



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO  
AMAPÁ – IFAP

CURSO SUPERIOR DE TECNOLOGIA EM REDES DE  
COMPUTADORES

DANIELA FERREIRA SALES

**DESENVOLVIMENTO DE APLICATIVOS MÓVEIS:** O mercado das aplicações nativas  
e híbridas

MACAPÁ – AP

2023

DANIELA FERREIRA SALES

**DESENVOLVIMENTO DE APLICATIVOS MÓVEIS: O mercado das aplicações nativas  
e híbridas**

Trabalho de Conclusão de Curso apresentado ao curso Superior de Tecnologia em Redes de Computadores, do Instituto Federal de Educação, Ciência e Tecnologia do Amapá – Ifap, como requisito avaliativo para obtenção de título de Tecnólogo em Redes de Computadores.

Orientador: Prof. Me. Allan Meira de Medeiros

MACAPÁ - AP

2023

**Biblioteca Institucional - IFAP**  
**Dados Internacionais de Catalogação na Publicação (CIP)**

---

S163d Sales, Daniela Ferreira  
Desenvolvimento de Aplicativos Móveis: O mercado das aplicações nativas e híbridas / Daniela Ferreira Sales - Macapá, 2023.  
63 f.: il.

Trabalho de Conclusão de Curso (Graduação) -- Instituto Federal de Educação, Ciência e Tecnologia do Amapá, Campus Macapá, Curso de Tecnologia em Redes de Computadores, 2023.

Orientador: Allan Meira de Medeiros.

1. Mobile apps. 2. Mobile development. 3. Apps. I. Medeiros, Allan Meira de, orient. II. Título.

DANIELA FERREIRA SALES

**DESENVOLVIMENTO DE APLICATIVOS MÓVEIS: O mercado das aplicações nativas  
e híbridas**

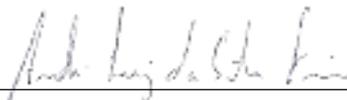
Trabalho de Conclusão de Curso apresentado ao curso Superior de Tecnologia em Redes de Computadores, do Instituto Federal de Educação, Ciência e Tecnologia do Amapá – Ifap, como requisito avaliativo para obtenção de título de Tecnólogo em Redes de Computadores.

Orientador: Prof. Me. Allan Meira de Medeiros

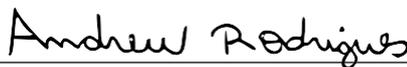
BANCA EXAMINADORA



Prof. Me. Allan Meira de Medeiros



Prof. Me. Professor André Luiz da Silva Freire



Prof. Me. Andrew Hemerson Galeno Rodrigues

Apresentado em: 12 / 06 / 2023.

Conceito/Nota: 9,6

## **AGRADECIMENTOS**

A todas as mulheres antecessoras que se propuseram a lutar por mudanças, das quais pude colher os frutos através da oportunidade de estar onde estou, em especial minha sogra Sônia Maria Vieira Forselini que me incentivou a voltar estudar.

A todos os profissionais da educação, que mesmo em cenários tão áridos continuam semeando a grande riqueza, o conhecimento, destaco meu orientador, Professor Allan Medeiros por sua dedicação e ao corpo docente do Instituto Federal do Amapá por todo acompanhamento no decorrer desta graduação.

Ao meu companheiro de vida, Rafael Vieira Forselini, por além de me proporcionar tempo e oportunidade, também transbordou sua generosidade, orientação e a tão necessária paciência ao longo dessa jornada.

A minha força interior, que mesmo em momentos extremos me recorda de ser: A Resistência!

“O que vale é o desafio.”

FORSELINI, 2005.

## RESUMO

O desenvolvimento de aplicativos móveis é, no momento presente dividido em aplicações nativas, cuja sua construção utiliza uma linguagem específica para um determinado sistema operacional, como por exemplo, Android e iOS, ou em aplicações híbridas, que funcionam através de frameworks na criação de aplicativos multiplataformas, isto é, poderão ser usados em mais de um sistema operacional. O crescimento exponencial da utilização de celulares na última década impulsionou uma indústria de aparelhos cada vez mais rápidos e com diversas tecnologias, desta forma o objetivo geral deste trabalho é pesquisar as alterações ocorridas nos últimos anos, após o surgimento da pandemia, no mercado de desenvolvimento móvel em relação a divisão pré-estabelecida entre as aplicações híbridas e nativas, e em seus objetivos específicos buscar investigar sobre as aplicações mais utilizadas, empregabilidade e remuneração, além de preferências dos profissionais desenvolvedores. É utilizado como principal método a pesquisa documental e com relação aos objetivos a pesquisa exploratória. Os resultados serão apresentados através da coleta de dados e análise das impressões observadas, em conclusão com as recomendações de possíveis contribuições futuras para a comunidade científica juntamente com as considerações finais em que se observa as alterações no mercado comprovado pelo aumento do tempo de uso e downloads nos apps gerando novas demandas, assim como a migração de websites para aplicativos gerando a discussão de como realizar suas construções que são avaliadas e decididas ainda de forma peculiar. Ademais o acompanhamento evolutivo da linguagem nativa Kotlin, o número de ofertas consideráveis ainda presente no mercado brasileiro da linguagem Java e a consolidação da linguagem Swift substituindo o Objective-C, outrossim a aplicação híbrida Flutter sendo considerada favorita pelos desenvolvedores e a diferença das remunerações no cenário internacional entre os desenvolvedores iOS e Android.

Palavras-chave: aplicativos móveis; desenvolvimento mobile; *Apps*; aplicações nativas; aplicações híbridas.

## ABSTRACT

The development of mobile applications is currently divided into native applications, whose construction uses a specific language for a given operating system, such as Android and iOS, or hybrid applications, which work through frameworks in the creation of applications. cross-platform, that is, they can be used on more than one operating system. The exponential growth in the use of cell phones in the last decade has driven an industry of increasingly faster devices and with different technologies, so the general objective of this work is to research the changes that have occurred in recent years, after the emergence of the pandemic, in the development market. mobile in relation to the pre-established division between hybrid and native applications, and in its specific objectives to seek to investigate the most used applications, employability and remuneration, in addition to preferences of professional developers. Documentary research is used as the main method and exploratory research is used in relation to the objectives. The results will be presented through data collection and analysis of observed impressions, in conclusion with recommendations for possible future contributions to the scientific community together with final considerations in which changes in the market are observed, proven by the increase in usage time and downloads. in apps, generating new demands, as well as the migration of websites to applications, generating discussion on how to carry out their constructions, which are still evaluated and decided on in a peculiar way. In addition to the evolutionary monitoring of the native language Kotlin, the considerable number of offers still present in the Brazilian market for the Java language and the consolidation of the Swift language replacing Objective-C, the hybrid application Flutter being considered a favorite by developers and the difference in remuneration in the international scene among iOS and Android developers.

Keywords: mobile apps; mobile development; *Apps*; native applications; hybrid applications.

## LISTA DE FIGURAS

Figura 1 - Crescimento dos Dispositivos.....	13
Figura 2 - Figura 2 – Camadas do iOS .....	19
Figura 3 – Anatomia Fisiológica de um Android .....	21
Figura 4 – Diagrama de uma chamada de uma API Nativa .....	23
Figura 5 – SDK Apple .....	24
Figura 6 – Xcode .....	24
Figura 7 – Swift Playground Online.....	26
Figura 8 – Replit.....	26
Figura 9 – Edições da Plataforma Java 2.....	28
Figura 10 – Plataforma Java 2 Micro Edition .....	30
Figura 11 – Java SE 8 Downloads (JDK).....	31
Figura 12 – Android Studio .....	31
Figura 13 – IntelliJ IDEA .....	32
Figura 14 – Página de Download Flutter.....	34
Figura 15 – Esquema de Produção de Dados Data.ai.....	39
Figura 16 – Capa do The State of Mobile 2022 – Brazil.....	40
Figura 17 – State of the Octoverse GitHub .....	48
Figura 18 – Linguagem de Maior Crescimento do State of the Octoverse .....	49
Figura 19 – Maior Crescimento Open Source por Contribuidores.....	49
Figura 20 – Maior Crescimento Open Source por Projeto .....	50
Figura 21 – Quadro de Divisão de Equipes por Aplicações.....	51
Figura 22 – Resultado de Desempenho das Aplicações .....	52

## LISTA DE TABELAS

Tabela 1 – Linguagens mais populares.....	42
Tabela 2 – Linguagens amadas, odiadas e querem aprendem.....	42
Tabela 3 – Remuneração média anual em dólar (US\$) .....	43
Tabela 4 – Vagas para desenvolvimento Android e iOS.....	44
Tabela 5 – Vagas para desenvolvimento Android e iOS.....	44
Tabela 6 – Vagas para aplicações mobile.....	45
Tabela 7 – Vagas para aplicações mobile.....	46
Tabela 8 – Média salarial.....	46
Tabela 9 – Média salarial.....	46
Tabela 10 – Desenvolvedores mobiles cadastrados no LinkedIn.....	47

## LISTA DE ABREVEATURAS E SIGLAS

ABES	Associação Brasileira das Empresas de Software
API	Application Programming Interface
APPS	Application
BRASSCOM	Associação das Empresas de Tecnologia da Informação e Comunicação
DVM	Máquina Virtual Dalvik
HTML	HyperText Markup Language
IA	Inteligência artificial
IDE	Integrated Development Environment
IOS	iPhone Operation System
JCP	Java Community Process
JDBC	Java DataBase Connectivity
JRE	Java Runtime Environment
JSDK	Java Software Development Kit
JSP	Java Server Pages
JSR	Java Specification Request
JVM	Máquina Virtual Java
NUDS	Nubank Design System
OHA	Open Handset Alliance
PWA	Progressive Web App
SDK	Software Development Kit
SO	Sistema Operacional
TIC	Tecnologia da Informação e Comunicação
WSDP	Web Services Developers Pack

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>13</b>
1.1	Justificativa .....	15
1.2	Objetivos.....	16
<b>2</b>	<b>REFERENCIAL TEÓRICO</b> .....	<b>17</b>
2.1	Sistemas Operacionais e Plataformas .....	17
2.2	iOS .....	18
2.3	Android .....	19
2.4	Aplicações Nativas.....	21
2.5	Objective-C .....	23
2.6	Swift.....	24
2.7	Java.....	26
2.8	Kotlin .....	31
2.9	Aplicações Híbridas.....	32
2.10	Flutter .....	33
2.11	React Native .....	34
2.12	Aplicações Web .....	35
<b>3</b>	<b>METODOLOGIA</b> .....	<b>37</b>
3.1	Coleta de Dados .....	38
3.1.1	GitHub.....	38
3.1.2	Stack Overflow.....	38
3.1.3	Data.ai .....	39
3.1.4	LinkedIn .....	40
3.1.5	Glassdoor.....	40
<b>4</b>	<b>RESULTADOS</b> .....	<b>42</b>
4.1	Aplicações .....	42
4.2	Empregabilidade .....	43
4.3	Profissionais .....	46
4.4	Mercado .....	47
4.4.1	Experiência Nubank .....	50
4.4.2	Experiência Airbnb.....	53
<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b> .....	<b>55</b>

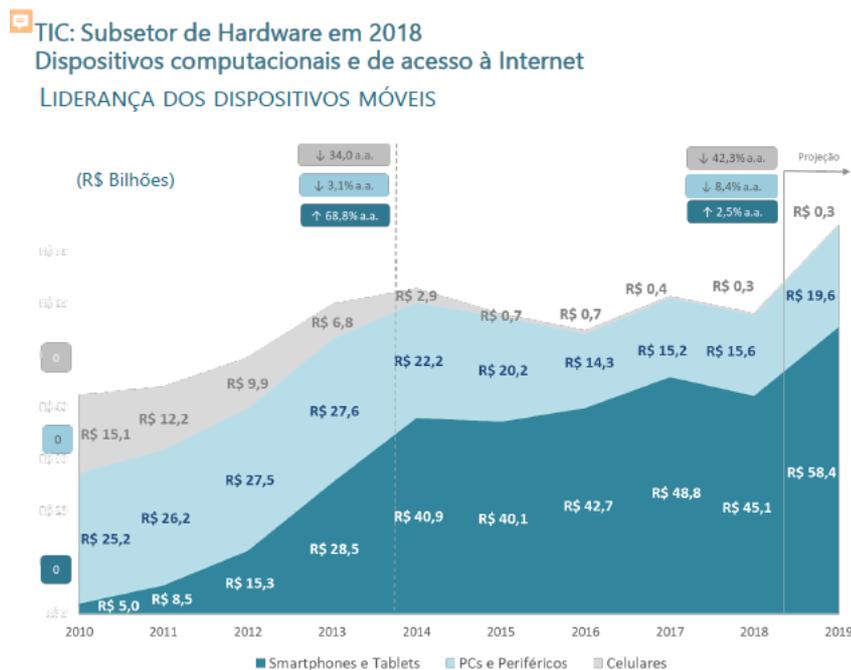
<b>REFERÊNCIAS .....</b>	<b>56</b>
--------------------------	-----------

## 1 INTRODUÇÃO

No final do ano de 2019, com o surgimento do vírus SARS-CoV-2, popularmente conhecido como COVID-19, instalou-se uma pandemia mundial. O distanciamento social foi uma das principais formas encontradas ao combate contra sua contaminação, ou seja, com o contato físico precisando ser evitado outras formas de comunicação tornaram-se mais populares para a permanência de atividades e rotinas essenciais. Conseqüentemente um grande salto das conexões remotas, milhões de usuários conectados a seus aparelhos eletrônicos e o aumento do volume de dispositivos móveis, justificado por sua acessibilidade e oferta de modelos de baixo custo quando comparados a computadores desktops e notebooks.

Mesmo com o *boom* de crescimento impulsionado pela pandemia, a ascensão da utilização de celulares antecede a esse período, pois em âmbito nacional, segundo a BRASSCOM – Associação das Empresas de Tecnologia da Informação e Comunicação (TIC) e de Tecnologias Digitais, em seu relatório setorial de 2018, divulgava a análise do crescimento de 31,5% entre os anos de 2010 a 2018 dos dispositivos de smartphones e tablets, comparando além disso que, em 2018 a venda de smartphones correspondeu a 96,1% entre os dois dispositivos, observados na Figura 1.

Figura 1 - Crescimento dos Dispositivos



Fonte: (BRASSCOM, 2019).

Em meio a pandemia, um dos primeiro impactos observados segundo a Data.ai Analytics, companhia que analisa métricas de aplicativos nos *marketplaces* da Apple e Google, foi a mudança de comportamento através do tempo gasto na utilização dos aplicativos (*apps*), números expressivos foram mostrados no relatório *The State of Mobile 2019*, que colocava o Brasil em 5º posição no ranking dos países em que o usuário médio gastava cerca de 3 horas e 40 minutos por dia em um app, já em seu relatório do ano de 2022, *The State of Mobile 2022*, o crescimento teve um salto para quase 6 horas de tempo de uso em um aplicativo. Além da alta performance dos brasileiros em tempo gasto demonstrados pela Data.ai, outro fator importante pelo qual o Brasil teve destaque foi o número de downloads, se posicionando na 4º posição atrás apenas respectivamente dos Estados Unidos, Índia e China. A adoção de dispositivos móveis experimentou um *boom* em 2021, com crescimento em downloads, uso e gastos de consumidores nas lojas de aplicativos: mais de 232 bilhões de downloads, US\$ 168 bilhões em gastos de consumidores e 3,9 trilhões de horas globalmente (DATA.AI, 2021).

