



Volume 4, número 1, ano 2021
REVISTA DE TECNOLOGIA INVEST

Artigo 7

LOJA DE SERVIÇOS: APLICATIVO ANDROID PARA LOCALIZAR PRESTADORES DE SERVIÇOS

Deivid de Almeida Padilha da Silva¹
Agnaldo José Proença Moraes

RESUMO: O artigo apresenta o processo de desenvolvimento de um aplicativo *mobile*, nativo para *Android*, que visa auxiliar pessoas que necessitem contratar um profissional para solucionar determinado problema, seja no ramo de construção, hidráulica, carpintaria e etc., sem que o mesmo necessite se deslocar. Visando tornar mais simples e eficiente a contratação de prestadores. A aplicação permite aos prestadores de serviços, autônomos ou empresas, se cadastrarem e divulgarem seus trabalhos e permitir que outros usuários os encontrem com mais facilidade em sua região. Através do GPS o aplicativo indicará os mais variados prestadores na região com suas respectivas informações, porém também será possível pesquisar por um serviço. Por questão de confiança, os usuários deverão efetuar um cadastro para acessar o aplicativo, no qual em que em um primeiro momento será realizado no próprio aplicativo, com as informações básicas do usuário. O Desenvolvimento do aplicativo procura apresentar um aplicativo simples e direto de forma totalmente gratuita que atenda às necessidades dos usuários, para tal será observado o aplicativo GetNinjas, que atualmente está disponível para download *Google Play*, e o artigo referente ao desenvolvimento do aplicativo *Yellow List*. O aplicativo será desenvolvido utilizando a linguagem de programação *Java*, e o IDE *Android Studio* com integração ao *Firebase*, para armazenar os dados em nuvem.

¹ Possui graduação em tecnologia em redes de computadores, pelo IFMT- Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso (2013). É especialista em Docência no Ensino Superior pela FTED (2014) e em Educação de Jovens e Adultos (E.J.A) pela FLC (2016). Possui Mestrado em Ciência da Computação pela UNIFACCAMP (2018). Tem experiência em Ciência da Computação com ênfase na área de Redes de Computadores e Telecomunicações, Reconhecimento de Padrões, Processamento de Imagens e Arquitetura de Computadores. Atualmente é professor da faculdade Invest.

Palavras-chave: Aplicativo Android. Prestação de Serviço. Geolocalização. Firebase.

ABSTRACT: The article presents the development process of a mobile application, native to Android, which aims to help people who need to hire a professional to solve a certain problem, be it in construction, hydraulics, carpentry, etc., without having to travel. Aiming to make the hiring of providers simpler and more efficient. The application allows service providers, freelancers or companies, to register and publicize their works and allow other users to find them more easily in their region. Through GPS the application will indicate the most varied providers in the region with their respective information, but it will also be possible to search for a service. For reasons of trust, users must register to access the application, which will first be done in the application itself, with the user's basic information. The development of the application seeks to present a simple and direct application that meets the needs of users completely free of charge, for this will be observed the GetNinjas application, which is currently available for download Google Play, and the article on the development of the Yellow List application. The application will be developed using the Java programming language, and the Android Studio IDE with integration to Firebase, to store data in the cloud.

Keywords: Android application. Service provision. Geolocation. Firebase.

Introdução

Atualmente a utilização de dispositivos eletrônicos tem se tornado cada vez mais comum para realizar compras e acessar outras aplicações. Através dos meios de comunicação é possível ter acesso a várias informações sobre determinado produto sem a necessidade de locomoção ou de uma busca exaustiva.

Um dos maiores meios de comunicação utilizados pela sociedade atualmente são os *smartphones*, ou celulares, que a muito tempo deixaram de ser apenas uma grande caixa para realizar ligações e trocar mensagens, e se tornou um dispositivo complexo que permite executar diversas funções como navegar na internet, mídias sociais, compras e muito mais. Uma pesquisa realizada pelo *Paypal* em 2019, revela que cerca de 92% da população brasileira realiza compras online e 66% desses compradores navegam pela internet através do seu *smartphone* procurando por serviços ou produtos que são mais recomendados e oferecem os melhores preços.

Quando se trata de prestadores de serviço é muito incomum ouvir uma recomendação de aplicativo ou site, existem algumas aplicações no ramo de prestação de serviço, porém, geralmente atende apenas uma área de atuação, são pagos ou não apresentam o que o usuário realmente necessita.

Há poucas opções de localizações para situação do cotidiano como eletricitas, encanadores, entre outros, muitas vezes as informações sobre esses prestadores estão desatualizadas ou são apresentadas de forma confusa. Em algum momento nos deparamos com necessidade da realização de algum tipo de serviço como hidráulico ou elétrico, encontrar algum prestador de serviço que faça essa tarefa com preço combinado é uma solução.

Este projeto concentra-se especialmente em apresentar uma aplicação *mobile* que possa auxiliar na situação descrita anteriormente. De modo geral, os objetivos deste trabalho podem ser elencados da seguinte maneira: (i) Apresentar uma ferramenta móvel para auxiliar as

peçoas a encontrarem prestadores de serviços e realizarem contratos para resolução de seus problemas de imediato, bem como apresentar os conceitos sobre as tecnologias que permitem a implementação do mesmo. (ii) realizar uma análise das interfaces utilizadas nos *app* baseado em princípios e heurísticas em IHC.

A construção do artigo foi organizada em seis sessões (6). Nesta primeira seção foi apresentada uma breve introdução ao tema central da pesquisa, suas motivações e objetivos. Posteriormente na seção 2 será apresentado os métodos, ferramentas e meios utilizados no desenvolvimento do aplicativo. Na seção 3 falaremos brevemente sobre outras aplicações similares e suas diferenças. Seguindo, temos a seção 4, no qual será abordado a aplicação desenvolvida. Encerrando, temos a seção 5 que será discutido os resultados alcançados neste projeto e a seção 6 que serão feitas as considerações finais.

2. Fundamentação teórica

Nesta seção serão abordados métodos, ferramentas e meios utilizados no processo de desenvolvimento do aplicativo.

2.1 Java e Orientação a Objetos

"Computadores são inúteis, eles apenas dão respostas" – Picasso.

Antes de falar sobre o desenvolvimento, devemos falar brevemente sobre a linguagem de programação a ser utilizada no processo de desenvolvimento, o *Java*. Uma linguagem de programação nada mais é do que uma série de instruções que são passadas para o computador a fim de produzir algum tipo de dado. Podendo ser utilizadas para desenvolver programas específicos que controlam o comportamento lógico e físico de um dispositivo.

