nas unidades de hemodiálise: Estudo de caso da Santa Casa de Misericórdia de Araçatuba**. Revista Nacional De Gerenciamento De Cidades, 5(31), 2017.

BORGES, Admir. **Marketing Digital: Um guia para iniciados e iniciantes**. Clube de Autores. 2019. 160 p.

CARVALHO, Henrique. **A Psicologia das Cores no Marketing e no Dia-a-Dia**.

Disponível em: <<https://viverdeblog.com/psicologia-das-cores/>>. Acesso em: 20 out. 2022.

CONTENT, Rock. **Conheça FireBase: a ferramenta de desenvolvimento e análise de aplicativos**. Disponível em: <<https://rockcontent.com/br/blog/firebase/>> Acesso em: 03 set. 2022.

CRUZ, R. **React Native + Expo**. Disponível em: <<https://rogercruz.medium.com/reactnative-com-expo-um-exemplo-40e5574c6904>>. Acesso em: 21 de nov. 2022.

DUARTE, L. **Introdução ao React Native**. Disponível em: <<https://www.luiztools.com.br/post/introducao-ao-react-native/>>. Acesso em : 23 out. 2022.

EISENMAN, Bonnie. **Learning React Native: Building Native Mobile Apps with JavaScript**. 2ª ed. O'Reilly Media. 2017, 272 p.

FIREBASE, A. **Firestore Authentication**. Disponível em: <<https://firebase.google.com/docs/auth>>. Acesso em: 31 out. 2022.

FLANAGAN, David. **JavaScript: O Guia Definitivo**. 6ª ed. Bookman, 2012. 1080 p.

GUEDES, Marylene. **O que é OAuth 2**. Disponível em : <<https://www.treinaweb.com.br/blog/o-que-e-oauth-2#:~:text=OAuth%20%20%C3%A9%20um%20protocolo,n%C3%A3o%20o%20acesso%20%C3%A0%20aplica%C3%A7%C3%A3o.>> Acesso em: 29 de nov. 2022.

GIL, Antonio Carlos. **Como Elaborar Projetos de Pesquisa**. 6ª. ed. São Paulo: Atlas, 2019.

HANSSON, N.; VIDHALL, T. **Effects on performance and usability for cross-platform application development using react native**. Disponível em : <<http://www.diva-portal.org/smash/get/diva2:946127/FULLTEXT01.pdf>>. Acesso em: 15 nov. 2022.

HORTON, A.; VICE, R.. **Mastering React**. Packt Publishing, ed. [S. l.: s. n.], 2016. 254 p.

KOTH, D. **A influência da iluminação e das cores no ambiente hospitalar: a saúde vista com outros olhos**. Especialize, 2013.

LECHETA, Ricardo R. **Google Android 5ª edição**: Aprenda a criar aplicações para dispositivos móveis com o Android SDK. 5ª. ed. São Paulo: Novatec Editora, v. 1, f. 536, 2016. 1072 p.

MANAIA, M. B. **A influência da iluminação no comportamento humano**. IJEP Instituto Junguiano, 2013.

MAWSTON, Neil. **Strategy Analytics: Half the World Owns a Smartphone**. Disponível em: <<https://news.strategyanalytics.com/press-releases/press-release-details/2021/Strategy-Analytics-Half-the-World-Owns-a-Smartphone/default.aspx>>. Acesso em: 28 set. 2022.

MELO, Diogo. **O que é JavaScript? [Guia para iniciantes]**. Disponível em: <<https://tecnoblog.net/responde/o-que-e-javascript-guia-para-iniciantes/>>. Acesso em: 19 mai. 2022.

MILANI, André. **MySQL - Guia do Programador**. Novatec Editora. 2007, 400 p.

MOZILLA e colaboradores individuais. **MDN web docs**. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/What_is_JavaScript>. Acesso em: 10 ago. 2022.

NODEJS, G. **About Node.js**. Disponível em: <<https://nodejs.org/en/about/>>. Acesso em: 21 de nov. 2022.

PRUSTY, Narayan. **Modern JavaScript Applications**. Packt Publishing, ed. [S. l.: s. n.], 2016. 330 p.

SANTOS, Antonio Raimundo Dos. **Metodologia Científica: A CONSTRUÇÃO DO CONHECIMENTO**. 7ª. ed. Lamparina, f. 80, 2018. 160 p.

SILVA, Eduardo. **FireBase: O que é e quando usar no desenvolvimento mobile?** Disponível em: <<https://blog.geekhunter.com.br/firebase-o-que-e-e-quando-usar-no-desenvolvimento-mobile/>>. Acesso em: 31 out. 2022.

SOUZA, Ivan. de. **Saiba o que é Node.js, como ele funciona e como usá-lo no seu site**. Disponível em: <<https://rockcontent.com/br/blog/node-js/>>. Acesso em: 15 de nov. 2022.

SHUTE, Zachary. **Advanced JavaScript: Speed up web development with the powerful features and benefits of JavaScript**. Packt Publishing, ed. [S. l.: s. n.], 2019. 330 p.

TECMUNDO. **MySQL: o que é e como usar o sistema?** Disponível em: <<https://www.tecmundo.com.br/software/223924-mysql-usar-o-sistema.htm>> . Acesso em: 30 set. 2022.

VALNET, Christian. **Chromotherapy - The power of colors**. Edizioni, [S. l.: s. n.], 2019. 154 p.

WAZLAWICK, Raul. **Metodologia de pesquisa para ciência da computação**. 2ª ed. [S. l.: s. n.], 2020. 152 p.