O desenvolvimento de softwares aplicativos móveis acompanha a indústria de celulares que, a partir de 2008, um ano após a criação do iPhone pela famosa Apple, o iOS (sistema operacional móvel da Apple Inc.) ganhou espaço e se consolidou, acompanhando esse lançamento, a concorrente Google lançou a Android Market (atualmente Google Play Store), oferecendo ao mercado um sistema operacional móvel a outras marcas de aparelhos (Samsung, Motorola, LG, Oppo, Huawei, Xiaomi etc.). A criação de aplicativos que operem nos sistemas Android e iOS é complexa, pois assim como os aparelhos celulares que se atualizam a todo instante, os métodos de desenvolver um software aplicativo (*app*) estão sendo alterados, é constante o surgimento de novos caminhos e ideias, isto é, aplicações, linguagens e frameworks, que começam a ser apresentadas a desenvolvedores e empresas por darem maior performance dentre outras vantagens que necessitam ser investigadas. Desta forma, como objetivo geral a pesquisa buscou apresentar as alterações ocorridas nos últimos anos no mercado de desenvolvimento mobile em relação a divisão pré-estabelecida entre as aplicações híbridas e nativas, em seus objetivos específicos buscou investigar sobre as aplicações mais utilizadas, empregabilidade e remuneração, além da oferta e preferência por linguagens de profissionais desenvolvedores, apresentada nos capítulos 4 e 6.

No capítulo 5 consta a metodologia deste trabalho que, utilizou como principal método a pesquisa documental, cuja natureza tratou-se de uma pesquisa aplicada e as abordagens metodológicas de uma pesquisa quantitativa, devido a se preocupar com a representatividade numérica, já no tocante aos objetivos, o trabalho seguiu através da linha da pesquisa exploratória. Outrossim, foi apresentado os sites e organizações dos quais alimentaram a coleta de dados da pesquisa.

Os resultados dos objetivos deste trabalho foram apresentados no capítulo 6, mediante da apresentação da coleta de dados e análise dos resultados, e no capítulo 7 tangeu as considerações finais sobre a problemática desta pesquisa, suas limitações e recomendações para uma possível continuidade futura.

## 1.1 Justificativa

Segundo a ABES – Associação Brasileira das Empresas de Software, o mercado mundial de Software e Serviços atingiu em 2018 o valor de US\$ 1,220 bilhões, e o Brasil se manteve na 9ª posição no ranking mundial, com um mercado interno de aproximadamente US\$ 23 bilhões. Em seu relatório *Mercado Brasileiro de Software 2019 – Panorama e Tendências*, ao segmentar o desenvolvimento de software aplicativos os resultados mostraram um crescimento de 16,7% comparados ao ano de 2017. Já em seu último relatório *Mercado Brasileiro de Software – Panorama e Tendências* publicadas em 2021, o mesmo segmento de softwares de aplicativos mostra o crescimento de 20,9% comparado ao ano de 2020. Este segmento avançava antes mesmo da pandemia, mas após o surgimento desse *novo normal* quais foram seus impactos e novas demandas?

Sob essa expansão, os impactos da pandemia alteram comportamentos sociais, globais e mercadológicos, o desenvolvimento de software de aplicativos caminha aceleradamente e necessita de mais visibilidade a respeito dos mecanismos que estão por de trás das telas na construção de *apps* atingindo milhões de usuários. Como estão sendo construídos? Em quais tecnologias os desenvolvedores e empresas estão baseando suas escolhas no momento de planejar e implementar um projeto?

No tempo presente, as formas mais utilizadas na produção do desenvolvimento mobile estão divididas entre o desenvolvimento nativo, do qual utiliza-se uma linguagem específica para um determinado sistema operacional, como por exemplo, Android ou iOS, e o desenvolvimento híbrido, através de uma frameworks que criam aplicativos multiplataformas, ou seja, poderão funcionar para ambos sistemas. Tendo em vista como foco da pesquisa investigar as escolhas que mais populares e tendências mercadológicas na criação dos *apps*, é pretendido investigar as formas de desenvolvimento nativo e híbrido mais utilizadas no momento atual, suas características e predisposições de profissionais desenvolvedores e empresas contratantes no mercado brasileiro.

## **1.2 Objetivos**

### **Geral**

Pesquisar as mudanças no mercado de desenvolvimento de mobile em relação a escolha das aplicações nativas e híbridas.

### **Específicos**

- Pesquisar as aplicações mais utilizadas no mercado mobile atualmente.
- Apontar quais aplicações oferecem maior número de vagas de emprego e remuneração.
- Analisar quais aplicações mais os profissionais desenvolvedores utilizam e tem interesse em aprender.

## 2 REFERENCIAL TEÓRICO

Para melhor compreensão do tema abordado, o capítulo será dedicado a revisão de alguns conceitos referentes ao desenvolvimento mobile como: Sistemas Operacionais e Plataformas, iOS, Android, Aplicações Nativas, Objective-C, Swift, Java, Kotlin, Aplicações Híbridas, Flutter, React Native e Aplicações Web.

### 2.1 Sistemas Operacionais e Plataformas

Na última década, devido a perda de popularidade, encerramento de atividades e desuso dos sistemas para Blackberry, Windows Phone e outros, este trabalho terá como foco exclusivo as tecnologias voltadas para Android e iOS por atualmente corresponderem a mais de 99% do mercado mobile (GARTNER, 2018).

Um sistema operacional móvel gerencia smartphones, tablets e outros dispositivos usados em movimento ou da interação desses com outros aparelhos fixos como TVs, ar-condicionado, geladeiras, etc. Além do design físico do próprio aparelho, o sistema operacional móvel é responsável pela forma como os usuários interagem com o dispositivo. Essa interface do usuário inclui a aparência, o design geral e a funcionalidade, sendo cada um desses elementos cruciais para garantir uma experiência de usuário positiva.

Uma das bases para esse processo é a utilização de SDK (Software Development Kit) que é um conjunto de ferramentas que permite o desenvolvimento de software para um determinado sistema operativo, consola, dispositivo móvel, ou qualquer outra plataforma similar. Segundo SANTOS (2013), um SDK, normalmente, tem um IDE (Integrated Development Environment) associado que dá suporte aos programadores com o objetivo de criar software com mais qualidade, pois inclui funcionalidades como debug e validação sintática e semântica no momento em que o desenvolvimento ocorre. Os SDK permitem a construção de componentes desde os mais simples aos mais complexos com mais agilidade, além de muitos deles oferecerem a integração com outros softwares, documentação, bibliotecas de códigos, editores visuais e guias, possibilitando o teste das aplicações desenvolvidas (PEDRO, 2022).

Outra ferramenta importante é a API (Application Programming Interface) que segundo Foldoc (2015), seria a interface pela qual um programa aplicativo acessa o sistema operacional e outros serviços. Uma API é definida no nível do código-fonte e fornece um nível de abstração entre o aplicativo e o kernel (ou outros utilitários privilegiados) para garantir a portabilidade do código. Através das APIs é possível o fornecimento de uma interface entre uma linguagem de alto nível e utilitários e serviços de nível inferior, sendo também usadas como tradutoras de

listas de parâmetros entre formatos e a interpretação de argumentos.

## 2.2 iOS

Junto ao lançamento do primeiro celular iPhone em 2007, o sistema iOS (*iPhone Operation System*) surgiu da necessidade de um ambiente móvel, foi criado pela Apple Inc. baseado no sistema operacional MAC OS X e idealizado para atender as necessidades desse novo tipo de produto. É através do sistema iOS que ocorre a abstração entre a comunicação do hardware com o aplicativo (APPLE, 2008).

A arquitetura do iOS possui 4 camadas, que segundo Yates (2010) são:

- **CocoaTouch:** Esta camada fornece ferramentas e infraestrutura para implementar eventos e aplicações para a interface do iPhone. Fornece uma gama de frameworks utilizados para a construção dos aplicativos como por exemplo: frameworks para aparência, multi tasking, touch-based input, push notifications, dentre outros (DEVMEDIA, 2020).
- **Media:** Camada responsável por fornecer recursos áudio e vídeos. A tecnologia mais avançada para experiência multimídia e são encontradas bibliotecas como: OpenGL ES, QuartzCore etc. De acordo com a filosofia da Apple, o desenvolvedor deve se preocupar em oferecer uma boa experiência aos seus usuários, portanto é umas das principais camadas (DEVMEDIA, 2020).
- **Core Services:** A camada que fornece os serviços fundamentais do sistema como AdressBook, Core Location, CFNetwork, Security, SQLite etc. Tendo como seus principais frameworks o Core Foundation e o Foundation.
- **Core OS:** Nesta camada que se encontra o kernel do sistema. Além do kernel, encontramos os drivers e as interfaces básicas do sistema. Sendo uma camada de baixo nível é muito utilizada pelas camadas de nível superior de forma indireta. Sua utilização ocorre além disso em negociações de segurança ou comunicação com hardwares externos, como por exemplo, comunicação via bluetooth (DEVMEDIA, 2020).

Essas camadas podem ser visualizadas na Figura 2.

Figura 2 - Figura 2 – Camadas do iOS



Fonte: (YATES, 2010)

Segundo Magalhães (2018), para um aplicativo iOS oferecer interações ricas e responsivas aos seus usuários a Apple disponibiliza para seus desenvolvedores Frameworks que tornaram possível o desenvolvimento, mas com anos, o uso cada vez mais de frameworks como UIKit, Foundation e CoreData se tornaram mais comuns, o que gerou grandes mudanças e, ao ouvir a comunidade, a partir de 2014, o que antes era apenas possível ser criado na linguagem nativa Objective-C, a Apple apresentou a linguagem Swift para o desenvolvimento iOS entregando algumas melhorias como uma tipagem mais forte, adição de extensões, métodos mais curtos e seguros com menor complexidade, o que impactou profundamente o modo de se desenvolver iOS comparado ao que vinha sendo realizado em Objective-C (HABCH, 2018).

### 2.3 Android

Desenvolvido por Andy Rubin, Rich Miner, Nick Sears e Chris White em 2003 e posteriormente adquirida pelo Google em 2005, o Android é um sistema operacional móvel *open source*, possui uma arquitetura baseada na versão 2.6 do kernel Linux para o controle das principais tarefas do sistema como segurança, gerenciamento de memória, gerenciamento de processos, pilha de rede e modelo de driver (GOOGLE INC, 2011).

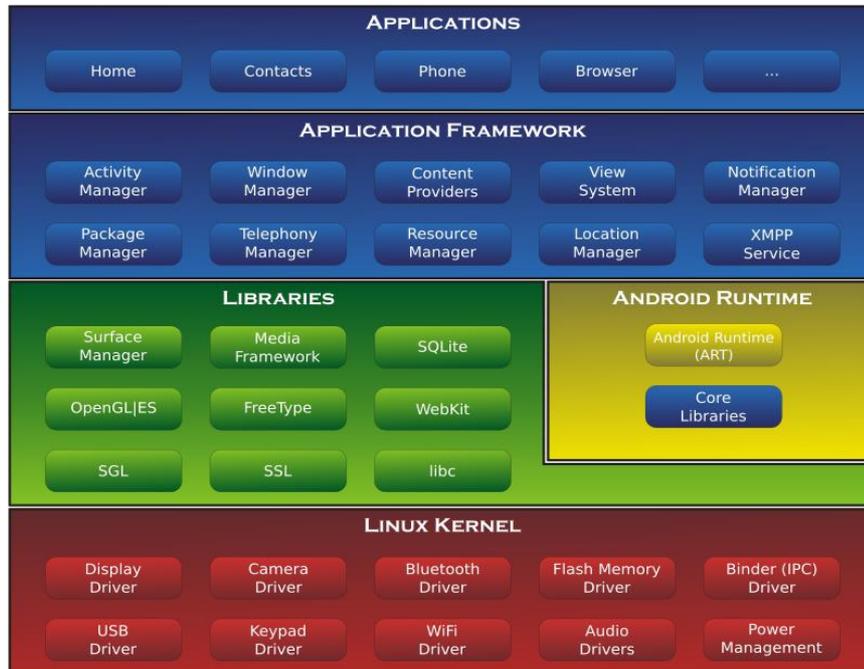
Em 2007, o desenvolvimento do Android foi transferido para um grande consórcio de tecnologia, o Open Handset Alliance (OHA), com objetivo de popularizar e melhorar os dispositivos móveis, o Google tornou-se um dos membros do consórcio, continuando com a responsabilidade por controlar e gerenciar importantes etapas do desenvolvimento do sistema e engenharia de processos (DIAS et al, 2011), entre outros participantes do consórcio integraram os fabricantes de telefones como HTC, Sony e Samsung, operadoras de telefonia como Sprint Nextel e T-Mobile e fabricantes de chipsets como Qualcomm e a Texas Instruments (OpenHeadsetAlliance 2011). Segundo Dias et al (2011), a arquitetura do sistema

Android é composta por cinco camadas:

- **Applications:** é a primeira camada, mais perto do usuário e a dos aplicativos. Ela é composta pelos aplicativos nativos do Sistema Operacional Android, como cliente de e-mail, programa de SMS, calendário, mapas, navegador etc.
- **Applications Frameworks:** camada onde estão os componentes responsáveis pelo gerenciamento das Activitys, gerenciamento das Views, gerenciamento de Janelas, Provedores de Conteúdo e etc. Nesta camada ocorre a manipulação dos desenvolvedores durante o processo de desenvolvimento.
- **Android Runtime:** A camada responsável pela execução dos aplicativos. Os aplicativos são escritos na linguagem Java, quando compilados eles geram bytecodes Dalvik ao contrário de bytecodes JVM devido ao sistema não possuir Máquina Virtual Java (JVM). No Android está presente a Máquina Virtual Dalvik (DVM) desenvolvida para dispositivos móveis, ela é uma versão otimizada da JVM por possuir menos instruções.
- **Libraries:** camada onde localiza-se várias bibliotecas como a biblioteca C padrão, SQLite (Banco de Dados), OpenGL (Renderização 3D) etc.
- **Linux Kernel:** A última camada é a de mais baixo nível e do kernel. O kernel atua como uma camada de abstração entre hardware e as camadas superiores, permitindo acesso a recursos como áudio, vídeo e protocolos de rede. (DIAS et al, 2011)

Na Figura 3 a explicação visual do detalhamento referido.

Figura 3 – Anatomia Fisiológica de um Android



Fonte: (SMIEH, 2006).

Segundo MONTEIRO (2013), ainda no ano de 2007, ocorreu o lançamento da versão beta do primeiro SDK para Android, o que gerou o aumento de novos aplicativos devido a compatibilidade do sistema com diversos fabricantes e empresas. O Kit de Desenvolvimento de Software (SDK) do Android inclui um conjunto abrangente de ferramentas de desenvolvimento, como o IDE (Android Studio), emuladores, ferramentas de análise de performance, entre outras. No desenvolvimento de aplicações nativas Android utiliza-se Kotlin ou Java como linguagem de programação (ASP, 2018; HANSSON; VIDHALL, 2016).