A linguagem de programação *Java*, originalmente desenvolvida pela *Sun Microsystems* que posteriormente seria adquirida pela até hoje *Oracle*, além de ser gratuito e de código-fonte aberto, é orientada a objeto e multiplataforma, possuindo um diversificado acervo de bibliotecas e classes que facilitam o processo de desenvolvimento de programas, e a torna excelente para desenvolvimento *Android*, levando-se em conta o fato do *Android* também seguir a proposta de código aberto.

Mesmo com suas quedas ao longo dos anos, a linguagem *Java* ainda é a mais utilizada por muitos programadores ao redor do mundo. Embora existam muitas linguagens de programação que podem ser utilizadas no desenvolvimento de aplicativos para *Android*, *Java* é umas das linguagens utilizadas para criação de aplicações *Android* e é recomendada pela própria *Google* para utilização e desenvolvimento de aplicativos de sua plataforma, apresentando diversas soluções de problemas e possuindo uma comunidade enorme que está sempre auxiliando os desenvolvedores a encontrarem soluções sobre determinado problema.

2.1 Android

Os dispositivos móveis já são um objeto comum na vida das pessoas:

Nos dias de hoje, ninguém consegue ficar longe de um celular, seja para mandar um email, tirar uma foto, assistir um vídeo, conversar com os amigos, navegar na internet, acompanhar as redes sociais etc. Portanto, os smartphones

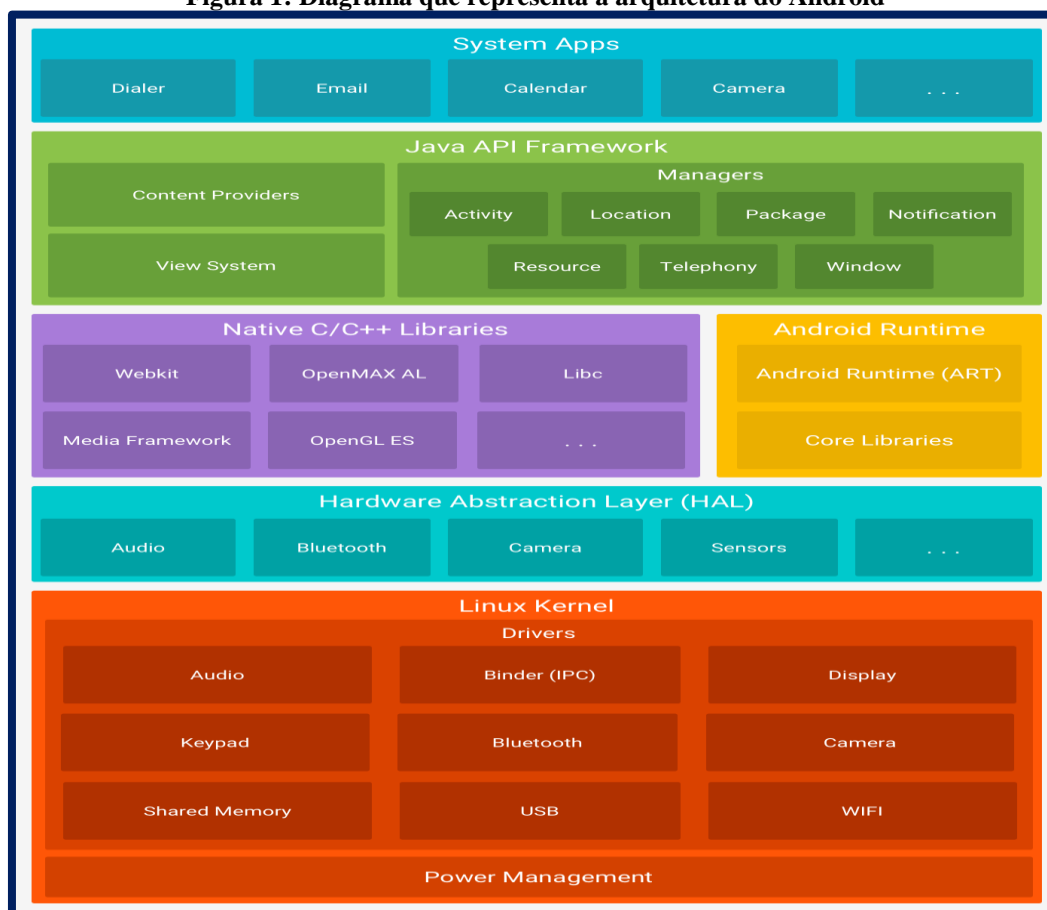
e tablets atualmente são objetos praticamente inseparáveis da maioria das pessoas.(LECHETA. 2015, p.25)

O sistema *Android* não faz parte apenas dos *smartphones* que muitas pessoas utilizam diariamente, a muito tempo o *Android* deixou de fazer parte apenas dos celulares e atingiu outros dispositivos, como TVs, relógios, carros e até mesmo óculos tecnológicos. Aplicativos desenvolvidos para dispositivos móveis não abordam apenas interesses da sociedade, mesmo empresas aderem a utilização de dispositivos móveis para agilização de processos dentro da empresa ou como uma ferramenta de comunicação mais ágil fora da empresa.

A evolução do sistema *Android* ao longo dos anos já nos apresenta funcionalidades extraordinárias, design de interface intuitivo e de fácil acesso que a algumas décadas atrás ninguém imaginaria que isso seria possível, as funcionalidades evoluíram ao ponto de não se limitar apenas às funções básicas que um celular apresentava, nós permitindo navegar através de navegadores *web*, realizar ligações utilizando a internet como meio de comunicação, até mesmo viajar ao redor do mundo sem a necessidade de locomoção, apenas usufruindo da tecnologia do *GPS* inclusa nos dispositivos.

O sistema operacional *Android* foi construído baseado no *kernel* do *Linux*, o *kernel Linux* é o responsável por fazer a comunicação entre o *software* e o *hardware* do computador, e gerenciar a *CPU* e memória, por exemplo. Apesar do sistema *Android* ser construído baseado no *kernel* do *Linux*, ele possui diversas modificações específicas que estão presentes apenas em sistemas mobile.

Figura 1: Diagrama que representa a arquitetura do Android



Fonte: <http://developer.android.com/guide/platform?hl=pt-br>.

