



UNIVERSIDADE DO SUL DE SANTA CATARINA
RENÔ CARNEIRO DA SILVA NETO

APLICATIVO MOBILE
PARA SERVIÇOS EM BARZINHOS DE KARAOKÊ

Florianópolis
2020

RENÔ CARNEIRO DA SILVA NETO

**APLICATIVO MOBILE
PARA SERVIÇOS EM BARZINHOS DE KARAOKÊ**

Trabalho de Conclusão de Curso apresentado
ao Curso de Sistemas de Informação da
Universidade do Sul de Santa Catarina como
requisito parcial à obtenção do título de
Bacharel em Sistemas de Informação.

Orientador: Prof. Aran Bey Tcholakian Morales, Dr.

Florianópolis

2020

RENÔ CARNEIRO DA SILVA NETO

**APLICATIVO MOBILE
PARA SERVIÇOS EM BARZINHOS DE KARAOKÊ**

Este Trabalho de Conclusão de Curso foi julgado adequado à obtenção do título de Bacharel em Sistemas de Informação e aprovado em sua forma final pelo Curso de Sistemas de Informação da Universidade do Sul de Santa Catarina.

Florianópolis, (dia) de (mês) de 2020.

Professor e orientador Aran bey Tcholakian Morales, Dr.
Universidade do Sul de Santa Catarina - UNISUL

Prof. Maria Inés Castiñeira, Dra.
Universidade do Sul de Santa Catarina - UNISUL

Prof. Vera R.N. Schuhmacher, Dra.
Universidade do Sul de Santa Catarina - UNISUL

RESUMO

A falta do uso da tecnologia atualmente em alguns tipos de serviço ainda é muito grande. Um exemplo são os bares de karaokê que ainda lidam com listas impressas. Um aplicativo visando trazer mais comodidade e praticidade para os frequentadores, e um diferencial para alguns bares seria uma solução. Para isto foi feito um estudo aprofundado em algumas tecnologias de desenvolvimento mobile. Uma vez escolhido o framework foi então feito a modelagem do sistema e a prototipação das telas. Em seguida o desenvolvimento do protótipo da aplicação foi feito e uma vez concluído foi submetido a alguns testes com alguns usuários. O questionário respondido após a utilização do sistema seguindo alguns passos previamente explicados, avaliou algumas questões como legibilidade e expectativas do usuário. Após analisar os resultados pode-se concluir que é um projeto muito interessante a ser aplicado pois de fato mudará a experiência dos usuários e dos administradores de bares de karaokê.

Palavras-chave: Mobile. Karaokê. Aplicativo.

ABSTRACT

The lack of in depth use of technology today in some types of services is still high. One example of this is Karaoke Bars, they still use folders with a long song list. An application aimed at bringing more convenience and practicality to regulars, and a differential for some bars would be a solution. For this, a study was done on some mobile development technologies. Once the framework was chosen, the system was modeled, and the screens were prototyped. Then the development of the application prototype was done and once completed it was subjected to some tests with some users. The questionnaire answered after using the system following some previously explained steps, evaluated some questions such as readability and user expectations. After analyzing the results, it can be concluded that it is a very interesting project to be applied because it will change the experience of users and administrators of karaoke bars.

LISTA DE ILUSTRAÇÕES

Figura 1 – Modelo MVC.	14
Figura 2 - Arquitetura da plataforma Android.....	16
Figura 3 - Arquitetura da plataforma iOS.....	17
Figura 4 - Exemplo da aplicação “GuDog”.....	20
Figura 5 - Exemplo da aplicação “Pet Stop”.	21
Figura 6 - Exemplo do "Theme Roller".....	22
Figura 7 - Exemplo do tema padrão do jQuery Mobile.....	23
Figura 8 - Exemplo da aplicação "HealthTap",	25
Figura 9 - Exemplo da aplicação "Instant Pot".....	27
Figura 10 - Exemplo da aplicação "Aggreko".	28
Figura 11 - Exemplo da aplicação “Discord”.....	30
Figura 12 - Exemplo da aplicação do "NuBank".	31
Figura 13 - Arquitetura da solução proposta.	34
Figura 14 - Protótipo da tela de login	39
Figura 15 - Diagrama de caso de uso - UC1 - Cadastrar música.....	42
Figura 16 - Diagrama de caso de uso - UC2 - Gerenciar a lista de execução	43
Figura 17 - Arquitetura do sistema.	44
Figura 18 - Tela de entrada do aplicativo.	47
Figura 19 - Tela de lista de músicas.	48
Figura 20 - Tela de pesquisa.....	48
Figura 21 - Tela de entrada do administrador.....	49
Figura 22 - Lista de músicas pelo perfil de admin.	50
Figura 23 - Tela de inserção de música nova.	50
Figura 24 - Menu de editar ou remover músicas.	51
Figura 25 - Menu de edição de música.....	51
Figura 26 - Lista de músicas em execução.....	52
Figura 27 - Remover da lista de execução.....	52