## 2.4 Aplicações Nativas

Segundo REIS (2019), as aplicações nativas são focadas no desenvolvimento de uma aplicação para uma plataforma específica, seja IOS ou Android, por conseguinte, são executadas apenas em uma plataforma específica. Através desse tipo de desenvolvimento há a necessidade de que o código base seja alterado quando muda de plataforma, pois cada uma dessas terá sua própria linguagem de programação, portanto IDE divergente. Nas aplicações nativas é fundamental o uso de SDK, algumas funcionalidades são mencionadas por (KUMAR, 2014):

- Multi-touch: Gestos compostos de interface do usuário;
- API gráficas rápidas: gráficos extremamente rápidos;
- Animação fluida: Altamente interativos e uso intenso de algoritmos

computacionais;

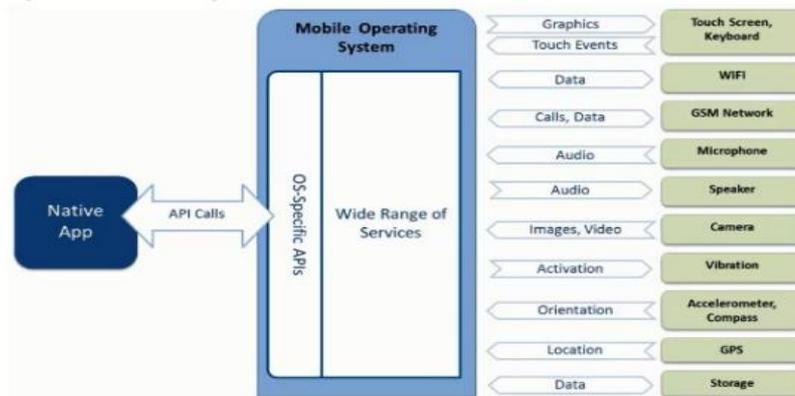
- Componentes embutidos (nativos): Câmera, geolocalização e lista de contatos;
- Interação: Capaz de integrar com outros aplicativos na tela inicial;
- Flexibilidade: Responde a comandos nativos como controle de volume;

Documentação: Uma quantidade muito alta de material disponível para IOS e Android.

Ainda segundo REIS (2019), um fator sobre desenvolvimento nativo que geralmente é levado em consideração é o desempenho, pois há um ambiente mais apropriado para a execução da aplicação, como guias do próprio sistema operacional para seguir e melhorar a experiência do usuário, outra vantagem é a possibilidade de se utilizar das capacidades integradas do próprio dispositivo devido não haver uma camada adicional para o acesso dessas funcionalidades nativas do dispositivo. De acordo com ABED (2016) um dos motivos para a melhoria do desempenho é sua própria estrutura de aplicação que é mais rápida e confiável, devido ao conteúdo, estrutura e componentes visuais estarem embutidos no próprio dispositivo, o que proporciona um carregamento imediato.

As aplicações nativas oferecem também a possibilidade do armazenamento de dados offline De acordo com VIJAYAJUMAR (2017), há diversas formas de lidar com capacidade offline e sincronização de dados em aplicações nativas, como a sincronização de dados inicial ser realizada assim que um novo dado é requisitado e guardado localmente, do mesmo modo que, a sincronização de dados locais para servidores, quando ocorre a necessidade que as aplicações sejam capazes de armazenar dados locais durante o uso offline e envie os dados previamente salvos quando a conexão com a internet estiver restabelecida. Tal sincronização acontece devido ao recebimento de um callback do próprio sistema operacional comunicando sua realização, na Figura 4 é possível visualizar o funcionamento de uma aplicação nativa em um dispositivo móvel.

Figura 4 – Diagrama de uma chamada de uma API Nativa



Fonte: (SANTOS, 2014)

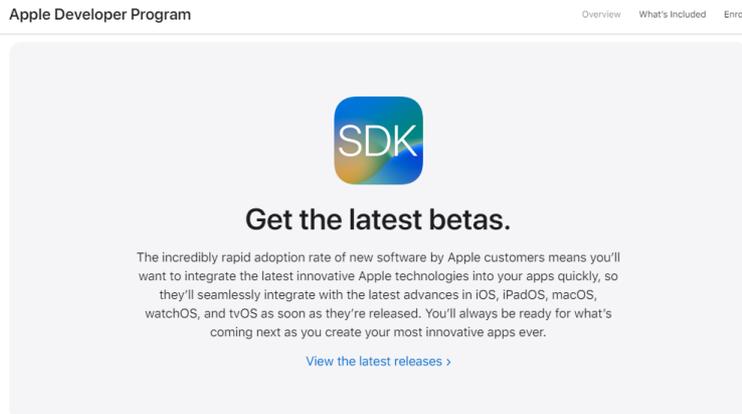
## 2.5 Objective-C

Objective-C foi é uma linguagem orientada a objetos, possui características da linguagem C na maior parte de sua sintaxe (incluindo pré-processamento, expressões, declaração e chamadas de funções), e algumas características de Smalltalk, como o uso de parâmetros dentro do nome do método em vez de em seção de parâmetros no mesmo.

Criado no início da década de 1980 por Brad Cox e Tom Love, foi posteriormente nos anos 90, após sua demissão da Apple, um dos investimentos realizados por Steve Jobs, que adquiriu parte da empresa NeXT e licenciou o Objective-C. Inicialmente, foi escolhida para ser utilizada no Mac OS X e GNUstep, ambientes inspirados no padrão OpenStep sendo a principal linguagem utilizada nos aplicativos estruturais NeXTSTEP, OPENSTEP e Cocoa (versão da Apple do ambiente de desenvolvimento NeXtstep com algumas melhorias). Uma das características dessa linguagem é que o Objective-C suporta polimorfismo, posing, categorias, e é uma linguagem dinâmica, com typing e binding dinâmicos (SILVA ET AL, 2012).

Para utilização do Objective-C são necessárias instalações do SDK (*Software Development Kit* - Kit de Desenvolvimento de Software) de desenvolvimento para iOS, disponível gratuitamente no site da Apple, visto na Figura 5, da mesma forma, do IDE (*Integrated Development Environment* - Ambiente de Desenvolvimento Integrado) *Xcode*, visualizado na Figura 6, por meios legais, não é possível programar para iOS sem usar um Mac com processador Intel, caso utilize outro tipo de sistema operacional terá que simular um Mac OS X (virtualização).

Figura 5 – SDK Apple



Fonte: (DEVMEDIA, 2018)

Figura 6 – Xcode



Fonte: (DEVMEDIA, 2018)

## 2.6 Swift

Nos meados de 2014 a Apple anunciou o lançamento de uma nova linguagem de programação nativa do OS X e do iOS, o Swift. Em conferência anual de desenvolvedores da Apple, o vice-presidente de engenharia de software Craig Federighi chamou a nova linguagem de *“rápida, moderna e projetada para segurança”, é como “Objective-C, sem o C”* (VentureBeat, 2014). Segundo Federighi, *“quando se trata de velocidade, o Swift é ótimo. O Swift também é moderno com recursos como encerramentos, vários tipos de retorno e namespaces. Swift é completamente nativa de Cocoa e Cocoa Touch”*.

O Swift foi projetado por Chris Lattner, que contou com apoio dos demais desenvolvedores da Apple, para desenvolver uma linguagem de programação baseada nos

melhores conceitos de linguagens como Objective-C e C, porém também adotando abordagens de linguagens como Rust, Haskell, Ruby, Python, C#, CLU, entre muitas outras (DEVMEDIA, 2018). O Objective-C era apenas uma adaptação da linguagem C. Isso causava uma linguagem que não era totalmente voltada a objetos, possuindo muitos valores escalares. Além disso, a sintaxe do Objective-C era complicada, e a correção de erros era mal executada pelos programas (MACEDO, 2017).

Segundo Macedo (2017), a linguagem teve seu aprimoramento apoiado em seis principais pontos:

- **Orientação ao objeto:** é uma linguagem puramente orientada ao objeto. Todos os elementos em Swift devem ser considerados objetos;
- **Clareza:** é uma linguagem de fácil leitura. Sua sintaxe é limpa, consistente e explícita;
- **Segurança:** o Swift reforça que o programador seja bem descritivo em sua linguagem, de maneira que haja menos ambiguidades ao explicitar diretamente o tipo de cada objeto reverenciado;
- **Economia:** apesar de bastante descritiva, o Swift é uma linguagem de poucos tipos e funcionalidades internas a ela. As demais funcionalidades devem ser providas diretamente pelo programa;
- **Gerenciamento de memória:** o Swift gerencia a memória automaticamente, não sendo necessário por parte do programador a necessidade de realizar o gerenciamento dentro do código da aplicação;
- **Compatibilidade com Cocoa:** o API Cocoa é escrito em C e Objective-C. Swift é desenvolvido para interagir com esse API (MACEDO, 2017).

Mesmo possuindo várias diferenças com o Objective-C, o Swift interage muito bem com essa linguagem, fazendo com que o desenvolvimento de um aplicativo possa ter ambas as linguagens sem que haja maiores problemas na transição entre elas.

Outra característica do Swift é ter o código aberto (*open source*), o que disponibiliza o compartilhamento ao público, sem restrições, de forma descentralizada e colaborativa, contando com revisões e produções pela comunidade Apple.

O open source é hoje um movimento tecnológico e uma forma de trabalho que vai além da produção de software. Esse movimento usa os valores e o modelo descentralizado de produção do software open source para descobrir maneiras inovadoras de resolver problemas em suas comunidades e setores (RED HAT, 2019).

Atualmente o Swift é utilizado para desenvolvimento de aplicativos iOS, MacOS, tvOS e watchOS, com resultados de crescimento exponencial no mercado, podendo oferecer também

soluções em back-end com o auxílio de algum framework, como o Vapor ou o Kitura (ALURA, 2020).

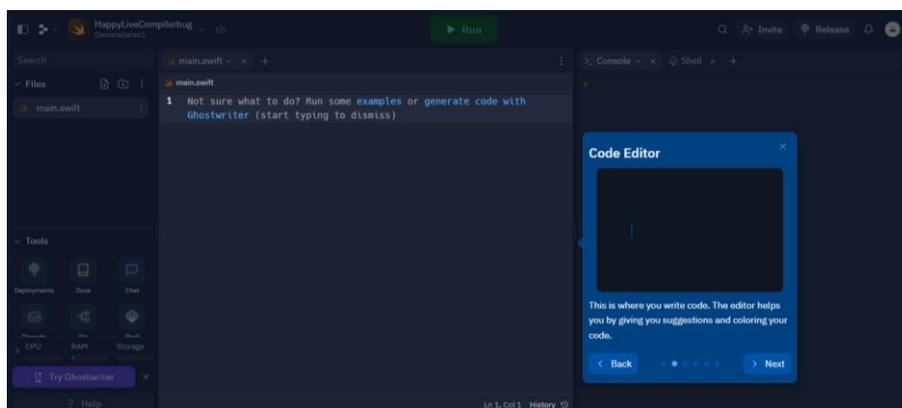
Assim como Objective-C, o Swift utiliza a do IDE (Integrated Development Environment - Ambiente de Desenvolvimento Integrado) Xcode, visualizado na Figura 6, que só está disponível no ambiente MacOS, mas é possível utilizar um compilador online, como é o caso do Swift Playground Online (que está disponível para MacOS também) ou até mesmo o Replit, visualizados respectivamente nas Figuras 7 e Figura 8 para criar aplicações iOS ou arquivos como forma de aprendizagem da linguagem Swift.

Figura 7 – Swift Playground Online



Fonte: (APPLE, 2017)

Figura 8 – Replit



Fonte: Autoria própria

## 2.7 Java

Segundo DEITEL (2017), após o surgimento dos microprocessadores e o impacto gerado nos aparelhos eletrodomésticos e eletrônicos de consumo, a Sun Microsystems em 1991,

observando esse mercado, resolveu financiar um projeto corporativo interno de desenvolvimento de codinome Green. Um dos resultados obtidos foi a criação de uma linguagem baseada em C++, que o criador, James Gosling, denominou Oak ('carvalho', em inglês, em homenagem a uma árvore que ele via através da janela de seu escritório na Sun), mas posteriormente esse nome foi alterado por já existir outra linguagem esse nome, foi então que ao visitarem a cafeteria local, a equipe sugeriu o nome Java.

Ainda na década de 90, de acordo com DEITEL (2017), o projeto Green passou por dificuldades, devido ao mercado de aparelhos eletrônicos de consumo inteligentes não se desenvolver rapidamente o que gerava o risco do cancelamento do projeto. Mas com o crescimento da World Wide Web explodiu em 1993, a Sun visualizou uma nova forma para poder aplicar a Java e criar páginas da Web com o chamado conteúdo dinâmico (por exemplo, interatividade, animações e coisas desse gênero).

A semelhança era que, na internet, havia uma grande quantidade de sistemas operacionais e browsers e, com isso, seria uma grande vantagem poder programar em uma única linguagem, independentemente da plataforma, surgindo assim o Java 1.0, focado em transformar o browser de apenas um cliente magro (thin client ou terminal burro) em uma aplicação que poderia também realizar operações avançadas, e não apenas renderizar HTML (ALURA, 2018). Java é uma linguagem compilada para bytecode que é interpretado por uma máquina virtual (Java Virtual Machine, abreviada JVM).

Formalmente, a Java foi anunciada em uma conferência em maio de 1995 e, além de ser utilizada na World Wide Web evoluiu em suas versões para poder desenvolver aplicativos empresariais de grande porte, aumentando a funcionalidade de servidores Web, oferecer aplicativos a aparelhos eletrônicos de consumo (tais como telefones celulares, tablets e TVs) e muitas outras finalidades. Em 2009, a Oracle comprou a Sun, fortalecendo a marca. A Oracle sempre foi, junto à IBM, uma das empresas que mais investiram e fizeram negócios por meio do uso da plataforma Java (ALURA, 2018).

Por se tornar uma linguagem amplamente utilizada, para compreender como a Java é utilizada no desenvolvimento de aplicativos móveis é necessário visualizar um pouco sobre a arquitetura a partir da sua segunda versão que, de acordo com PALUDO (2003), foi a partir dessa versão, lançada em 1998, onde houveram grandes modificações e reestruturação dos pacotes divididos em três tipos diferentes, sendo eles:

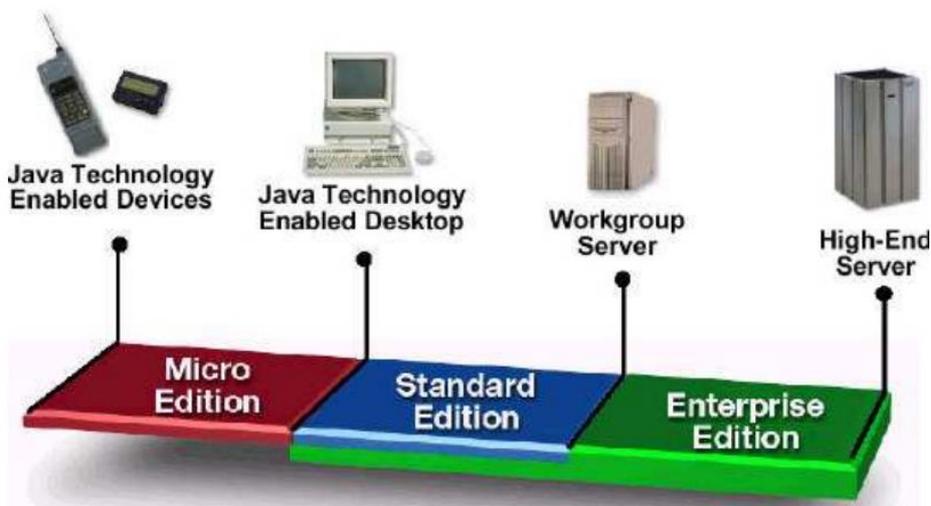
- J2SE (Java 2 Standard Edition): É a versão básica, destinada ao desenvolvimento da maior parte das aplicações de desktop.
- J2EE (Java 2 Enterprise Edition): Versão destinada ao desenvolvimento de aplicações de grande porte, as quais fazem extenso uso como exemplo de EJB

(Enterprise Java Beans) e Servidores de Aplicação. Outro grande personagem desse pacote são as servlets.

- J2ME (Java 2 Micro Edition): Destinada ao desenvolvimento de programas para periféricos móveis ou de pequeno porte, como celulares e relógios. (PALUDO, 2003).

Na figura 9 as edições da plataforma Java 2.