2.2 Android Studio

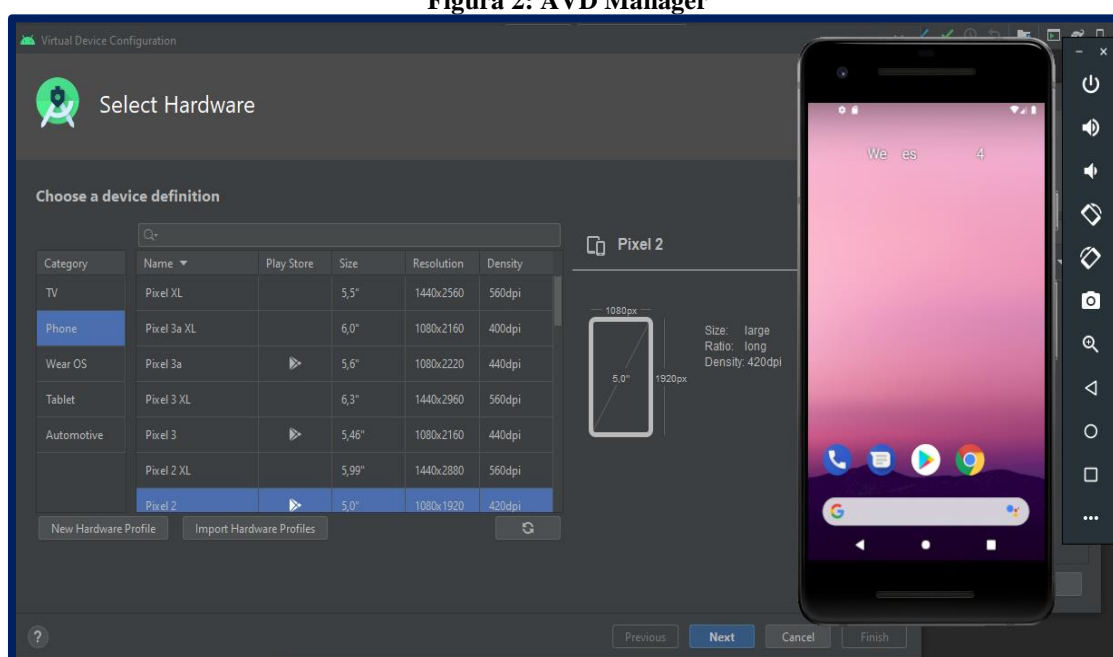
No processo de desenvolvimento de um software são necessários ferramentas e bibliotecas que atendam a necessidade e objetivo de um software a ser desenvolvido, para isso, é utilizado um ambiente de desenvolvimento ou IDE, sua sigla em inglês, uma IDE pode variar de um simples editor de código a um sistema complexo que fornece diversas funcionalidades para os desenvolvedores que auxiliam no processo de desenvolvimento.

O *Android Studio*, é a IDE oficial de desenvolvimento *Android*. Anunciado no *Google I/O 2013* e baseado no *IntelliJ IDEA* da *JetBrains* (LECHETA, 2015), disponibilizando as mais diversas e recentes SDK's para auxiliar no desenvolvimento e está em constantes atualizações para apresentar novas ferramentas dentro do ambiente.

Através do *Android Studio* é possível desenvolver aplicações não só para *smartphones*, como também pode ser desenvolvido aplicações para tablets, relógios (*Wear OS*), TV (*Google TV*), carros (*Google Auto*) e *Android Things*. Com um editor de código inteligente, fornece suporte para as linguagens *Kotlin*, *Java* e *C/C++*, com verificação de erros de sintaxe e depuração da aplicação para verificar erros na execução e muitas outras funcionalidades específicas.

Para possibilitar a realização de testes durante o desenvolvimento o *Android Studio* oferece o *AVD Manager*, dispositivo virtual *Android*, que possibilita simular um dispositivo com sistema *Android* na máquina, onde é possível simular desde a versão 2.3.3 do *Android* até as versões mais recentes no mercado, inibindo a utilização de um dispositivo físico no processo de desenvolvimento. Porém o *Android Studio* também possibilita a conexão com um *smartphone* para fins de teste da aplicação em um dispositivo real.

Figura 2: AVD Manager



Fonte: Captura do autor em outubro de 2020.

2.3 Banco de dados e Web Services

Em grandes projetos e aplicações que trafegam dados, é necessário a utilização de banco de dados para armazenar informação. Grandes empresas no mercado possuem servidores locais para armazenar os dados de seus clientes, porém um dos grandes serviços da atualidade são os servidores em nuvem para armazenar dados.

O Android também possibilita a integração com o banco de dados *SQLite*. O *SQLite* é um banco de dados que permite armazenar algumas informações básicas no dispositivo, sem que seja necessário se conectar a um banco de dados de grande porte, como por exemplo a configurações e informações básicas do aplicativo.

Porém para manipulação de grande quantidade de informações e transações é necessário um banco de dados de grande porte. Em um servidor remoto o processo de acessar e enviar dados para um servidor é bem simples, definindo mensagens e regras dentro da aplicação, é possível enviar e recuperar informações a qualquer momento, desde que esteja conectado à internet, não se limitando a uma linguagem de programação ou aplicação mobile, *web* ou *desktop*, porém cada linguagem de programação e banco de dados possui suas sintaxes próprias, sendo necessário consultar suas documentações.

Para uma aplicação acessar o banco de dados em nuvem de forma segura e padronizada no servidor, é necessário a utilização de *web service* independente da linguagem, onde sua função é acessar o banco de dados e retornar os dados no formato *XML* ou *JSON* para o *mobile*/cliente ler os dados (LECHETA, 2017).

Um dos métodos mais utilizados para comunicar uma aplicação com o servidor é a *web service* do tipo *REST*, criado sobre o protocolo *HTTP* e utiliza os métodos *GET*, *DELETE*, *POST* e *PUT*, que são métodos de acesso. *GET* permite recuperar informações, *DELETE* permite excluir, e *POST* permite incluir dados. No *REST* o retorno mais comum é o *JSON* (LECHETA, 2015).

Para tornar mais simples a criação de uma *web service*, o *firebase* automaticamente cria um arquivo *json*, que quando integrado e configurando dentro da aplicação, realiza todo o processo de comunicação com o banco de dados e posteriormente devolve a informação que for solicitada pela aplicação.

2.3.1 Firebase

Pertencente à *Google*, o *Firebase* é um servidor de dados em nuvem que possui suporte para *Android*, *iOS*, *web*, *C++* e também a plataforma *Unity*, ambiente de desenvolvimento de jogos.