LISTA DE GRÁFICOS

Gráficos 1 - Legibilidade das telas.	53
Gráficos 2 – Títulos de botões e telas.....	54
Gráficos 3 - Feedback das ações.....	54
Gráficos 4 - Avaliação das telas.	55
Gráficos 5 - Avaliação das necessidades atendidas.....	55
Gráficos 6 - Avaliação do tempo de resposta.	56
Gráficos 7 - Avaliação do conforto do usuário.....	56

LISTA DE QUADROS

Quadro 1 - Quadro de comparação dos frameworks.	32
Quadro 2 – Protótipo das telas de lista de música e pesquisa respectivamente.....	40
Quadro 3 - Protótipo das telas de lista de execução e painel de administrador.....	41

SUMÁRIO

1	INTRODUÇÃO.....	10
1.1	OBJETIVOS	11
1.1.1	Objetivo geral	11
1.1.2	Objetivos específicos	11
1.2	JUSTIFICATIVA	11
1.3	ESTRUTURA DA MONOGRAFIA	12
2	REVISÃO BIBLIOGRÁFICA	13
2.1	APLICAÇÕES WEB	13
2.2	APLICAÇÕES MOBILE.....	15
2.2.1	Desenvolvimento de Web app	17
2.2.2	Desenvolvimento de App Nativo	17
2.3	APLICAÇÕES HÍBRIDAS	18
2.3.1	Framework.....	18
2.3.1.1	Apache Cordova	18
2.3.1.2	Framework 7.....	20
2.3.1.3	JQuery Mobile	21
2.3.1.4	Adobe PhoneGap.....	23
2.3.1.5	Ionic	25
2.3.1.6	Xamarin	27
2.3.1.7	React Native	28
2.3.1.8	Flutter	30
2.4	CONSIDERAÇÕES FINAIS.....	32
3	MÉTODO	33
3.1	CARACTERIZAÇÃO DO TIPO DE PESQUISA	33
3.2	ETAPAS METODOLÓGICAS	33
3.3	ARQUITETURA DA SOLUÇÃO PROPOSTA	34
3.4	DELIMITAÇÕES	35
4	MODELAGEM.....	36
4.1	REQUISITOS	36
4.1.1	Requisitos funcionais	36
4.1.2	Requisitos não funcionais	38
4.2	PROTOTIPAGEM.....	38

4.3	CASOS DE USO	41
4.3.1	Casos de uso do sistema	42
5	DESENVOLVIMENTO	44
5.1	ARQUITETURA E TECNOLOGIAS	44
5.1.1	Visual Studio Code	45
5.1.2	SDK Plataform-Tools.....	45
5.1.3	Genymotion.....	45
5.1.4	JavaScript	45
5.1.5	React Native	45
5.1.6	Firebase Real Time Database.....	45
5.1.7	React Native Paper	46
5.1.8	React Navigation	46
5.2	APRESENTAÇÃO DO PROTOTIPO	46
5.3	VALIDAÇÃO DO PROTÓTIPO	53
5.3.1	Análise dos resultados obtidos	53
5.4	CONCLUSÕES DO PROTÓTIPO.....	57
6	CONCLUSÃO E TRABALHOS FUTUROS.....	58
6.1	CONCLUSÕES	58
6.2	TRABALHOS FUTUROS	58
	REFERÊNCIAS	60

1 INTRODUÇÃO

Dado os atuais avanços da tecnologia, está cada vez mais fácil trazer algumas comodidades e praticidades para o nosso dia a dia. Seja através de aplicativos como por exemplos os de banco, onde o usuário pode fazer quase tudo que precisaria fazer em uma agência, diretamente do conforto da sua casa ou no meio de uma viagem a trabalho, ou também através de sites e outras ferramentas tecnológicas.

Cada vez mais as pessoas procuram lugares e lazeres que atendam esta necessidade e tragam esse tipo de conforto ao usuário.

Algumas áreas do comércio com o tempo vêm reparando nisso e se inovando e renovando dia a pós dia. Algo que está demorando para acontecer na maioria dos bares e restaurantes.

Apesar de usarem aplicativos de tele entrega como o iFood ou o Rappi, ainda usam comandas físicas de papel, cardápios de papel, dependendo do dia e do movimento pode haver filas enormes de espera para entrar ou simplesmente para pagar e sair, ocorrem erros com os pedidos e as mesas, valor a pagar calculado manualmente, estoque é verificado e atualizado manualmente entre outras coisas.