Figura 9 – Edições da Plataforma Java 2



Fonte: (PALUDO, 2003)

Entre as principais definições mais importantes do Java estão:

- JVM (Java Virtual Machine): É o mecanismo que interpreta os arquivos .class (pré-compilados), fazendo o programa funcionar em qualquer plataforma. A JVM pode ser considerada o coração do Java, ela tem este nome pois "simula" um computador. A JVM pode não apenas interpretar o código, como às vezes ela compila alguns trechos deste para acelerar a execução (just in time compiler).
- JSDK (Java Software Development Kit), ou simplesmente JDK, é o ambiente de desenvolvimento Java. Entre outras coisas, inclui: Java Virtual Machine, compilador, appletviewer para executar applets, códigos de exemplo e bibliotecas básicas do Java.
- JRE (Java Runtime Environment): Indicado para quem precisa apenas executar as aplicações Java, sendo que o pacote contém somente as bibliotecas necessárias para tal e a virtual machine (não se consegue compilar código Java apenas com isto).
- JCP (Java Community Process): É uma organização internacional de

desenvolvedores e fabricantes cuja missão é desenvolver e revisar as especificações, referências e outras implementações relacionadas ao Java. Para maiores informações, a página oficial é <http://www.jcp.org>.

- JSR (Java Specification Request): São as propostas de mudanças e novas implementações para o Java. Em outras palavras, são os "documentos" que definem tal funcionalidade, ou propõem mudanças em algo existente, por exemplo.
- WSDP (Web Services Developers Pack): Conjunto de APIs e ferramentas para o desenvolvimento de WebServices e servlets em Java.
- JSP (Java Server Pages): Tecnologia baseada em Java para o desenvolvimento de aplicações Web. É uma linguagem de script parecida com PHP, com a diferença de usar Java e ser traduzida para puro Java no momento da execução.
- JDBC (Java DataBase Connectivity): Especificação que deve ser seguida para oferecer suporte ao acesso a Banco de Dados em Java (PALUDO, 2003).

A partir da segunda versão, através da criação da plataforma Java Micro Edition ou J2ME como popularmente mais conhecida, que o desenvolvimento voltado para os dispositivos móveis se tornou possível, pois a tecnologia J2ME trouxe vantagens específicas para aparelhos de pequeno porte, aplicações integradas ao serviço das operadoras, juntamente com a interatividade entre aparelho e usuário, na Figura 10 é possível observar a estrutura do Java 2 Micro Edition.

J2ME é, afinal, um conjunto de especificações, que define uma JVM (Máquina Virtual Java) simplificada, um conjunto de APIs (Interfaces para Programação de Aplicações) especializadas para pequenos dispositivos e ferramentas direcionadas aos dispositivos com poder de processamento, memória e conectividade menor que um computador de mesa, (acima citados, telefones, handhelds, sistemas embutidos), com capacidade de execução local de aplicações (PALUDO. 2003).

Figura 10 – Plataforma Java 2 Micro Edition

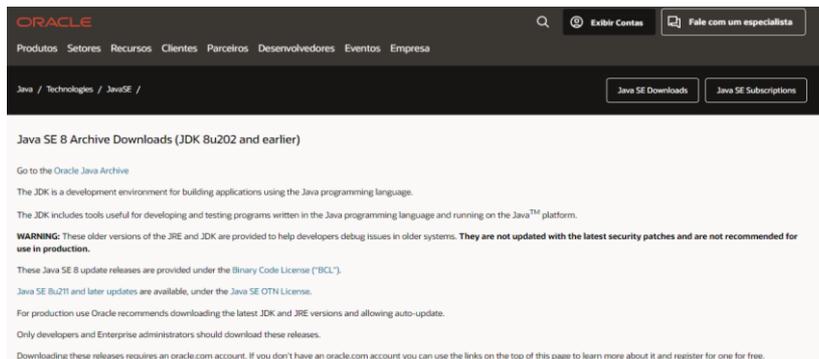


Fonte: (PALUDO, 2003)

Após décadas de seu lançamento, a linguagem Java se popularizou, por ser uma linguagem orientada a objetos, com o surgimento da internet apresentando dinamismo, como interatividade e animações, às páginas da web. Atualmente sabemos que a Java domina o mercado de aplicações para celulares com mais de 2.5 bilhões de dispositivos compatíveis, pelo simples fato de a Java ter se tornado a linguagem-chave no desenvolvimento de aplicativos Android adequados a smartphones e tablets. Sob a supervisão da Oracle desde 2010, continua sendo uma das linguagens mais usadas no desenvolvimento de software. Ela é executada em 3 bilhões de dispositivos em todo o mundo e é usada por mais de 12 milhões de desenvolvedores (ORACLE, 2020).

Para a utilização e aplicação da linguagem Java no desenvolvimento de aplicações móveis são necessários alguns downloads como, JDK, ou Java Development Kits, o "kit para desenvolver aplicações Java" disponível gratuitamente no site da Oracle, atualmente recomendado na versão 8, visualizado na Figura 11.

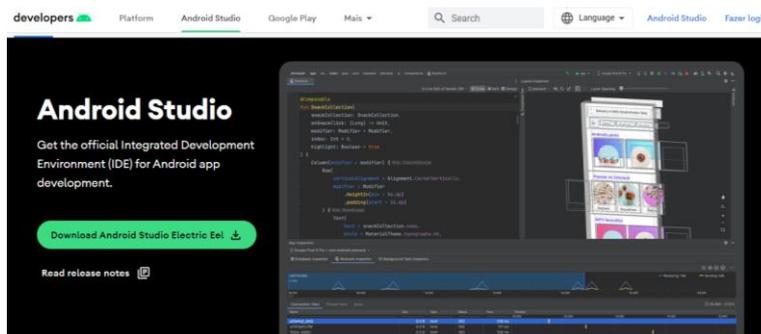
Figura 11 – Java SE 8 Downloads (JDK)



Fonte: (ORACLE, 2023)

Em seguida será necessário o uso de um IDE (Integrated Development Environment, que em português "Ambiente Integrado de Desenvolvimento"). Um IDE não é simplesmente um editor pois integra em um único local a linguagem, o editor, o compilador, a biblioteca e a documentação. Os principais IDEs utilizados por quem programa em Java são: o NetBeans, da própria Oracle, o IntelliJ IDEA, usado como base para Android, e também conhecido por Android Studio, e o Eclipse, projeto em código aberto absorvido pela IBM. Abaixo é possível visualizar na Figura 12 o Android Studio cujo download é possível ser realizado gratuitamente.

Figura 12 – Android Studio



Fonte: (ANDROID, 2023)

## 2.8 Kotlin

Kotlin se originou na JetBrains, a empresa por trás do IntelliJ IDEA, em 2010, e tem código aberto desde 2012, o projeto Kotlin no GitHub tem mais de 770 colaboradores. É uma linguagem de programação “pragmática” de uso geral, livre, de código aberto e estaticamente tipada inicialmente projetada para JVM (Java Virtual Machine) e Android, e combina recursos de programação funcional e orientados a objetos. Ela é focada em interoperabilidade, segurança, clareza e suporte de ferramentas (INFOWORD, 2022).

De acordo com TOLEDO (2019), a Kotlin foi projetada para corrigir alguns problemas conhecidos na Java, trouxe como principal novidade uma apresentação de linguagem null-safe, ou seja, seu compilador tem capacidade de detectar possíveis objetos nulos antes da execução do código, o que evita erros em tempo de execução.

O Android Studio oferece compatibilidade total com o Kotlin. Assim, você pode adicionar arquivos Kotlin ao seu projeto e converter o código da linguagem Java em Kotlin. Em seguida, você poderá usar todas as ferramentas atuais do Android Studio com o código Kotlin, como preenchimento automático, verificador de Lint, refatoração, depuração, entre outras (ANDROID, 2020).

Por ser uma linguagem baseada em Java é necessário para o desenvolvimento de aplicativos em Kotlin as instalações da JDK e a IDE IntelliJ IDEA visualizada na Figura 13.

Figura 13 – IntelliJ IDEA



Fonte: (JETBRAINS, 2023)

## 2.9 Aplicações Híbridas

Abordagens de desenvolvimento de plataforma híbrida oferecem aos desenvolvedores de aplicativos móveis a oportunidade de implementar aplicativos em várias plataformas móveis a partir de uma única base de código. Eles permitem escrever código uma vez e executá-lo em qualquer lugar. Sem essas abordagens, os desenvolvedores têm que desenvolver e manter bases de código separadas para cada plataforma que eles querem oferecer seus produtos. Essa abordagem é conhecida como desenvolvimento nativo. Atualmente, o mercado de telefonia móvel é dominado por duas plataformas Google Android e Apple iOS (DORFER; DEMETZ; HUBER, 2020)

*Cross-plataform* é uma abordagem de desenvolvimento de software em que uma aplicação é desenvolvida uma única vez e pode ser utilizada ou transportada para diferentes sistemas. Esse tipo de metodologia surgiu para aumentar a produtividade dos desenvolvedores, reduzir a repetição de código e facilitar a criação de programas que sejam executados em várias plataformas, permitindo às empresas abrangerem uma gama maior de usuários (HEITKÖTTER.; HANSCHKE; MAJCHRZAK, 2012).

Desenvolvimento *Cross-Platform* traduzido para português como multiplataforma, também conhecido como desenvolvimento híbrido, tem como objetivo produzir aplicações, com pouca ou nenhuma modificação, capaz de ser executada em várias plataformas. De acordo com a taxonomia de El-Kassas et al. (2015), as abordagens multiplataforma ou Cross-Platform podem ser classificadas em: cross-compilada, de máquina virtual, baseada na web, em componentes, na nuvem e uma abordagem mista das anteriores. Essas abordagens favorecem em sua maioria uma maior economia, tanto financeira quanto de tempo de produção, ao custo de se utilizar menos o potencial de cada plataforma.

## 2.10 Flutter

De acordo com Neves et al. (2020), em seu artigo sobre uma análise comparativa entre Flutter e React Native como frameworks para desenvolvimento híbrido de aplicativos mobile cita o livro *Flutter in Action* publicado em dezembro de 2019 por WINDMILL, Eric:

Flutter é um SDK mobile (Kit de Desenvolvimento de Software Móvel) e open-source desenvolvido pelo Google, a sua essência é sobre capacitar todos a criar belos dispositivos móveis. Seja você do mundo do desenvolvimento web ou do desenvolvimento móvel nativo, o Flutter facilita mais do que nunca a criação de aplicativos móveis de uma maneira familiar e simplificada. (2019, Cap. 1, p. 1).

Para REIS (2021), diversas são as características do Flutter, dentre elas, algumas que merecem destaque são:

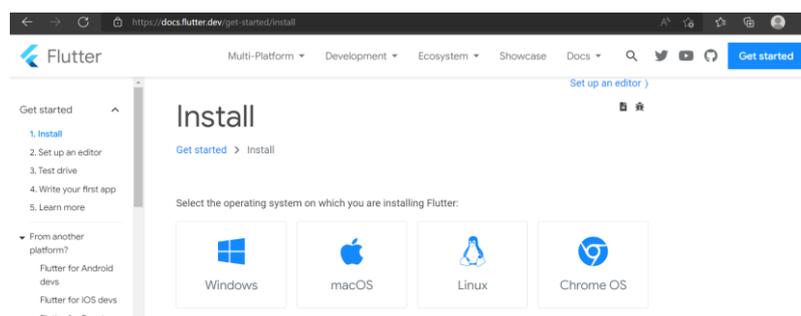
- Multiplataforma – Desenvolvimento de aplicações em qualquer sistema operacional (Windows, Linux e MacOS);
- Criação de aplicações nativas a partir de um único código base – Desenvolvimento de aplicações nativas para Android e iOS;
- Acesso direto aos recursos nativos do sistema – Uma aplicação criada com Flutter possui acesso nativo aos recursos do dispositivo (câmera, wifi, memória, entre outros);
- Maior desempenho – As aplicações criadas com Flutter possuem um maior desempenho quando comparadas ao React Native, por exemplo, todo código-fonte

é transformado em código nativo (REIS, 2021).

Por utilizar como base a linguagem Dart que é orientada a objetos o Flutter é simples e tem uma curva de aprendizagem relativamente curta, outra facilidade é que além de ser usado para o desenvolvimento mobile, é possível produzir sites Web e aplicações Desktop para Windows, Linux, macOS, gerando uma maior interoperabilidade. Mas o que foi apontado por ALURA (2020) como negativo é que, ao rodar benchmarks para comparar os códigos com uma linguagem nativa, é notado que o Flutter é levemente mais lento, portanto, em caso de projetos com o foco exclusivo para apenas uma plataforma, Android ou iOS, o Flutter pode não ser a melhor opção.

Assim como as demais aplicações citadas, o Flutter necessita para seu desenvolvimento da instalação de uma IDE, e atualmente há muitos softwares disponíveis, como os já citados Xcode, Android Studio, IntelliJ IDEA, do mesmo modo que a VS Code. É indispensável também a instalação do SDK do Flutter, disponível gratuitamente no próprio site do Flutter, visualizado na Figura 14.

Figura 14 – Página de Download Flutter



Fonte: (FLUTTER, 2023)

## 2.11 React Native

React Native é o framework de código aberto para desenvolvimento de aplicações da empresa Facebook disponibilizado em 2015, durante a *React.js* Conf 2015 (OCCHINO, 2015). Possui código aberto, em que os aplicativos desenvolvidos em Javascript, mas se portam como aplicações nativas. É uma extensão do Framework React (FACEBOOK, 2018), do qual o intuito é facilitar a criação de interfaces gráficas. O React Native segue os mesmos padrões de código do React, sendo assim, é possível criar funções que podem ser padronizadas e utilizadas de formas genéricas dentro do projeto em que está trabalhando (ALCANTARA, 2018).

Ela é construída com a popular biblioteca ReactJS, utilizando o modelo DOM (Document Object Model), que manipula o aplicativo de uma forma declarativa, o que fornece uma maior abstração para o desenvolvedor e torna o processo de criação do app mais simples.

O DOM virtual armazena uma cópia do modelo de objetos e altera somente as partes que sofreram mudanças do aplicativo, o que traz a vantagem de o desenvolvedor não necessitar esperar o aplicativo completo ser recompilado, bastando apenas recarregar a página para visualizar a mais nova versão (NUNES et al, 2021).

A essência do React Native é a camada de abstração conhecida como “bridge” (ponte), que permite o acesso de APIs específicas para cada plataforma que irão renderizar os componentes da interface. Enquanto nas aplicações web ReactJS a ponte acessa o DOM do browser, no React Native a ponte acessa APIs em swift ou objective-C no iOS, e Java ou Kotlin no Android. Um interpretador Javascript é utilizado em conjunto com a ponte para renderizar a interface de usuário, transformando as marcações típicas das linguagens web em componentes visuais da interface nativa do sistema operacional (NUNES et al, 2021).

Uma desvantagem desta Framework é a utilização de pontes para a compilação da linguagem React ou Javascript para a linguagem nativa do SO, sendo isto responsável por problemas de desempenho. Entretanto, uma vantagem a ser citada é a forma como é estruturado em componentes e ao alto e ativo apoio da comunidade, que reflete no alto grau de reutilização e personalização de funções e telas (LIMA, 2019).

Para iniciar o desenvolvimento em React Native são necessárias as seguintes instalações: Chocolatey, Node.js na versão LTS e JDK 11 - Disponíveis para download no próprio terminal powershell (Windows), e para o Android Studio como IDE.

## 2.12 Aplicações Web

As aplicações Web, conhecidas como PWA - *Progressive Web App*, termo usado para denotar um conjunto de processos de desenvolvimento de software. É uma aplicação web aprimorada que utiliza algumas tecnologias que permitem um comportamento nativo no dispositivo através de APIs do navegador (FRANSSON; DRIAGUINE, 2017).

Segundo REIS (2019), devido às aplicações PWA fazerem uso essencialmente de características chaves como Service Worker e Application Shell, e necessitarem apenas de um navegador a execução pode ocorrer em qualquer dispositivo que tenha o navegador instalado, ou seja, uma variedade de plataformas suportadas.