Sobre a utilização dos servidores nas aplicações Android, podemos definir que:

Os servidores utilizados para aplicativos *Android* são *Oracle SQL*, *Microsoft SQL Server*, e *MySQL* que estão conectados ao servidor com arquivos *PHP*. Então a *Firebase* passou a existir para Aplicativos *Android* que utiliza *JSON* para armazenamento de dados. Os outros servidores usam um formato de tabela

(linhas e colunas) para armazenamento de dados. (KHAWAS, SHAH, 2018, p.49, tradução nossa)

Através do *Firebase* é possível criar o banco de dados de um aplicativo rapidamente, possibilitando construir, testar e monitorar o aplicativo. O *Firebase* é uma ferramenta bem completa para desenvolvimento, dentre as suas funcionalidades estão serviços de análise, que fornece informações sobre a utilização das aplicações, serviço de autenticação, que permite integrar o login ou cadastro do usuário através do *Facebook*, *Google* ou *Twitter*, por exemplo. Além de fornecer um banco de dados em tempo real, que está sempre devolvendo os dados atualizados a aplicação, e um banco de dados para armazenamento de mídias, por exemplo fotos e vídeos, também possui integração com *Google Cloud Platform* para grandes aplicações que possuem muitas informações.

No site oficial do *Firebase* é possível acessar uma documentação abrangente com vários idiomas para leitura, incluindo o português brasileiro, apresentado os passos a passo de integração com projetos em cada linguagem que possui suporte, através de exemplos bem didáticos, e muitos outros tutoriais para facilitar o desenvolvimento do projeto com o banco de dados.

O *Firebase* possui dois planos disponíveis no momento para sua utilização, o primeiro é denominado plano *Spark*, é um plano gratuito, extremamente útil para *apps* que não necessitam de muito armazenamento de dados e para realizar testes antes do lançamento de algum *app* grande. O segundo plano, denominada plano *Blaze*, possui a dinâmica de pague por utilização, possibilitando aumentar o tamanho do banco de dados através do *Google Cloud Platform*, onde o proprietário irá pagar apenas o que foi e está sendo utilizado pela *app*, diferente de um valor fixo, e possibilita realizar o backup dos dados sempre que desejar.

3. Aplicações Similares

Nesta seção será apresentado outras aplicações similares e suas diferenças e realizar uma análise comparativa entre o aplicativo proposto e as soluções mencionadas.

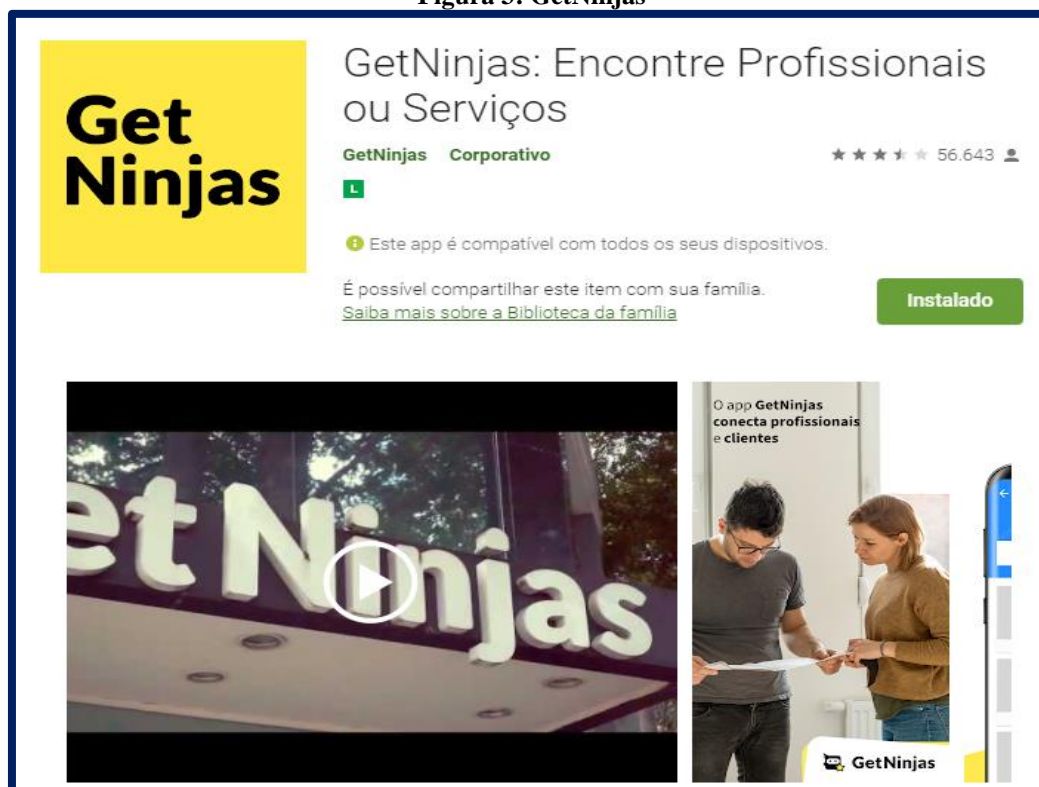
No atual mercado de aplicativos de prestação de serviço, é possível encontrar diversos *apps* que apresentam as funcionalidades de localizar um prestador de serviço, neste tópico será apresentado o mais relevante e que possuem maior utilização de usuários e também um projeto de aplicação que possui o mesmo objetivo.

3.1 GetNinjas

Com mais de 1 milhão de downloads e mais de 50 mil avaliações, o GetNinjas é o mais popular *app* para localizar prestadores de serviço. Dentro do *app*, é possível ter acesso aos contratos e prestadores, tanto para usuários comuns quanto para os prestadores cadastrados.

O mesmo é um *app* que faz uso de geolocalização, o GetNinjas apresenta uma dinâmica em que o cliente informa qual serviço deseja encontrar no aplicativo e após o *app* apresenta os prestadores disponíveis, porém é necessário realizar um pagamento com algumas moedas que são compradas dentro do aplicativo, para enfim, poder ter acesso completo as funcionalidades do aplicativo. A seguir é apresentado uma imagem deste aplicativo.

Figura 3: GetNinjas



Fonte: <https://play.google.com/store/apps/details?id=br.com.getninjas.pro>.

Ao realizar uma busca no *Google Play* é possível encontrar diversas aplicações similares ao GetNinjas, porém, muitas das aplicações gratuitas apresentam bugs que se tornam desagradáveis a seus utilizadores ou não são atualizados a um longo tempo para acompanhar as expectativas dos usuários.

Outra aplicação no segmento de serviços que foi usado no estudo de desenvolvimento do aplicativo, é o Yellow List, apresentado em um artigo pela revista Prospectus, que demonstra um projeto de aplicativo que busca divulgar prestadores de serviço aos clientes. Apesar de não está disponível para download no *Google Play*, o aplicativo busca entregar o seu objetivo de forma simples, diferente da “Loja de Serviços”, o Yellow List não apresentou uma forma de cadastro para pessoas como prestador de serviços, apenas a realiza a busca de prestadores por geolocalização e apresenta os perfis e a opção de entrar em contato através do número telefônico ou e-mail.