Um exemplo é como o simples fato de ir em um barzinho de karaokê pode se tornar de uma experiência agradável e relaxante para algo desagradável ou estressante, e até desestimular o cliente, caso o mesmo tenha alguns problemas e empecilhos para realizar tarefas básicas e simples antes de poder cantar sua música. Esse é o caso de alguns bares do gênero, na grande Florianópolis.

O ato de ter que procurar o “Id” da música que o cliente deseja interpretar, ou de informar ao estabelecimento, ou quem estiver cuidando da lista de próximos participantes, não deveria ser algo que fizesse o cliente ter algum tipo de espera ou trabalho. Afinal de contas são tarefas simples. Acontece que quando o usuário precisa procurar este “Id” em uma lista de registro enorme, com inúmeros registros, de forma manual, seja ela por ordem alfabética, ordem de gênero ou qualquer outra, levando às vezes mais de 10 minutos para achar, ou ter que enfrentar uma fila para informar ao atendente qual música ele deseja, pode vir a fazer o usuário repensar em retornar ao estabelecimento.

1.1 OBJETIVOS

A proposta deste trabalho é trazer comodidade e praticidade para os clientes de barzinhos de karaokê, bem como trazer um diferencial e um atrativo para os bares que o implementarem.

1.1.1 Objetivo geral

Sendo assim, o objetivo geral do trabalho é desenvolver uma aplicação mobile voltada a serviços específicos de bares de karaokê.

1.1.2 Objetivos específicos

- Analisar e comparar diversos frameworks para o desenvolvimento de aplicações mobile e web.
- Desenvolver uma aplicação com o serviço de acesso as listagens de músicas desejadas, e ao cadastro da fila de espera para sua apresentação.
- Gerar maior conforto, e uma boa experiencia, ao cliente do bar que implementar tal solução para que ele retorne mais vezes ao estabelecimento.
- Gerar um diferencial para os bares que implantarem a solução, fazendo os mais atrativos aos olhos dos clientes e investidores.

1.2 JUSTIFICATIVA

Um bom estabelecimento está sempre preocupado em como atender melhor seus clientes e fazê-los voltarem o mais breve possível. Pensando nisso a solução geraria maior conforto, uma boa experiencia ao cliente e seria um diferencial para os bares que optarem por implementá-la em seus estabelecimentos. Fazendo com que eles sejam mais atrativos e atendam essas demandas, suprimindo algumas necessidades e comodidades dos clientes e atraindo a atenção de possíveis investidores.

Com uma aplicação hibrida ou multiplataforma, e tendo em vista que a maioria das pessoas possuem Smartphones hoje em dia, seria muito simples procurar a música desejada, bem como colocar ela na lista de próximas a serem executadas diretamente da mesa, evitando transtornos e estresses com filas, espaços para passar até chegar ao balcão e outros.

1.3 ESTRUTURA DA MONOGRAFIA

Capítulo 1: No capítulo em questão se encontra a introdução ao problema encontrado, os objetivos que se deseja alcançar e as justificativas para realização desta presente pesquisa.

Capítulo 2: Neste capítulo consta a revisão bibliográfica para o tema abordado e é apresentado o embasamento teórico bem como algumas ferramentas necessárias para realização da solução do problema.

Capítulo 3: Este capítulo aborda a respeito da metodologia utilizada nesta pesquisa e as delimitações do projeto.

Capítulo 4: Capítulo cujo intuito é apresentar a modelagem do sistema proposto.

Capítulo 5: Este capítulo é destinado a apresentar o desenvolvimento do sistema.

Capítulo 6: Apresenta as conclusões do trabalho monográfico e sugestões para trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo, são abordados conceitos técnicos para servir de fundamentação teórica e trazer uma melhor compreensão sobre desenvolvimento de aplicações web e mobile. Tendo como foco o desenvolvimento multiplataforma.

2.1 APLICAÇÕES WEB

Para Bertagnolli e Miletto (2014, p. 3), “[...] quando o software for desenvolvido para a plataforma web, vários aspectos são incorporados de modo que ele possa ser acessado de forma remota e segura por meio de um navegador.”

Ainda segundo Bertagnolli e Miletto (2014) um software web precisa ser separado em camadas para facilitar e ser possível a implementação de correções, alterações ou atualizações futuras, de maneira independente, sem que as demais partes sejam afetadas. Essas camadas são abordadas pelos autores como: de visão ou apresentação, de processamento, de gerenciamento e a camada de banco de dados e estão ligadas as boas práticas de desenvolvimento de software.

A figura abaixo mostra um padrão de projeto que pode ser aplicado quando se trata de aplicações web, o MVC (Model, View e Controller) que em português é conhecido como a camada de visão, camada de modelo ou negócio, e a camada de controle.

Figura 1 – Modelo MVC.