O Service Worker é um script JS responsável por executar no background separado do navegador. Esse script é capaz de responder a eventos que são executados dentro da aplicação, isso permite que se tenha um maior controle do cache da aplicação em geral. Através do service worker é possível armazenar o conteúdo necessário para a aplicação funcionar razoavelmente mesmo sem acesso à internet e permitir a experiência do usuário mesmo offline (REIS, 2019). Um outro fator bastante importante na melhoria de desempenho é o Application Shell que é o

mínimo de HTML, CSS e Javascript necessário para que seja apresentada a interface do usuário (MISHRA, 2016). Combinando esses dois conceitos é possível obter o conteúdo necessário a ser mostrado e também a estrutura básica do layout na qual esse conteúdo será exibido de uma maneira mais otimizada (REIS, 2019).

Ao falarmos em web apps, falamos em aplicativos mais leves, que consomem menos memória do dispositivo, já que não são instalados no mesmo, pois o serviço que será utilizado está hospedado em um link que é acessado pelo ícone de acesso a ele (PRADO, 2019).

### 3 METODOLOGIA

O capítulo se dedica a esclarecer os métodos e técnicas adotadas ao longo da pesquisa para elaboração do trabalho científico.

O método é conjunto das atividades sistemáticas e racionais que com maior segurança e economia, permite alcançar o objetivo – conhecimentos válidos e verdadeiros – traçando o caminho a ser seguido, detectando erros e auxiliando as decisões do cientista (MARCONI; LAKATOS, 2010). De acordo com NASCIMENTO (2015), a pesquisa científica pode ser diferenciada quanto à **natureza, aos métodos (ou abordagens metodológicas), quanto aos objetivos e quanto aos procedimentos.**

Quanto à natureza, o trabalho se desenvolve através da pesquisa aplicada que, caracteriza-se por seu interesse prático, isto é, que os resultados sejam aplicados ou utilizados, imediatamente, na solução de problemas que ocorrem na realidade (MARCONI; LAKATOS, 2010).

Em relação às abordagens metodológicas, trata-se de uma pesquisa quantitativa. A pesquisa quantitativa se preocupa com a representatividade numérica, utiliza-se medição objetiva e quantificação, verifica-se nela a presença do emprego de dados estatísticos, na coleta de dados tem por finalidade medir as relações entre variáveis. Além disso, quando retrata os resultados quantifica-os, transformando-os em dados estatísticos. Muito utilizado em pesquisas que medem opiniões, atitudes e preferências como comportamentos (ZANELLA, 2009).

No tocante aos objetivos, o trabalho segue a linha da pesquisa exploratória, pois proporciona ao pesquisador obter maiores informações sobre determinado assunto, facilitando a delimitação do tema de trabalho e a definição dos objetivos, e ainda, provoca a formulação das hipóteses da pesquisa, as quais a investigação pode resultar na descoberta de um novo tipo de enfoque para o trabalho que se tem em mente (ANDRADE, 1999).

Por fim, acerca dos procedimentos utilizados no trabalho, trata-se de uma pesquisa documental.

A pesquisa documental trilha os mesmos caminhos da pesquisa bibliográfica, não sendo fácil por vezes distingui-las. A pesquisa bibliográfica utiliza fontes constituídas por material já elaborado, constituído basicamente por livros e artigos científicos localizados em bibliotecas. A pesquisa documental recorre a fontes mais diversificadas e dispersas, sem tratamento analítico, tais como: tabelas estatísticas, jornais, revistas, relatórios, documentos oficiais, cartas, filmes, fotografias, pinturas, tapeçarias, relatórios de empresas, vídeos de programas de televisão, etc. (FONSECA, 2002, p. 32).

### 3.1 Coleta de Dados

O processo de coleta de dados decorreu por intermédio de documentos e informações disponibilizadas em sites, sendo eles:

#### 3.1.1 GitHub

O Git é um sistema de controle de versão distribuído e amplamente adotado. O Git nasceu e foi tomando espaço dos outros sistemas de controle. Seu criador principal é o mesmo que o do Linux: Linus Torvalds, e ganhou o coração das pessoas que trabalham com open source (ALURA, 2023). Além de ser uma plataforma de hospedagem (nuvem) de código-fonte e arquivos com um sistema de controle de versão (SCV) usando o Git, que permite a seus usuários o compartilhamento e visualização de projetos privados ou *open source*, este site disponibiliza informações como reportagens, blogs e rankings, demonstrativos baseados no fornecimento dos dados de sua própria comunidade que atualmente possui 94 milhões de usuários, sendo 3 milhões compostos por brasileiros (ALECRIM, 2022). Ademais, com presença de mais de 90% das empresas que participam do ranking da Fortune 100 nesta plataforma o volume de dados e as conclusões tiradas por cada edição publicada em seu relatório *State of the Octoverse* são consideradas de elevada relevância para o mercado, portanto, o GitHub é utilizado para estudar as aplicações mobiles.

#### 3.1.2 Stack Overflow

O Stack Overflow é um site de perguntas e respostas favorito de programadores, estudantes e entusiastas da programação, funcionando como um fórum, com questões agrupadas por temas, portanto é possível a organização e busca por linguagem de programação, todo o conteúdo do Stack Overflow é gerenciado pela própria comunidade com um sistema de votação similar ao do Reddit: os usuários votam “para cima” ou “para baixo” nas perguntas e respostas, dando destaque, ou não, para os conteúdos mais relevantes (MELO, 2021).

Ademais, anualmente é publicado o relatório *Stack Overflow Developer Survey* baseado em uma pesquisa com os desenvolvedores de software de 180 países ao redor do mundo. De acordo com a OVERFLOW (2022), os entrevistados são recrutados principalmente por meio de canais pertencentes ao Stack Overflow. As principais fontes dos entrevistados são mensagens no local, postagens em blog, listas de e-mail, banners e postagens de mídia social. Através dos relatórios divulgados pela Stack Overflow que a pesquisa se baseou na análise da utilização e popularidade das linguagens entre os desenvolvedores mobiles.

### 3.1.3 Data.ai

A Data.ai, anteriormente conhecida como App Annie, foi fundada por Bertrand Schmitt e Bjørn Stabell em 2010, é uma empresa unificada de dados e inteligência artificial (IA) que combina dados de consumidores e mercado para fornecer informações baseadas em IA sobre dados para o mercado de aplicativos móveis. Seus insights são utilizados por desenvolvedores, editores e anunciantes para gerar receita e engajamento de negócios (GOLDEN, 2023).

A primeira e única plataforma de IA de dados unificados da indústria unifica seus dados first-party com estimativas de mercado abrangentes em uma experiência de análise envolvente e capacitadora. Munido com insights oportunos e significativos movidos por IA, você pode criar de forma rápida e fácil estratégias vencedoras de investimentos, go-to-market, concorrência, campanhas e publicidade. Nossos clientes estão impactando de forma positiva CAC, LTV, ROAS e ROI (DATA.AI, 2023). Na Figura 15 é mostrado o esquema de sua produção de análise de dados.

Figura 15 – Esquema de Produção de Dados Data.ai



Fonte: (DATA.AI, 2023)

Anualmente a Data.ai publica relatórios divulgando o panorama do mercado mobile mundial em vários aspectos, como é o caso do *The State of Mobile 2022 - Brazil*, relatório exclusivo direcionado para o Brasil, do qual aponta o grande crescimento e utilização dos usuários nas plataformas. Tais relatórios serviram como base na análise inicial dos nichos e a confirmação das demandas de mercado.

Figura 16 – Capa do The State of Mobile 2022 – Brazil



Fonte: (DATA.AI, 2023)

#### 3.1.4 LinkedIn

Fundado por Reid Hoffman, e oficialmente lançado em 2003, o LinkedIn conta atualmente com um modelo de negócios diversificado, com receita proveniente de assinaturas, vendas de publicidade e de soluções de recrutamento. Em dezembro de 2016, a Microsoft concluiu sua aquisição do LinkedIn, reunindo o maior serviço de nuvem profissional do mundo com a maior rede profissional do mundo, com mais de 850 milhões de usuários em 200 países e territórios (LINKEDIN, 2023).

Sendo atualmente o principal site para publicação e busca por vagas de emprego, a pesquisa utilizou a plataforma para analisar a oferta e tipo de vagas para o desenvolvimento mobile.

#### 3.1.5 Glassdoor

A Glassdoor foi fundada por Rich Barton, Robert Hohman e Tim Besse em 2007 e, em 2018, a empresa foi adquirida pela japonesa Recruit Holdings (GOLDEN, 2023). Sendo um site de vagas de emprego gratuito, permite buscas e candidaturas a vagas, além de conter informações sobre empresas como: cargos, entrevistas, salários, avaliações, prós e contras, benefícios e ambiente de trabalho. O serviço é feito de forma colaborativa, os próprios funcionários e ex-funcionários adicionam comentários sobre a empresa, trazendo conhecimento sobre o local de trabalho de forma anônima (TECHTUDO, 2012). Em 2016, a Glassdoor adquiriu a Love Mondays, site brasileiro de avaliações de empresas, salários e vagas de emprego, que ajuda profissionais a saber como é trabalhar em mais de 75.000 empresas, conhecer faixas salariais e candidatar-se a vagas de emprego no Brasil (EXAME, 2016).

O site é um aliado na busca de novas oportunidades no mercado de trabalho, pois segundo a TEAMGUIDE (2021), são mais de 900 mil organizações para conhecer. Por fim, a

principal informação extraída da Glassdoor é o fornecimento da média salarial fornecida pelo site e utilizada como base na pesquisa.

## 4 RESULTADOS

Neste capítulo, será apresentado a estrutura dos métodos utilizados na obtenção dos dados da pesquisa através das seções de Aplicações, Empregabilidade, Profissionais e Mercado.

### 4.1 Aplicações

De acordo com o relatório *Stack Overflow Developer Survey 2022*, cuja pesquisa foi realizada durante os meses de maio e junho daquele mesmo ano, com 73.268 desenvolvedores de software de 180 países ao redor do mundo, os entrevistados foram recrutados principalmente por meio de canais pertencentes ao Stack Overflow e as respostas referentes a salários foram convertidas para dólares americanos, usando taxa de câmbio em 24 de maio de 2022, assim como as remunerações foram convertidas para salários anuais, assumindo 12 meses e 50 semanas úteis (OVERFLOW, 2022).

Dentre as tecnologias mais populares, no tópico de linguagens de programação, script e marcação, foram retiradas as informações das seis aplicações pesquisadas neste trabalho para desenvolvimento mobile visualizadas na Tabela 1

Tabela 1 – Linguagens mais populares

Linguagem	Java	Kotlin	Swift	Objective-C	Flutter	React Native
Total	23.644	6.507	3.489	1.698	5.799	5.765

Fonte: (OVERFLOW, 2022).

É importante observar que a linguagem Java tem um resultado expressivamente superior às demais devido a sua utilização não ser apenas empregada ao desenvolvimento mobile, portanto não é possível ter uma confirmação confiável sobre sua popularidade superior às demais. Com relação as linguagens nativas para iOS, Swift e Objective-C, é notório a migração para a utilização do Swift, confirmando como a linguagem oficial mais atual da Apple. Nas aplicações híbridas receberam um número muito próximo de votos, não sendo possível notar diferença relevante entre sua popularidade.

Outra informação extraída do relatório da Stack Overflow é no tópico em que foi perguntado aos desenvolvedores as linguagens que amavam, odiavam e que tinham o interesse em começar a aprender, os resultados em relação às linguagens mobiles estão destacados na Tabela 2 abaixo:

Tabela 2 – Linguagens amadas, odiadas e querem aprender

Linguagem	Java	Kotlin	Swift	Objective-C	Flutter	React Native
Ama	45,75%	63,29%	62,88%	23,44%	66,17%	55,98%
Odeia	54,25%	36,71%	37,12%	76,56%	33,83%	44,02%
Quer Aprender	2.679	5.250	2.920	495	6.439	6.223

Fonte: (OVERFLOW, 2022).

Objective-C, além de receber mais de 70% de votos como uma linguagem odiada, o interesse em querer aprender é relativamente baixo, confirmando o seu desuso pelos desenvolvedores mobile que atualmente utilizam como linguagem nativa para iOS o Swift. Outro destaque são as aplicações híbridas, o Flutter recebeu quase 10% de vantagem em relação ao React Native como linguagem amada, quanto ao interesse em querer aprender, com exceção do Kotlin que também teve um número elevado de votos, são as linguagens híbridas que receberam a maior parte dos votos.

## 4.2 Empregabilidade

Na pesquisa da Stack Overflow há também informações sobre os ganhos financeiros dos desenvolvedores de aplicações móveis, conforme exposto na Tabela 3, observou-se média das remunerações anuais utilizando a moeda dólar e o câmbio de maio de 2022.

Tabela 3 – Remuneração média anual em dólar (US\$)

Linguagem	Java	Kotlin	Swift	Objective-C	Flutter	React Native
2021	U\$51.890	U\$55.070	U\$58.910	U\$64.860	U\$32.430	U\$44.160
2022	U\$64.570	U\$69.320	U\$78.470	U\$83.170	U\$42.660	U\$54.000

Fonte: (OVERFLOW, 2022).

Os desenvolvedores que trabalham com linguagens nativas para iOS, Objective-C e Swift ganharam mais nos anos de 2021 e 2022, e tiveram um crescimento em suas remunerações do ano de 2022 de mais de U\$18.000,00 (dezoito mil dólares), quando comparados com a média das remunerações dos desenvolvedores de Android das linguagens nativas Java e Kotlin seus ganhos são aproximadamente 20% superiores em 2022, já com a média das remunerações dos desenvolvedores de aplicação híbridas são de aproximadamente 66% neste mesmo ano. Quanto a diferença dos ganhos entre as linguagens nativas e híbridas é possível notar, ao calcular a média entre os resultados colhidos, a diferença de aproximadamente 52% a mais na remuneração das linguagens nativas do ano de 2022 segundo dados da Stack Overflow.

Na busca por um aprofundamento maior em relação ao cenário nacional, a pesquisa utilizou a plataforma LinkedIn para busca de vagas disponíveis no Brasil utilizando a palavra-chave: *desenvolvedor de aplicativos móveis* e foram encontradas 477 (*quatrocentos e setenta e sete*) vagas no período dos meses de março a abril do ano de 2023. Ao realizar nova pesquisa alterando a palavra-chave para desenvolvimentos em aplicações específicas foram apresentados os seguintes resultados visualizados na Tabela 4.

Tabela 4 – Vagas para desenvolvimento Android e iOS

<b>Palavra-chave</b>	<b>Quantidade</b>
Desenvolvedor Android	193
Desenvolvedor iOS	137
<b>Total</b>	<b>330</b>

Fonte: (LINKEDIN, 2023).

Quando não é especificado na busca o tipo de sistema operacional (Android ou iOS) há o surgimento de mais vagas, devido algumas publicações não especificarem em seu título o sistema que buscam, ou por sua vez necessitarem de profissionais *full Stack*, ou seja, que saibam trabalhar com ambos os sistemas nativos.

A plataforma Glassdoor foi conjuntamente utilizada para pesquisa de vagas disponíveis através do uso da palavra-chave: *desenvolvedor mobile* e foram encontradas 615 (seiscentas e quinze) vagas no período dos meses de março a abril do ano de 2023, na Tabela 5 é demonstrado os resultados da busca com a utilização de palavras-chaves para desenvolvimentos específicos.

Tabela 5 – Vagas para desenvolvimento Android e iOS

<b>Palavra-chave</b>	<b>Quantidade</b>
Desenvolvedor Android	339
Desenvolvedor iOS	233
<b>Total</b>	<b>572</b>

Fonte: (GLASSDOR, 2023).