Pensando em soluções para melhorar a experiência dos usuários em relação ao *apps* que buscam divulgar e localizar prestadores de serviço, a aplicação desenvolvida buscou apresentar uma interface e funcionalidade que outras aplicações não entregavam a seus usuários.

Conforme tabela a seguir, podemos ver uma comparação base entre o GetNinjas e a Loja de Serviços, quanto às suas similaridades e diferenças que ambas aplicações apresentam ao seus usuários.

Tabela 1 - Comparação de propostas

	GetNinjas	Loja de Serviços
Similaridades	<ul style="list-style-type: none"> - Gratuito para download. - Divulga prestadores de serviço. - Comparação de distância entre usuários. - Divulga perfil do prestador - Opção de cadastro como cliente ou prestador. 	
Diferenças	<ul style="list-style-type: none"> - Parcialmente gratuito. - Desenvolvido para Android e iOS. - Clientes publicam o serviço desejado. - Avaliação de prestador. 	<ul style="list-style-type: none"> - Totalmente gratuito. - Disponível apenas para Android. - Clientes procuram pelo serviço desejado. - Entrar em contato através do aplicativo. - Exibe prestadores diversos nas proximidades.

Fonte: Elaborada pelo autor, 2020.

4. Loja de Serviços - Desenvolvendo o aplicativo

Nesta seção serão abordadas algumas classes e arquivos que possuem maior ênfase no processo de desenvolvimento da aplicação, e será apresentado a interface de usuário que a aplicação apresenta.

As aplicações desenvolvidas no Android Studio, utilizando a linguagem *Java*, apresentam diferenças estruturais comparadas a outras plataformas. Uma aplicação *Android* apresenta separação entre os *Layouts* visuais e a programação lógica para realizar as funcionalidades desejadas.

Rodrigues, Silveira e Fuini (2019, p. 71) dividem o processo de desenvolvimento através de quatro componentes básicos, sendo eles a *Activity* que representa a tela da aplicação, o *services* que executa tarefas em segundo plano, *content* e *providers* que gerencia os dados da aplicação e por fim, *broadcast receivers* que exibe mensagens do sistema ou de outras aplicações.

É possível utilizar diversas funcionalidades para desenvolver uma aplicação, porém será apresentada neste artigo apenas as principais utilizadas no desenvolvimento do *app* “Loja de Serviços”, sem aprofundar muito nos fundamentos técnicos de desenvolvimento.

Ao iniciar uma aplicação o sistema *Android* precisa saber qual *Activity* será iniciada primeiro e definir quais outras *Activitys* existem na aplicação, para isso existe o *Android Manifest*. O *Android Manifest* “é do tipo *XML* e declara todos os componentes que fazem parte da aplicação, qual é o componente inicial a ser carregado e também todas as permissões necessárias para a sua execução” (RODRIGUES, SILVEIRA, FUINI, 2019).

O *Manifest* é um arquivo necessário em toda aplicação, que permite descrever as informações essenciais do aplicativo no momento da compilação. Dentre as informações essenciais que devem estar no *Manifest* está o local das entidades do código, os componentes do aplicativo, permissões e recursos que será exigido do *hardware* e *software*.

Figura 7: Androidmanifest.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3    package="com.ajdev.lojadeservicos">
4
5    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
6    <application
7      android:allowBackup="true"
8      android:icon="@mipmap/ic_launcher"
9      android:label="Loja de Serviços"
10     android:roundIcon="@mipmap/ic_launcher_round"
11     android:supportsRtl="true"
12     android:theme="@style/AppTheme">
13

```

Fonte: Captura do autor em outubro de 2020.

Uma das permissões utilizadas para “Loja de Serviços” funcionar corretamente, é o *ACCESS_FINE_LOCATION*, essa permissão permite que o aplicativo acesse a localização do usuário. Essa é uma das funções muito úteis do *Android*, pois ele elimina a necessidade de integração com a *API* do *Google Maps* já que o *app* não busca exibir os prestadores em ponteiros em um mapa.

Uma das principais classes do *Android* é a *Activity*, classe responsável por exibir a tela para os usuários e controlar o que deve ser executado na *Activity* que estiver apresentando (LECHETA, 2015). As *Activitys* possuem seus próprios ciclos de vida que diferem uma das outras, onde é necessário definir em cada caso de utilização, porém podem ser utilizadas para apresentar a aplicação de forma completa. Através de uma *Activity* ou *Fragment* é possível acessar telas de outras aplicações, como por exemplo a câmera ou galeria do dispositivo, muito utilizado em aplicações de mídias sociais.

Figura 8: Activity e seu ciclo de vida

```

public class Activity extends ApplicationContext {
    protected void onCreate(Bundle savedInstanceState);

    protected void onStart();

    protected void onRestart();

    protected void onResume();

    protected void onPause();

    protected void onStop();

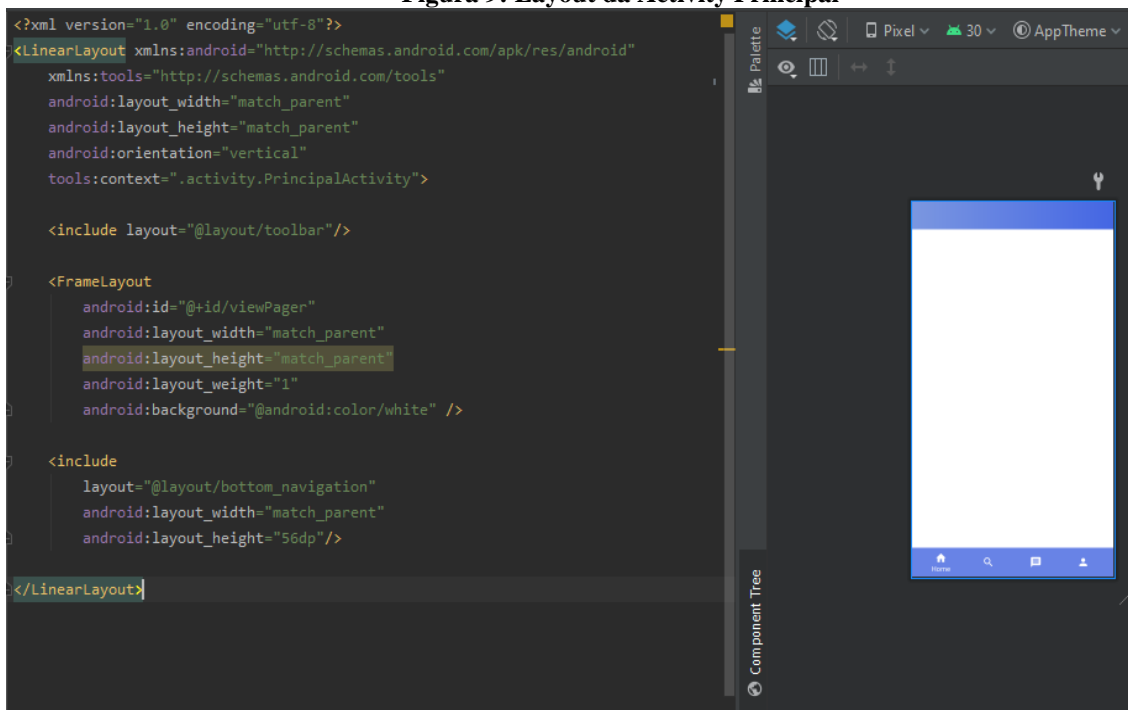
    protected void onDestroy();
}

```

Fonte: <http://developer.android.com>.

Uma *Fragment* é uma subclasse do *Android Studio* que permite navegar em várias telas quando configurada na mesma *Activity*, na aplicação isso ocorre através de um *Button Navigation* dentro da *Activity Principal*.

Figura 9: Layout da Activity Principal



Fonte: Captura do autor em outubro de 2020.

Um *Fragment* representa o comportamento ou uma parte da interface do usuário em um *FragmentActivity*, está refere-se faz parte da biblioteca de suporte. É possível combinar vários fragmentos em uma única atividade para criar uma IU de vários painéis e reutilizar um fragmento em diversas atividades.²

As buscas por um prestador no aplicativo são realizadas de acordo com as coordenadas dos usuários. Através endereço fornecido, no caso o CEP, a aplicação é capaz de retorna um endereço em latitude e longitude que fica gravado no perfil do usuário no banco de dados do *firebase*, e posteriormente, é utilizado para realizar a comparação de distância entre os usuários e realizar o cálculo e comparação entre as distâncias para apresentar em forma crescente os prestadores mais próximos de acordo com o filtro de distância definido e já formatado no formato km ou metros.

² Disponível em: <https://developer.android.com/guide/components/fragments> Acesso em: 01 outubro de 2020.

Figura 10: Classe localização

```

public class Localizacao {

    private static double Latitude;
    private static double Longitude;

    public Localizacao() {
    }

    public static void recuperarLatLng() {...}

    public static float calcularDistancia(Double prestadorLat, Double prestadorLng) {...}