Assim como observado na plataforma LinkedIn, quando não foi especificado na busca o tipo de sistema operacional (Android ou iOS) ocorreu o surgimento maior do número de disponibilidade de vagas. Já quando filtrado, foi analisado que a oferta para o desenvolvedor

Android é acima de 40% quando comparado ao iOS nas plataformas LinkedIn e Glassdoor.

Tendo como objetivo investigar quais aplicações ofertavam maior número de vagas, a pesquisa delimitou a busca por palavras-chaves através dos tipos de aplicações móveis, conforme exposto nas Tabela 6 e Tabela 7.

Tabela 6 – Vagas para aplicações mobile

<b>Palavra-chave</b>	<b>Quantidade</b>
Desenvolvedor Objective-C	40
Desenvolvedor Swift	120
Desenvolvedor Java	164
Desenvolvedor Kotlin	92
Desenvolvedor Flutter	34
Desenvolvedor React Native	162

Fonte: (LINKEDIN, 2023).

Tabela 7 – Vagas para aplicações mobile

<b>Palavra-chave</b>	<b>Quantidade</b>
Desenvolvedor Objective-C	0
Desenvolvedor Swift	114
Desenvolvedor Java	171
Desenvolvedor Kotlin	191
Desenvolvedor Flutter	105
Desenvolvedor React Native	112

Fonte: (GLASSDOOR, 2023).

Nas linguagens nativas foram encontradas poucas vagas para Objective-C na plataforma LinkedIn e nenhuma vaga na plataforma Glassdoor, nas aplicações híbridas o React Native apresentou um número maior de vagas quando comparado ao Flutter na plataforma LinkedIn. Foi observado a oferta de vagas na linguagem Java, mesmo estudos apontando a utilização do Swift como substituto para o desenvolvimento nativo Android, Java continua tendo uma um número relevante de vagas no mercado nacional, sendo superior ao Kotlin.

A plataforma Glassdoor oferece publicamente informações sobre as remunerações de alguns tipos de cargos e carreiras através dos dados coletados de empresas e usuários

participantes de sua comunidade, portanto, através desses dados foi possível obter as médias salariais, conforme exposto na Tabela 8.

Tabela 8 – Média salarial

<b>Palavra-chave</b>	<b>Média Salarial</b>	<b>Intervalo Júnior – Pleno - Sênior</b>	<b>Nº de salários coletados</b>
Desenvolvedor Mobile	R\$5.010,00	R\$2.087 - R\$ 29.678	346 Salários
Desenvolvedor iOS	R\$6.143,00	R\$1.000 - R\$ 29.600	314 Salários
Desenvolvedor iOS	R\$6.090,00	R\$3.000 - R\$ 16.795	250 Salários

Fonte: (GLASSDOOR, 2023).

Com relação à média salarial o desenvolvedor iOS e Android tem uma remuneração maior de aproximadamente 22% comparado ao desenvolvedor mobile, mas o crescimento das remunerações move-se de acordo com o grau de experiência do profissional, conforme demonstrado na Tabela 9.

Tabela 9 – Média salarial

<b>Palavra-chave</b>	<b>Júnior</b>	<b>Pleno</b>	<b>Sênior</b>
Desenvolvedor Mobile	R\$3.183,00	R\$6.572,00	R\$11.564,00
Desenvolvedor iOS	R\$4.314,00	R\$7.502,00	R\$11.101,00
Desenvolvedor iOS	R\$4.200,00	R\$8.000,00	R\$12.257,00

Fonte: (GLASSDOOR, 2023).

Mesmo com maiores remunerações, os números não mostram uma diferença significativa entre Android e iOS, e outro fator a ser considerado é a dificuldade encontrada pelo desenvolvedor iOS, pois devido ao seu sistema operacional utilizar apenas aparelhos da Apple, cujo o custo é elevado para ser adquirido no Brasil, resulta em um alto investimento em equipamentos, muito mais caro do que os desenvolvedores Android, conseqüentemente justifica um número menor desses profissionais disponíveis no mercado de trabalho.

### 4.3 Profissionais

Com relação ao número de profissionais encontrados atualmente no mercado, a pesquisa baseou sua busca na plataforma LinkedIn, devido ao grande número de usuários utilizando esta

ferramenta para publicação e portfólio de currículos e acesso público de alguns dados demonstrados na Tabela 10. Ao utilizar a palavra-chave *desenvolvedor mobile* na busca de pessoas foram encontrados 138.000 (*cento e trinta e oito mil*) usuários cadastrados, mas alterando a busca para a palavra-chave *desenvolvedor de aplicativos móveis* foram encontrados apenas 3.000 (*3 mil*) usuários cadastrados, já os resultados com as palavras-chaves *desenvolvedor Android* e *desenvolvedor iOS* tiveram os respectivos resultados de 27.00 (*vinte sete mil*) e 7.700 (*sete mil e setecentos*) usuários encontrados.

Tabela 10 – Desenvolvedores mobiles cadastrados no LinkedIn

Linguagem	Java	Kotlin	Swift	Objective-C	Flutter	React Native
Desenvolvedor + Aplicação	108.000	9.100	5.000	2.700	12.000	23.000
Desenvolvedor Mobile + Aplicação	27.000	4.400	28.000	25.000	136.000	10.000

Fonte: (LINKEDIN, 2023).

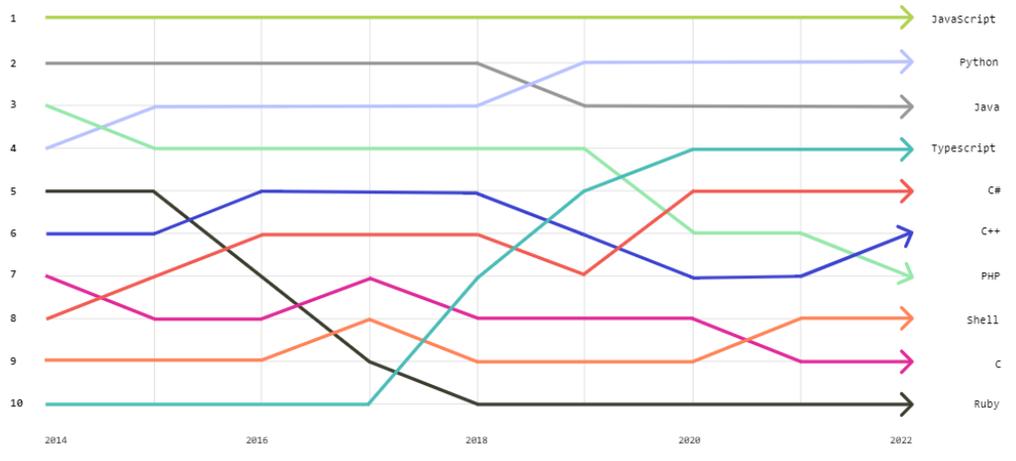
Na busca pelas as aplicações específicas foram utilizadas dois tipos de palavras-chaves, *desenvolvedor “+ aplicação”* e *desenvolvedor mobile “+ aplicação”*, devido a considerar que alguns usuários utilizam ou não o termo “*mobile*” para descreverem suas habilidades.

É possível observar na Tabela 10 um número expressivo de desenvolvedores utilizando a aplicação híbrida Flutter, quando comparado com as demais aplicações, até mesmo as nativas, foi observado na maioria dos perfis dos desenvolvedores mobiles a aplicação Flutter como habilidade. Dentre as linguagens nativas para Android o Java continua sendo a mais popular, considerando ser uma linguagem antiga, sendo base para muitas outras linguagens, portanto muitos programadores tem conhecimento em Java e o Kotlin ainda tem números bem abaixo quando comparados ao Java. Já para iOS notou-se a consolidação do Swift substituindo o Objective-C, a maioria dos perfis dos profissionais que declaravam a habilidade da linguagem Objective-C também possuíam conhecimento em Swift.

#### 4.4 Mercado

No relatório *State of the Octoverse* da plataforma GitHub referente ao ano de 2022 é mostrado o cenário da utilização das linguagens e aplicações no mercado mundial visualizado na Figura 17.

Figura 17 – State of the Octoverse GitHub



Fonte: (GITHUB, 2022)

Entre os destaque do relatório está a estabilidade que a linguagem Java teve nas últimas décadas, pois esteve ocupando o ranking entre as três principais linguagens desde seu lançamento em 1995, mesmo enfrentando algumas dificuldades no decorrer dos anos, hoje em dia a linguagem e seu ecossistema maior aceleraram o ritmo para atender às necessidades do futuro de várias maneiras, e tendo como exemplo clientes como as empresas Amazon, Google, Netflix, Pinterest, Spotify, Square e Zoom que empregam o Java em suas vastas bases de código. Em vez de impedir a inovação, o ecossistema Java floresceu. A mudança estimulou vários provedores de nuvem, como Amazon, Microsoft, IBM, Alibaba Cloud e Huawei, a oferecerem suas próprias versões do OpenJDK testadas e suportadas pelo Technology Compatibility Kit (TCK) (MELANSON, 2022).

Na Figura 18 retirado do mesmo relatório é mostrado as linguagens que mais crescem e o Kotlin aparece em 9º lugar, com crescimento de 22, 9%. Segundo o *State of the Octoverse 2022*, o desenvolvimento móvel continua a dominar, se tratando de Android a linguagem Kotlin vem sendo popular e nas aplicações híbridas o Flutter. Essa tem sido uma tendência em larga escala nos últimos anos e ainda não está diminuindo, especialmente agora que os aplicativos móveis Android são executados no Windows 11 (GITHUB, 2022).

Figura 18 – Linguagem de Maior Crescimento do State of the Octoverse

GROWTH IN PROGRAMMING LANGUAGES 2021-2022

01 HCL	56.1%
02 Rust	50.5%
03 TypeScript	37.8%
04 Lua	34.2%
05 Go	28.3%
06 Shell	27.7%
07 Makefile	23.7%
08 C	23.5%
09 Kotlin	22.9%
10 Python	22.5%

Fonte: (GITHUB, 2022)

Outra tendência apontada pelo GitHub são as contribuições pela comunidade em projetos *open source* disponibilizados por grandes empresas, dentre elas a Flutter e o Facebook com o React Native visualizados na Figura 19 e Figura 20.

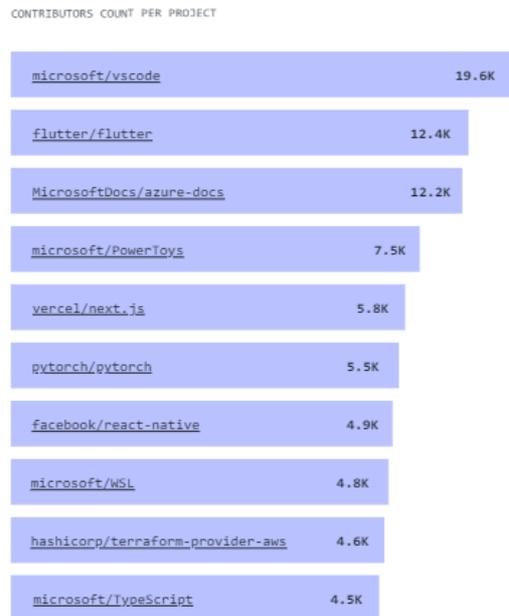
Figura 19 – Maior Crescimento Open Source por Contribuidores

TOP OPEN SOURCE PROJECTS BY CONTRIBUTORS

01 <a href="#">microsoft/vscode</a>	19.8K
02 <a href="#">home-assistant/core</a>	13.5K
03 <a href="#">flutter/flutter</a>	12.4K
04 <a href="#">MicrosoftDocs/azure-docs</a>	12.3K
05 <a href="#">microsoft/PowerToys</a>	7.5K
06 <a href="#">vercel/next.js</a>	5.9K
07 <a href="#">NixOS/nixpkgs</a>	4.8K
08 <a href="#">microsoft/TypeScript</a>	4.56K
09 <a href="#">mui/material-ui</a>	4.53K
10 <a href="#">tensorflow/tensorflow</a>	4.4K

Fonte: (GITHUB, 2022)

Figura 20 – Maior Crescimento Open Source por Projeto



Fonte: (GITHUB, 2022)

A estratégia dessas empresas em participarem com projetos no GitHub populariza suas linguagens, como é o caso da Flutter que está em 3º lugar no ranking de projetos open source e 2º lugar no ranking de usuários participando.

#### 4.4.1 Experiência Nubank

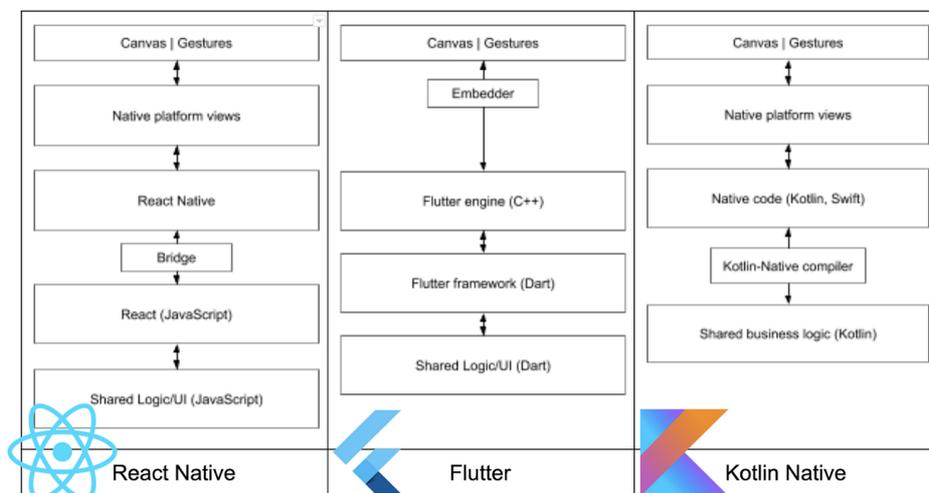
Fundada oficialmente em maio de 2013, por David Vélez, Cristina Junqueira e Edward Wible, a Nubank oferece um cartão de crédito aceito na maioria dos estabelecimentos do país, em uma parceria com a Mastercard. Todo o atendimento concentra-se em um aplicativo para iOS e Android. O programa registra os gastos das pessoas e, como diferencial, as auxilia a resolver eventuais problemas com as compras (ÉPOCA, 2015).

Em um artigo publicado em 2019 pelo website Medium, o diretor de engenharia da Nubank, Alexandre Freire, detalhou sobre o projeto de mudança que estava ocorrendo em relação a migrar todo desenvolvimento para Flutter, pois ao iniciarem em 2013, trabalhavam com linguagens nativa, suportando ambas as plataformas Android e iOS. Ao explicar um dos motivos da mudança, Freire destaca a escassez de profissionais desenvolvedores nativos, *“poucos especialistas em desenvolvimento nativo disponíveis. E também não era fácil de contratar, nós vimos (e ainda vemos) uma competição feroz no mercado por esses*

*professionais*”. Acreditando que times especializados eram mais difíceis de se manter, a Nubank resolveu apostar em equipes autônomas, ágeis e multifuncionais, que trabalhassem ponta a ponta, sem ter a necessidade de um especialista em alguma etapa do projeto, acreditando assim agregar mais valor e de forma mais rápida.

Segundo Freire (2019), o objetivo era fazer uma escolha onde, independente da especialização dos membros, os times seriam autônomos e produtivos ao desenvolver aplicações mobile e entregar valor a uma única arquitetura, linguagem de programação e conjunto de padrões, foram montadas 3 equipes com desenvolvedores mobile experientes, nas seguintes aplicações visualizados na Figura 21.