    public static String formatarDistancia(float distancia) {...}

    public static double getLatitude() { return Latitude; }

    public void setLatitude(double latitude) { this.Latitude = latitude; }

    public static double getLongitude() { return Longitude; }

    public void setLongitude(double longitude) { this.Longitude = longitude; }
}

```

Fonte: Captura do autor em outubro de 2020.

4.1 Interface do usuário.

É primordial que o aplicativo apresente um *layout* robusto que torne a experiência de usuário agradável para os utilizadores do *app*, atentando aos padrões de usabilidade. Para auxiliar no processo de desenvolvimento, o site feito para desenvolvedores *Android*, o *Developers*, introduz um material sobre os conceitos básicos sobre design de interface. Demonstrando modelos padrões de animações, layouts, componentes, estilos e usabilidade.

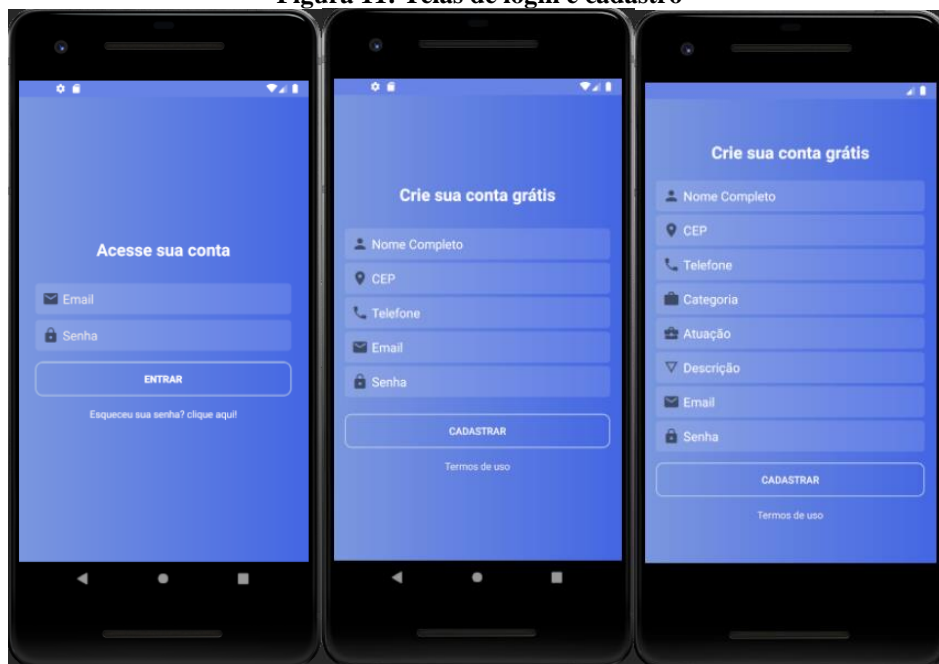
No processo de desenvolvimento a interação que o usuário possui com a interface do *app* é extremamente importante, uma “interface é tudo que utilizamos para interagir com algo” (JOAO, 2017), a interface pode definir o sucesso que um *app* pode alcançar e como eles irão se adaptar, os usuários não buscam um *app* complexo e difícil de entender para realizar uma tarefa simples, ou que apresente demasiada funcionalidades que não possuem utilidade.

Buscando seguir as aplicações modernas que estão no mercado para criar uma aplicação que seja aceitável a partir dos seus aspectos de custo, confiabilidade, utilidade e usabilidade, a aplicação desenvolvida buscou apresenta uma interface bem comum dentre o mercado *mobile*, no qual possui uma tela principal que permite navegar entre quatro telas com funções distintas sem muita dificuldade.

As telas de cadastro e login foram construídos baseados em padrões de *layouts* disponíveis no site de desenvolvedores do *Android* e também em similaridade a outras aplicações, com a utilização de ícones e descrição dos campos que permitem identificar qual informação deverá ser incluída nos campos.

Para facilitar a visualização e o cadastro foi desenvolvido uma interface para os usuários se cadastrarem apenas para buscar um prestador e uma segunda interface para os prestadores, no qual os permite especificar os detalhes dos seus serviços. Porém, mesmo o cadastro como prestador o permitirá realizar busca por outros prestadores dentro do *app*.

Figura 11: Telas de login e cadastro



Fonte: Captura do autor em outubro de 2020.

A tela principal, a “home”, é dividida em quatro *fragmentos*, sendo elas a *home*, pesquisa, mensagem e perfil. A *fragment home* apresenta os prestadores que estiverem cadastrados de acordo com a distância que o usuário especificar no filtro de pesquisa, o filtro é utilizado através de um botão na tela que exibe uma caixa flutuante e permite o usuário definir a distância que deseja encontrar um prestador.

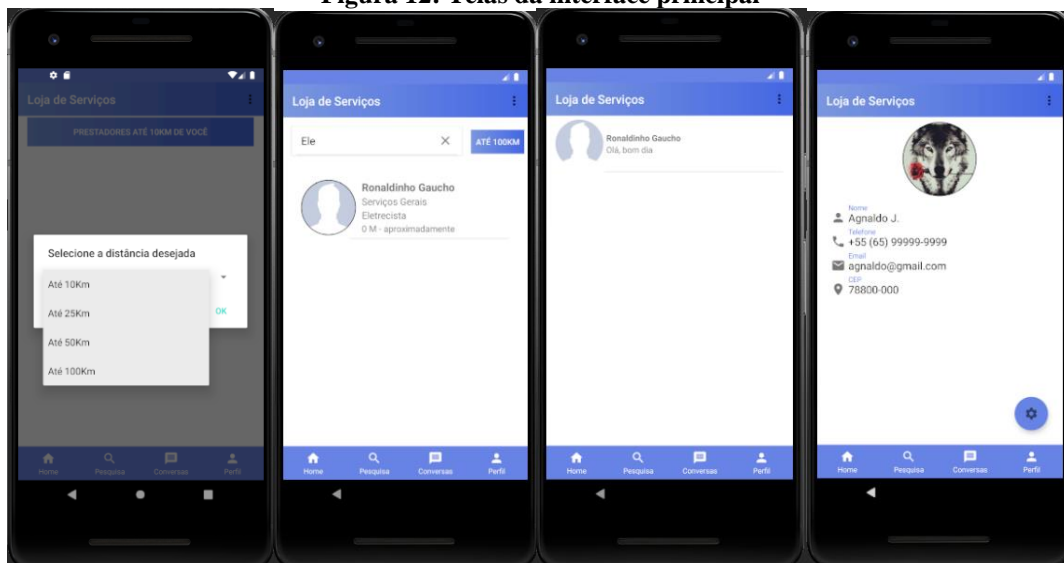
O *fragmento home* foi desenvolvida pensando em exibir qualquer prestador que esteja próximo do cliente para que os novos usuários possam visualizar o perfil de outros prestadores e tornar o usuário mais familiarizado com as informações dispostas no aplicativo, e então realizar uma busca pelo profissional desejado.

No *fragmento pesquisa* é exclusivamente para realizar pesquisas e apresentar os resultados no qual também utiliza o mesmo conceito de botão para a definição do filtro e um campo para digitar o serviço que procura, que retorna os resultados à medida que for digitando.

O *fragmento de mensagem* apresenta as conversas que já tenham sido iniciadas com outros prestadores e exibindo o nome da pessoa e a última mensagem que tenha sido enviado ou recebido.

Por fim, o *fragmento perfil* disponibiliza os dados cadastrados, a foto de perfil caso haja inclusa, e permite editar o perfil para realizar alteração de alguma informação através do botão flutuante no canto inferior com o símbolo de engrenagem.

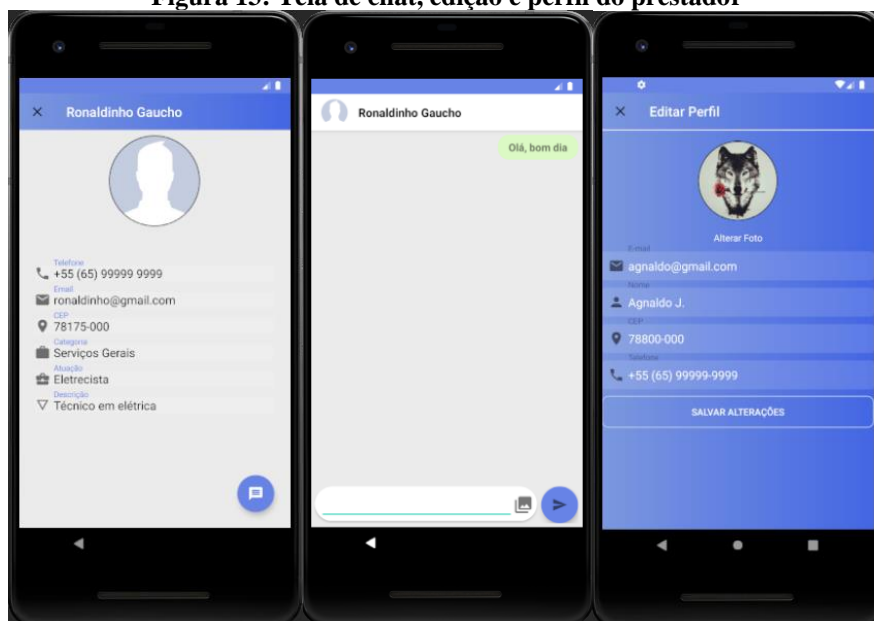
Figura 12: Telas da interface principal



Fonte: Captura do autor em outubro de 2020.

A tela de perfil do prestador possui a reutilização do fragmento perfil, com uma pequena alteração no botão flutuante que leva o usuário a tela de chat para enviar mensagens. A tela de chat é uma tela simples, que permite enviar mensagens de texto e imagens que estejam salvas na galeria do dispositivo. E por fim, a tela de edição dos dados do usuário que pode apresentar mais opções de edições de acordo com o perfil cadastrado.

Figura 13: Tela de chat, edição e perfil do prestador



Fonte: Captura do autor em outubro de 2020).

5. Resultados e Discussões

Tal aplicativo foi proposto pautado em dois pilares principais, auxiliar as pessoas nas atividades referente a encontrar um prestador de serviços básicos e garantir o máximo de compreensão e interação entre os usuários e a aplicação.

O aplicativo é de fácil acesso e compreensão, dispõe de interface intuitiva, com design agradável, pautados nos princípios de interação homem e máquina. Mantém a consistência e padrões aplicados em *softwares* semelhantes. Permite uso eficiente para pessoas iniciantes ou com conhecimento avançado, quanto ao uso.

Possibilita ao usuário localizar prestadores nas mais diversas áreas de atuação, e permite a indicação de outros prestadores disponíveis nas proximidades que sejam da mesma área ou de outras áreas de atuação.

Auxilia significativamente a comunicação e a divulgação dos prestadores de serviços, disponibilizando o perfil com informações sobre o telefone e e-mail, e até mesmo entrar em contato via mensagem dentro do próprio aplicativo.