Figura 21 – Quadro de Divisão de Equipes por Aplicações



Fonte: (FREIRE, 2019)

Dentre os critérios avaliados alguns dos mais importantes foram:

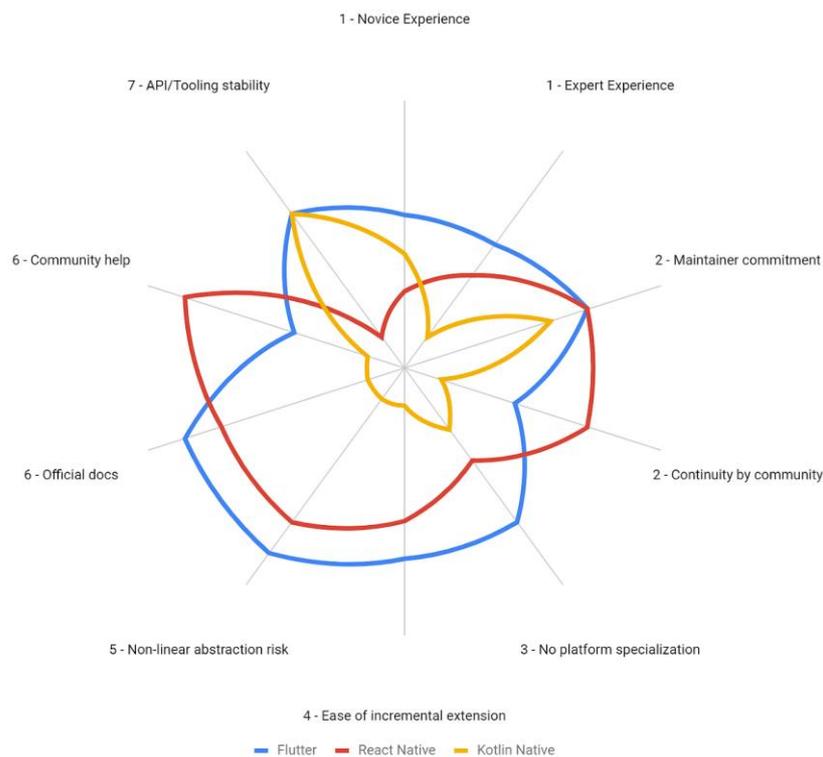
- Experiência dos desenvolvedores: O que possibilita um desenvolvedor a ser produtivo e entregar valor? Exemplos: hot reload; visualização de componentes; ferramentas de depuração; integração da IDE; e ferramentas de testes.
- Viabilidade a longo prazo: Descrever o nível de confiança no futuro da plataforma. O mantenedor irá continuar suportando-a à longo prazo (5 anos)? Qual a probabilidade de a comunidade suportar o projeto caso o mantenedor decidir abandoná-lo?
- Não necessitar especialização na plataforma: Um engenheiro deveria ser capaz de escrever código mobile para o produto sem diferenciar entre Android e iOS. O código tem uma aparência e se comporta igual no Android e iOS, com pouca

ocorrência de problemas específicos de SO?

- Custo de abstração incremental: O custo de estender a plataforma para cada tarefa do produto e a dificuldade de centralizar o trabalho em extensões, se necessário. O quão difícil é adicionar um novo componente? Nós criaríamos uma dependência em um time horizontal de plataforma?
- Risco de abstração não linear: Risco de repentinamente precisar de grandes e desproporcionais reescritas na nossa abstração interna. Nós precisaríamos fazer mudanças não triviais ao longo de toda nossa base de código para suportar um novo componente no NuDS (Nubank Design System)? (FREIRE, 2019)

Após diversas etapas, foi criado um relatório da pesquisa, agrupando os resultados e detalhando como cada critério foi avaliado, na Figura 22 é possível visualizar a decisão através dos 7 principais critérios.

Figura 22 – Resultado de Desempenho das Aplicações



Fonte: (FREIRE, 2019)

Em 2021, Alexandre Freire, Diretor de Engenharia do Nubank, declarou que: “Nós amamos Flutter no Nubank. Tornou-se 100% parte da nossa cultura quando falamos de

*desenvolvimento mobile”.*

#### 4.4.2 Experiência Airbnb

Fundado em 2008 por Brian Chesky, Joe Gebbia e Nathan Blecharczyk na cidade de São Francisco, na Califórnia, o Airbnb tem a proposta de permite que seus usuários aluguem imóveis, os ocupando totalmente ou apenas uma parte, como uma forma de acomodação extra. O site fornece uma plataforma de busca e reservas entre a pessoa que oferece a acomodação e o turista que busca pela locação (WIKIPÉDIA, 2023).

Em uma publicação em 2018 em no website Medium, o Airbnb descreveu sobre sua experiência ao longo dos anos entre o website e a migração para utilização do app.

Desde que nossos três primeiros hóspedes ficaram na Rausch Street em 2008, o uso de dispositivos móveis aumentou de zero para milhões de reservas por ano. Nossos aplicativos oferecem aos anfitriões a capacidade de gerenciar seus anúncios em movimento e fornecem aos viajantes inspiração para descobrir novos lugares e experiências na ponta dos dedos. Para acompanhar o ritmo acelerado do uso móvel, aumentamos nossa equipe para mais de 100 engenheiros móveis para permitir novas experiências e melhorar as existentes (AIRBNB, 2018).

Em 2016, o Airbnb adotou o React Native e, uma das principais justificativas para essa escolha foi devido à grande demanda do número de usuários de smartphones e a falta de desenvolvedores móveis suficientes para compor suas equipes. Devido ao website ser construído principalmente na linguagem React, o React Native foi considerado a melhor opção na época. Após 2 anos de utilização, diferentes equipes tiveram uma ampla gama de experiências com o React Native, que segundo o Airbnb, provou ser uma ferramenta incrível às vezes, enquanto apresentava desafios técnicos e organizacionais em outras.

A maioria dos engenheiros é proficiente em uma ou duas plataformas. É extremamente raro alguém ser altamente competente em Android, iOS e React. Mesmo que a grande maioria do trabalho em um ambiente maduro do React Native seja feito com JavaScript e React, há momentos em que construir ou depurar algo requer uma pesquisa nativa. Essas situações podem fazer com que os engenheiros fiquem sem experiência tentando depurar um problema em uma plataforma que nunca usaram. Isso piora quando um engenheiro não tem certeza de onde procurar devido à dificuldade de atribuição da causa raiz (AIRBNB, 2018).

Em meados do ano de 2018, foi anunciado a mudança para o uso das linguagens nativas, justificando que não conseguiram atingir todos os objetivos originais planejados com o React Native, o que faria com que continuar investindo nesse projeto fosse um desafio e anunciaram que a partir do ano de 2019 não estariam mais desenvolvendo em aplicações híbridas.

Muitas das dificuldades que encontramos foram devido à abordagem do modelo híbrido que adotamos. No entanto, nossa escala nos permitiu assumir e resolver alguns problemas difíceis que empresas menores podem não ter tido tempo de resolver. Fazer o React Native funcionar perfeitamente com o nativo é possível, mas desafiador. Cada empresa que usa o React Native terá uma experiência que é uma função única da

composição de sua equipe, aplicativo existente, requisitos do produto e maturidade do React Native (AIRBNB, 2018).

A experiência que o Airbnb relata mostra a dificuldade que muitas empresas encontram ao decidir qual plataforma utilizarem, e mesmo quando decidem é possível haver alterações ao longo do caminho. Além de ser uma decisão importante e depender de fatores exclusivos de cada equipe de desenvolvimento, não pode ser o veredito para poder ser aplicada a todos os formatos de empresa.

Em conclusão, mesmo trabalhando com híbrido ou nativo, nota-se um padrão nos casos da Nubank e Airbnb, ambas não deixaram de trabalhar, pesquisar e até mesmo investir em outras tecnologias além das que já utilizavam, o que pode vir a ser uma *carta na manga* para muitas empresas, pois se tratando do ramo de tecnologia a inovação é um recurso essencial, assim como novas formas de pensar sobre o desenvolvimento móvel.

## 5 CONSIDERAÇÕES FINAIS

Ao longo da pesquisa foi possível observar alterações no mercado de desenvolvimento de software aplicativos mundial, principalmente após o surgimento da pandemia e as alterações de comportamento dos usuários, comprovado pelo aumento do tempo de uso e downloads nos apps e gerando novas demandas para o mercado.

A migração de websites para *apps* tornou-se uma realidade para muitas empresas e a discussão de como realizar a construção desses *apps* ainda é avaliada e decidida de forma peculiar para atender a cada projeto e suas peculiaridades, pois em um amplos cenários, algumas empresas optaram em trabalhar com suas equipes utilizando aplicações híbridas, da mesma forma que outras empresas escolheram dividir suas equipes para utilizarem aplicações nativas, portanto, até o momento, essa decisão é conclusiva de forma particular, onde cada organização avalia sua capacidade de desenvolvimento, profissionais, investimento, demanda, dentre outros critérios.

Foi possível concluir a evolução de algumas linguagens como Kotlin que surgiu como linguagem nativa para substituir o uso do Java na construção dos apps para Android, mas notou-se ainda muitas vagas e desenvolvedores utilizando o Java, possivelmente essa migração ainda esteja ocorrendo de forma mais lenta no Brasil e merece maior investigação e aprofundamento, continuidade em uma nova pesquisa. De maneira oposta, a linguagem Swift se mostra mais consolidada e utilizada por desenvolvedores e empresas. Com relação as aplicações híbridas, o Flutter merece destaque e continuidade em uma pesquisa futura, por ser apontado como favorito pela maioria dos desenvolvedores brasileiros na plataforma LinkedIn.

Sobre as remunerações oferecidas no mercado, foi possível notar uma diferença expressiva no mercado internacional entre desenvolvedores iOS ter ganhos superiores ao desenvolvedor Android, de outro modo, no Brasil a diferença é pequena, as variações salariais se mostraram mais relevantes quando comparadas a experiência dos desenvolvedores em relação as suas habilidades sem diferenciá-los por aplicação.

Por se tratar de uma pesquisa que utilizou como método de procedimento técnico a pesquisa documental houveram algumas limitações baseadas no aprofundamento da coleta de dados, portanto como recomendação para a continuidade de uma futura pesquisa a utilização do método de estudo de caso, levantamento ou experimental, com o foco em alguns questionamentos ainda não conclusivos como: o aprofundamento da aplicação Flutter e seus favoritismo junto aos profissionais desenvolvedores, para que tipo de empresa é mais vantajoso as aplicações híbridas ou nativas e novas aplicações para o desenvolvimento mobile.

## REFERÊNCIAS

ABES. **Dados do Setor**. Associação Brasileira das Empresas de Software. Brasil: [s.n.], 2022. Disponível em: <https://abes.com.br/dados-do-setor/>. Acesso em: 19 maio 2023.

AIRBNB. **React Native at Airbnb**. [S.l.: s.n.], 2018. Disponível em: <https://medium.com/airbnb-engineering/react-native-at-airbnb-f95aa460be1c>. Acesso em: 18 maio 2023.

ALCANTARA, C. React Patterns — Começando pelo Render Props. **Medium**, 2018. Disponível em: <https://medium.com/collabcode/react-patterns-come%C3%A7ando-pelo-render-props-e0040ef723ce>. Acesso em: 19 maio 2023.

ALECRIM, Emerson. Brasil se destaca no GitHub com mais de 3 milhões de desenvolvedores ativos. 09 nov. 2022. **Tecnoblog**. Brasil 2022. Disponível em: <https://tecnoblog.net/noticias/2022/11/09/brasil-se-destaca-no-github-com-mais-de-3-milhoes-de-desenvolvedores-ativos/>. Acesso em: 18 maio 2023.

ALURA. **Preparando o ambiente: instalando o Xcode**. Brasil: [s.n.], 2020. Disponível em: <https://cursos.alura.com.br/course/swift-entendendo-linguagem/task/118477>. Acesso em: 18 maio 2023.

ALURA. **Preparando o ambiente: instalando e configurando o Flutter**. Brasil: [s.n.], 2020. Disponível em: <https://cursos.alura.com.br/course/flutter-widgets-stateless-stateful-imagens-animacoes/task/108005>. Acesso em: 18 maio 2023.

ALURA. **O que é Git e GitHub: como configurar e primeiros passos**. Brasil: [s.n.], 2023. Disponível em: <https://www.alura.com.br/artigos/o-que-e-git-github>. Acesso em: 18 maio 2023.

ANDRADE, Maria Margarida de. **Introdução à metodologia do trabalho científico**. São Paulo: Atlas, 1999.

ANDROID. Desenvolver apps Android com o Kotlin. **Developer Android**. [S.l.], 2023. Disponível em: [https://developer.android.com/kotlin?gclid=Cj0KCQjwmZejBhC\\_ARIsAGhCqnfeyYH4unwoAAQZhC1OwsSoJnShA2t2bPdJd5Hrx6Y3uGkLm1J-lZbAaAq2QEALw\\_wcB&gclid=aw.ds&hl=pt-br](https://developer.android.com/kotlin?gclid=Cj0KCQjwmZejBhC_ARIsAGhCqnfeyYH4unwoAAQZhC1OwsSoJnShA2t2bPdJd5Hrx6Y3uGkLm1J-lZbAaAq2QEALw_wcB&gclid=aw.ds&hl=pt-br). Acesso em: 18 maio 2023.

ASP, F. **A comparison of ionic 2 versus react native and android in terms of performance, by comparing the performance of applications**. [S.l.: s.n.], 2018.

BESSA, Hugo. Brasil está entre os dez países com mais smartphones ativados, segundo relatório da Flurry. **Mac Magazine**. São Paulo, 2012. Disponível em: <https://macmagazine.com.br/post/2012/08/27/brasil-esta-entre-os-10-de-paises-com-mais-smartphones-ativados-segundo-relatorio-da-flurry/>. Acesso em: 19 maio 2023.

BRASSCOM. **Relatório Setorial de TIC**. Brasil: [s.n.], 2022. Disponível em: <https://brasscom.org.br/inteligencia/relatorio-setorial/>. Acesso em: 19 maio 2023.

DATAAI, Analytics. **Sobre a plataforma data.ai**. [S.l.: s.n.], 2023. Disponível em:

<https://www.data.ai/pt/product/platform/>. Acesso em: 18 maio 2023.

DATAAI, Analytics. **State of Mobile 2022: Brazil, 2022**. [S.l.: s.n.], 2022. Disponível em: <https://www.data.ai/en/go/state-of-mobile-2022-brazil/>. Acesso em: 18 maio 2023.

DEITEL, P.; DEITEL, H. **Java: como programar**. 10. ed. São Paulo: Pearson Education do Brasil, 2017.

DEVMEDIA. **Conheça o iOS 8 para desenvolvedores**. [s.n.], Brasil, 2020. Disponível em: <https://www.devmedia.com.br/conheca-o-ios-8-para-desenvolvedores/32351>. Acesso em: 18 maio 2023.

DORFER, Thomas; DEMETZ, Lukas; HUBER, Stefan. Impact of mobile cross-platform development on CPU, memory and battery of mobile devices when using common mobile app features. **Procedia Computer Science**, [s. l.], v. 175, n. 2019, p. 189–196, 2020. Disponível em: <https://doi.org/10.1016/j.procs.2020.07.029>. Acesso em: 19 maio 2023.

EL-KASSAS, W. S. et al. Taxonomy of Cross-Platform Mobile Applications Development Approaches. **Ain Shams Engineering Journal**, 2015. ISSN 2090-4479. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2090447915001276>. Acesso em: 19 maio 2023;

ÉPOCA. Como muitos consumidores têm dispensado as agências bancárias. Guilherme Felitti – **Portal Revista Época**. Brasil, 2015. Disponível em: <https://epocanegocios.globo.com/Informacao/Visao/noticia/2015/06/como-muitos-consumidores-tem-dispensado-agencias-bancarias.html>. Acesso em: 18 maio 2023.

EXAME. Glassdoor compra Love Mondays, site de avaliação de empresas. **Portal Revista Exame**. Brasil, 2016. Disponível em: <https://exame.com/negocios/glassdoor-compra-love-mondays-site-de-avaliacao-de-empresas/>. Acesso em: 18 maio 2023.

FRANSSON, R.; DRIAGUINE, A. **Comparing progressive web applications with native android applications**: an evaluation of performance when it comes to response time. [S.l.: s.n.], 2017. p. 1-58.

FREIRE. Alexandre. Por que nós achamos que Flutter vai nos ajudar a escalar o desenvolvimento mobile no Nubank. **Medium**. Brasil, 2019. Disponível em: <https://medium.com/flutter-comunidade-br/porqu%C3%AA-n%C3%B3s-achamos-que-flutter-vai-nos-ajudar-a-escalar-o-desenvolvimento-mobile-no-nubank-95d07b4554d7> . Acesso em: 18 maio 2023.

FOLDOC. Application Program Interface. **Foldoc Org**. 2015. Disponível em: <https://foldoc.org/Application+Program+Interface>. Acesso em: 19 maio 2023.

FONSECA, J. J. S. **Metodologia da pesquisa científica**. Fortaleza: UEC, 2002.

GARTNER. Gartner Says Worldwide Sales of Smartphones Recorded First Ever Decline During the Fourth Quarter of 2017. **Gartner**. UK, 2018. Disponível em: <https://www.gartner.com/en/newsroom/press-releases/2018-02-22-gartner-says-worldwide-sales-of-smartphones-recorded-first-ever-decline-during-the-fourth-quarter-of-2017>. Acesso

em: 19 maio 2023.

GITHUB. **State of the Octoverse GitHub**: The top Programming languages. . [S.l.: s.n.], 2022. Disponível em: <https://octoverse.github.com/2022/top-programming-languages>. Acesso em: 18 maio 2023.

GITHUB. **Por que o Java ainda é uma linguagem campeã?**. Mike Melanson . [S.l.: s.n.], 2022. Disponível em: <https://www.linkedin.com/pulse/por-que-o-java-ainda-%C3%A9-uma-linguagem-campe%C3%A3-github-brazil/?originalSubdomain=pt>. Acesso em: 18 maio 2023.

GLASSDOOR. **Glassdoor página inicial**. [s.n.], Brasil, 2023. Disponível em: <https://www.glassdoor.com.br/index.htm>. Acesso em: 18 maio 2023.

GOLDEN. **Data.ai**. [S.l.: s.n.], 2023. Disponível em: <https://golden.com/wiki/data.ai-P95BRD>. Acesso em: 18 maio 2023.

GOLDEN. **Glassdoor**. [S.l.: s.n.], 2023. Disponível em: <https://golden.com/wiki/Glassdoor-YK4934>. Acesso em: 18 maio 2023.

HABCHI, S. et al. Code Smells in iOS Apps: How Do They Compare to Android? International Conference on Mobile Software Engineering and Systems (MOBILESoft). **IEEE Xplore**. [S.l.], 2017. Disponível em: <https://ieeexplore.ieee.org/document/7972725/>. Acesso em: 19 maio 2023.

HANSSON, N.; VIDHALL, T. **Effects on performance and usability for cross-platform application development using react native**. [S.l.: s.n.], 2016.

HEITKÖTTER, H.; HANSCHKE, S.; MAJCHRZAK, T. A. Evaluating cross-platform development approaches for mobile applications. In: . Web Information Systems and Technologies: 8th International Conference, WEBIST 2012, Porto, Portugal, April 18-21, 2012, **Revised Selected Papers**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 120–138. ISBN 978-3-642-36608-6. Disponível em: [https://dx.doi.org/10.1007/978-3-642-36608-6\\_8](https://dx.doi.org/10.1007/978-3-642-36608-6_8). Acesso em: 19 maio 2023.

INFOWORD. **What is Kotlin?** The Java alternative explained. [s.n.],. USA, 2022. Disponível em: <https://www.infoworld.com/article/3224868/what-is-kotlin-the-java-alternative-explained.html>. Acesso em: 18 maio 2023.

KUMAR, P. **Analysis of Native and Cross-Platform Methods for Mobile Application Development**. [S.l.], 2014. Tavant Technologies. White Paper, p. 1-9.

LAKATOS, Eva Maria; MARCONI, Marina de Andrade. **Metodologia científica**. 2. ed. São Paulo: Atlas, 1991.

LIMA, Fernando F. de. **Avaliação de Frameworks para o Desenvolvimento de Aplicações Híbridas**. 2019. 119 f. Trabalho de Conclusão de Curso (Bacharel em Engenharia de Software) – Universidade Federal do Pampa. Alegrete, 2019. Disponível em: <https://dspace.unipampa.edu.br/bitstream/riu/4224/1/Fernando%20Fortunato%20de%20Lima%20-%202019.pdf>. Acesso em: 19 maio 2023.

LINKEDIN. **Sobre o LinkedIn**. [S.l.: s.n.], 2023. Disponível em: <https://about.linkedin.com/pt-br#:~:text=O%20LinkedIn%20come%C3%A7ou%20na%20sala,e%20de%20solu%C3%A7%C3%B5es%20de%20recrutamento>. Acesso em: 18 maio 2023.

LINKEDIN. **LinkedIn página inicial**. [S.l.: s.n.], 2023. Disponível em: <https://www.linkedin.com/>. Acesso em: 18 maio 2023.

MACEDO, Tobias. **Desenvolvimento de aplicativo móvel para sistema operacional iOS em linguagem Swift para leitura de consumo de tomada inteligente em tempo real**. 2017. 110 f. Monografia (Graduação em Engenharia Elétrica) – Universidade Federal do Ceará, Fortaleza, 2017. Disponível em: [https://repositorio.ufc.br/bitstream/riufc/35049/1/2017\\_tcc\\_tv MacedoJunior.pdf](https://repositorio.ufc.br/bitstream/riufc/35049/1/2017_tcc_tv MacedoJunior.pdf). Acesso em: 19 maio 2023.

MAGALHÃES, Ícaro L. **Um estudo comparativo entre padrões arquiteturais para o desenvolvimento de aplicativos para a plataforma ios**. 2018. 51 f. Monografia (Bacharel em Ciências da Computação) – Centro de Informática, Universidade Federal da Paraíba, João Pessoa, 2018. Disponível em: <https://repositorio.ufpb.br/jspui/bitstream/123456789/15651/1/ILM12122018.pdf>. Acesso em: 19 maio 2023.

MELO, Diego. O que é Stack Overflow?. 2021. **Tecnoblog**. Disponível em: <https://tecnoblog.net/responde/o-que-e-stack-overflow/>. Acesso em: 18 maio 2023.

MENDONÇA, Vinicius R. L. de; BITTAR, Thiago; DIAS, Marcio. Um estudo dos Sistemas Operacionais Android e iOS para o desenvolvimento de aplicativos. **EnaComp**, São Carlos, 2011. Disponível em: [https://www.enacomp.com.br/2011/anais/trabalhos-aprovados/pdf/enacomp2011\\_submission\\_54.pdf](https://www.enacomp.com.br/2011/anais/trabalhos-aprovados/pdf/enacomp2011_submission_54.pdf). Acesso em: 19 maio 2023.

MISHRA, R. **Progressive webapp**: Review. International Research Journal of Engineering and Technology. V. 3, n. 6, p. 2–3, [S.1], 2016.

NASCIMENTO, F. P.; SOUSA, F. L. L. **Metodologia da pesquisa científica**: teoria e prática. Brasília: Thesaurus, 2015.

NEVES, Jonathan; MENDES JUNIOR, Vilmar. Uma análise comparativa entre Flutter e react native como frameworks para desenvolvimento híbrido de aplicativos mobile: estudo de caso visando produtividade. **Anima Educação**, Tubarão, 2020. Disponível em: <https://repositorio.animaeducacao.com.br/bitstream/ANIMA/15960/1/Artigo%20final.pdf>. Acesso em: 19 maio 2023.

NUNES, Vinicius G. C. ; QUEIROZ, Paulo G. G. Avaliação de abordagens e definição de diretrizes para o desenvolvimento de aplicativos mobile: um relato de experiência utilizando a abordagem não nativa de desenvolvimento. **Repositório Universidade Federal Rural do Semiárido**, Mossoró, 2021. Disponível em: [https://repositorio.ufersa.edu.br/bitstream/prefix/6917/1/ViniciusGCN\\_ART.pdf](https://repositorio.ufersa.edu.br/bitstream/prefix/6917/1/ViniciusGCN_ART.pdf). Acesso em: 19 maio 2023.

OCCHINO, T. **React Native**: Bringing modern web techniques to mobile. . [S.l.: s.n.], 2015.

Disponível em: . Acesso em: 19-11-2018.

OVERFLOW. **Stack Overflow Developer Survey 2022**. [S.l.: s.n.], 2022. Disponível em: <https://survey.stackoverflow.co/2022/>. Acesso em: 18 maio 2023.

PALUDO, Lauriana. **Um estudo sobre as tecnologias Java de desenvolvimento de aplicações móveis**. 2003. 118 f. Trabalho de Conclusão de Curso (Especialização em Ciência da Computação) – Universidade Federal de Santa Catarina, Florianópolis, 2003. Disponível em: [https://www.academia.edu/672665/Um\\_Estudo\\_Sobre\\_as\\_Tecnologias\\_JAVA\\_de\\_Develop\\_vimento\\_de\\_Aplica%C3%A7%C3%B5es\\_M%C3%B3veis](https://www.academia.edu/672665/Um_Estudo_Sobre_as_Tecnologias_JAVA_de_Develop_vimento_de_Aplica%C3%A7%C3%B5es_M%C3%B3veis). Acesso em: 19 maio 2023.

PEDRO, Wagner. O que é e para que serve um SDK?. 2022. **Tecnoblog**. Disponível em: <https://tecnoblog.net/responde/o-que-e-e-para-que-serve-um-sdk/#:~:text=SDK%2C%20ou%20kit%20de%20desenvolvimento,aplicativos%20em%20uma%20determinada%20plataforma>. Acesso em: 19 maio 2023.

REDHAT. **O que é open source?** [S.l.: s.n.], 2019. Disponível em: <https://www.redhat.com/pt-br/topics/open-source/what-is-open-source>>. Acesso em: 18 maio 2023.

REIS, Antônio C. S. dos. **Um estudo comparativo entre modelos de desenvolvimento de aplicações móveis**. 2019. 39 f. Monografia (Bacharel em Engenharia de Software) – Universidade Federal do Ceará, Quixadá, 2019. Disponível em: <[https://repositorio.ufc.br/bitstream/riufc/49707/1/2019\\_tcc\\_acsdosreis.pdf](https://repositorio.ufc.br/bitstream/riufc/49707/1/2019_tcc_acsdosreis.pdf)>. Acesso em: 19 maio 2023.

REIS. Lucas B. R. DOS. **Projeto de Trabalho de Conclusão de Curso: desenvolvimento de um aplicativo móvel para zeladoria urbana**. 2021. 52 f. Trabalho de Conclusão de Curso (Tecnólogo em Gestão de Tecnologia da Informação) – Instituto Federal de Santa Catarina, Florianópolis, 2021. Disponível em: [https://repositorio.ifsc.edu.br/bitstream/handle/123456789/2316/TCC\\_LucasBaggio\\_IFSC\\_asinado\\_assinado\\_assinado%20\(1\).pdf?sequence=1](https://repositorio.ifsc.edu.br/bitstream/handle/123456789/2316/TCC_LucasBaggio_IFSC_asinado_assinado_assinado%20(1).pdf?sequence=1). Acesso em: 19 maio 2023.

SANTOS, Bruno M A. **Revisão e Reestruturação da Arquitetura da Plataforma Vortal Next**. Vortal Software Development Kit. 2013. 108 p. Dissertação (Mestrado de Engenharia em Informática ) - Universidade do Minho Escola de Engenharia, Azurém, 2013. Disponível em: [https://repositorium.sdum.uminho.pt/bitstream/1822/27908/1/eeum\\_di\\_dissertacao\\_pg19838.pdf](https://repositorium.sdum.uminho.pt/bitstream/1822/27908/1/eeum_di_dissertacao_pg19838.pdf). Acesso em: 19 maio 2023.

SILVA, Felipe; SALDANHA, Mário. Objective-C. **UBIQ - Universidade Federal de Pelotas**, Pelotas, 2012. Disponível em: <http://ubiq.inf.ufpel.edu.br/clp/doku.php?id=objective-c-1-trabalho>. Acesso em: 19 maio 2023.

SILVA, Fransérgio A. de S.; PRADO Ely F. do. Análise teórica sobre o desenvolvimento de aplicativos nativos, híbridos e webapps. **Revista EduFatec: educação, tecnologia e gestão**. Franca. v. 2 n.1, 2019. Disponível em: <http://ric-cps.eastus2.cloudapp.azure.com/bitstream/123456789/5044/1/AN%c3%81LISE%20TE%c3%93RICA%20SOBRE%20O%20DESENVOLVIMENTO%20DE%20janeiro%20junho%2020>

19.pdf. Acesso em: 19 maio 2023.

TEAMGUIDE. **Bom Score no Glassdoor**. [S.l.: s.n.], 2021. Disponível em: <https://teamguide.app/blog/o-que-e-glassdoor/>. Acesso em: 18 maio 2023.

TECHTUDO. **Aplicativo de emprego: Glassdoor reúne vagas de trabalho**. [S.l.: s.n.], 2012. Disponível em: <<https://www.techtudo.com.br/tudo-sobre/glassdoor/>>. Acesso em: 18 maio 2023.

TOLEDO, Lucas H. B. **Desenvolvimento de aplicações Android adaptáveis utilizando a linguagem Kotlin**. 2019. 34 f. Monografia (Bacharel em Ciências da Computação) - Instituto de Ciências Exatas Bacharelado em Ciência da Computação, Universidade Federal de Ouro Preto, Ouro Preto, 2019. Disponível em: [https://monografias.ufop.br/bitstream/35400000/3042/6/MONOGRRAFIA\\_DesenvolvimentoAplica%c3%a7%c3%b5esAndroid.pdf](https://monografias.ufop.br/bitstream/35400000/3042/6/MONOGRRAFIA_DesenvolvimentoAplica%c3%a7%c3%b5esAndroid.pdf). Acesso em: 19 maio 2023.

VENTUREBEAT. **Apple announces 'Swift,' a new programming language for OS X & iOS** - Harrison Weber. [S.l.: s.n.], 2014. Disponível em: <https://venturebeat.com/business/apple-introduces-a-new-programming-language-swift-objective-c-without-the-c/>. Acesso em: 18 maio 2023.

VIJAYAJUMAR. **Handling Offline Capability And Data Sync In Native Mobile Apps**. . [S.l.: s.n.], 2017. Disponível em: <https://www.ideas2it.com/blog/offline-sync-native-apps/>. Acesso em: 17 maio 2018.

WIKIPÉDIA. **Airbnb**. [S.l.: s.n.], 2017. Disponível em: <https://pt.wikipedia.org/wiki/Airbnb>. Acesso em: 18 maio 2023.

WINDMILL, Eric. **Flutter in Action**. 1. ed. New York, Manning Publications, 2020. 310 p.

YATES, II, M. Practical investigations of digital forensics tools for mobile devices. **Information Security Curriculum Development Conference**. New York, NY, USA. ACM. InfoSecCD, 2010, p 156–162. Disponível em: <http://doi.acm.org/10.1145/1940941.1940972>. Acesso em: 18 maio 2023.

ZANELLA, Cleunice. **Metodologia de estudo de pesquisa em administração**. Fascículo de Metodologia Científica, UFSC/MEC/CAPES/PNAP, Florianópolis, 2009.