A aplicação é responsiva, independente do dispositivo, a interface irá se adaptar ao tamanho da tela, possuindo cores chamativas para as principais funcionalidades do aplicativo e layout que apresentam poucas informações para visualização. A experiência do usuário ocorre a partir dos usuários cadastrados, caso a aplicação não apresente ou não disponha de prestadores cadastrados, a aplicação não irá entregar o objetivo principal. A opções de cadastro são intuitivas e funcionais, possibilitando ao usuário se logar a qualquer momento com as informações já cadastradas.

O aplicativo já possui seu objetivo principal em funcionamento, a busca por prestadores é realizada no banco de dados, conforme os prestadores estiverem cadastrados, e caso encontre algum prestador com o mesmo contexto da pesquisa, será retornado os prestadores em uma tabela, no qual possibilita o usuário clicar e visualizar o perfil cadastrado.

6. Considerações Finais

Com a evolução das tecnologias da informação, está cada vez mais acessível e comum para as pessoas encontrarem informações sobre empresas e pessoas através do *smartphone* ou até mesmo realizar uma compra online.

No processo de desenvolvimento deste projeto foi possível identificar as ferramentas e métodos que são utilizados no processo de desenvolvimento de um aplicativo nativo para *Android*, assim como é apresentado o seu sistema operacional.

A integração com o banco de dados da *Google*, o *Firebase*, se mostra promissora e eficiente, possibilitando realizar diversas tarefas dentro do banco de dados que geralmente exigem muito mais trabalho quando utilizados outros bancos de dados, como serviços de autenticação, criptografia e criação manual das tabelas de dados.

Apesar de funcional, o aplicativo não teve sua publicação na loja oficial do *Android*. O *app* “loja de serviços” está em processo de desenvolvimento e ajustes das suas funcionalidades e correção de bugs de interface. Também se objetiva adicionar uma forma de monetização para manter o aplicativo sem cobrar dos utilizadores, utilizando por exemplo anúncios, para auxiliar financeiramente a melhorar e manter o banco de dados com todas as funcionalidades possíveis sempre em funcionamento.

Para tornar mais seguro e confiável para os usuários se cadastrarem, buscaremos integrar a aplicação a outras redes sociais para que possa ser feito o cadastro através dos mesmos. Também buscamos possibilitar a publicação de fotos no perfil dos prestadores para visualização dos clientes quanto aos serviços que o mesmo presta, tornando assim mais eficiente a divulgação dos trabalhos que o prestador realiza e a confiança do cliente em entrar em contato.

Por fim, aprimorar a estrutura de cadastro dos usuários no banco de dados, para possuir mais informações sobre o endereço dos usuários e categorizar os endereços por estado e cidade, fazendo com que se torne mais otimizando a busca no banco de dados à medida que o *app* possa atingir outros estados no país.

A utilização de aplicativos e dispositivos móveis para realizar comprar e localizar lugares e serviços é uma área que está em constante crescimento à medida que a tecnologia cresce, por isso, espera-se que a aplicação possa atingir seu objetivo primário, auxiliando e crescendo juntamente com as necessidades da população.

REFERÊNCIAS

ANDRADE, Alisson Wilker; AGRA, Ronaldo; MALHEIROS, Viviane. **Estudos de caso de aplicativos móveis no governo brasileiro**. Brasília: SERPRO, p. 780-791, 2013.

Android Developer. Disponível em: <https://developer.android.com/>. Acesso em: 17/10/2020.

Android Studio. Disponível em: <https://developer.android.com/studio/intro?hl=pt-br>. Acesso em: 12/04/2020.

Blog PayPal. **76% dos brasileiros usam smartphones para fazerem compras online**. Disponível em: <https://www.paypal.com/stories/br/76-dos-brasileiros-usam-smartphone-para-fazer-compras-online>. Acesso em: 14/09/2020.

Caelum. **Java e Orientação a Objetos Curso fj-11**. Disponível em: <https://www.caelum.com.br/download/caelum-java-objetos-fj11.pdf>. Acesso em: 18/03/2020.

DUARTE, Luiz. **Criando apps para empresas com Android - 1º edição**. Gravataí/RS:2017.

Firestore. Disponível em: https://firebase.google.com/?hl=pt_br. Acesso em: 12/04/2020.

G1. **Número de desocupados diante da pandemia sobe para 12,2 milhões, aponta IBGE**. Disponível em: <https://g1.globo.com/economia/noticia/2020/07/31/numero-de-desocupados-diante-da-pandemia-sobe-para-122-milhoes-aponta-ibge.ghtml>. Acesso em: 14/09/2020

Java. Disponível em: https://www.java.com/pt_BR/about/whatis_java.jsp. Acesso em: 12/04/2020.

JOÃO, Belmiro do Nascimento. **USABILIDADE E INTERFACE HOMEM-MÁQUINA**. São Paulo: Pearson Education, 2017.

KHAWAS, Chunnu; SHAH, Pritam. **Application of Firebase in Android App Development- A Study**. International Journal of Computer Applications(0975-8887), Volume 179 - No. 46, June 2018.

LECHETA, Ricardo. **Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 4ª edição. São Paulo: Editora Novatec Ltda – 2015.

LOPEZ, Bianca. **Mobile no Brasil: estatísticas de uso e perfil do usuário**. Pag|Brasil, 14 de Mar. de 2019. Disponível em: <https://www.pagbrasil.com/pt-br/noticias/mobile-no-brasil/>. Acesso em: 01/04/2020.

Redação RBA. **Trabalho sem carteira e autônomo cresce, bate recorde e segura taxa de desemprego**. São Paulo, 2019. Disponível em: <https://www.redebrasilatual.com.br/economia/2019/09/trabalho-sem-carteira-e-autonomo-cresce-bate-recorde-e-segura-taxa-de-desemprego/>. Acesso em 14/03/2020.

RODRIGUES, André Felipe Angeloni; SILVEIRA, Fábio Stefanini; FUINI, Mateus Guilherme. **Yellow List: aplicativo Android para localizar prestadores de serviço utilizando GPS e integração com redes sociais**. FATEC de Itapira: Revista Prospectus, v.1, n. 1, p.50-86, Ago/Fev, 2019. Disponível em: <http://www.prospectus.fatecitapira.edu.br/index.php/pgt/article/view/17/8>. Acesso em: 18/04/2020.

SOMMERVILLE, Ian. **Engenharia de software** – 10ª edição. São Paulo: Editora Pearson - 2018.

Tecmundo. **Android é Linux? Qual a relação do SO da Google com o Linux**. Disponível em: <https://www.tecmundo.com.br/software/127038-android-linux-kernel.htm>. Acesso em 17/09/2020.