Fonte: <https://cleveronsacramento.wordpress.com/2011/10/17/mvc-ou-arquitetura-em-camadas/> acesso em 19 nov. 2019.

Costa (2007) afirma como sendo o HTML ou o Hypertext Markup Language, a linguagem padrão de acesso e visualização de páginas Web e que ela é de uma forma genérica, composta por textos e códigos comumente chamados de tags.

Segundo Costa (2007, p. 7) “[...] originalmente o HTML definia estritamente a estrutura lógica de um documento, e não a sua aparência física. Mas, com a pressão dos utilizadores, as versões posteriores do HTML, foram forçadas a fornecer cada vez mais controles de aparência.”

Flatschart (2011), define o CSS como sendo uma linguagem de estilo. Diferente do HTML que é uma linguagem de marcação, o CSS é responsável pela formatação e apresentação do site e seu conteúdo.

Visando adicionar ainda mais interatividade e funções que o HTML e o CSS não possibilitavam, existe o JavaScript que é uma linguagem de programação a qual segundo Flatschart (2011), pode ser incorporada diretamente nas páginas HTML e é classificada como do tipo cliente-side, ou seja, na máquina do usuário web. Ele ainda cita como algumas das características do JavaScript como sendo uma linguagem de programação com sintaxe simples

e de fácil aprendizagem e pode ser programado para interpretar e modificar alguns conteúdos de um ou mais elementos do HMTL.

2.2 APLICAÇÕES MOBILE

Segundo Gabriel (2010), dispositivos móveis não são somente celulares (Smartphones), mas pode-se considerar também, tablets, e-books readers (como o kindle), alguns consoles de games (como o PSP da Sony) alguns relógios, e em um futuro não tão distante, com os avanços poderemos ver até em óculos, brinquedos e outros dispositivos.

Atualmente existem diversos sistemas e plataformas nas quais as aplicações mobile são desenvolvidas. Algumas mais conhecidas e extremamente presentes no nosso dia a dia como é o caso do Android ou o iOS, e outras não tão conhecidas ou comumente utilizadas como o Firefox OS, BlackBerry, Ubuntu Touch, Fire OS, entre outras.

O Android, comumente conhecido por ser da Google, é uma plataforma aberta. Para Pereira e Silva (2009) por ser aberto pode sempre ser adaptado para incorporar novas tecnologias a medida com que elas forem surgindo, sendo assim a plataforma está em constante evolução.

Ainda segundo Pereira e Silva (2009), a plataforma tem como base do seu desenvolvimento o sistema operacional Linux, mas não é um Linux, pois não possui alguns dos padrões apresentados em algumas distribuições Linux.

A figura a seguir tem como propósito trazer uma breve exemplificação da arquitetura da plataforma Android.

Figura 2 - Arquitetura da plataforma Android.

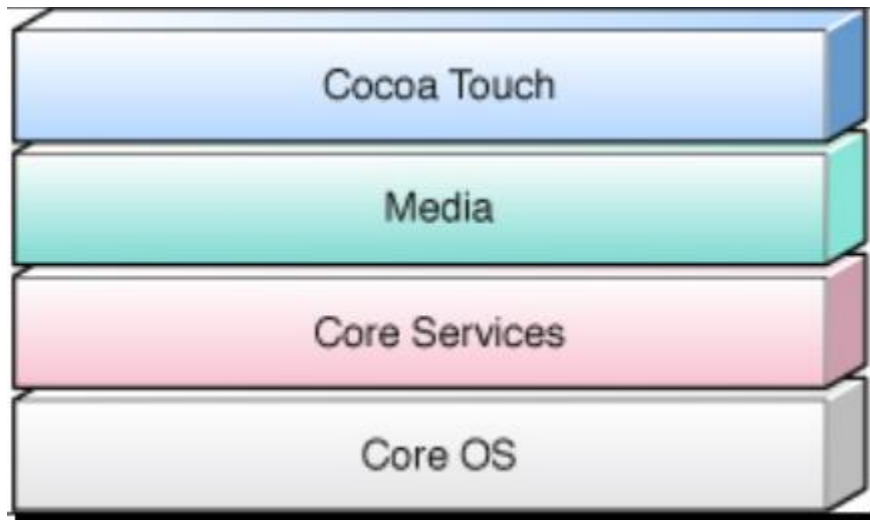


Fonte: <https://pt.slideshare.net/TheRonildoOliveira/arquitetura-da-plataforma-android> acesso em 19 nov. 2019.

O iOS, comumente conhecido por ser da Apple e concorrente direto do Android, tem segundo Morimoto (2009), a base do sistema como uma versão reduzida do MacOS X, adaptada para rodar com as limitações de hardware dos dispositivos, tendo diferenças quando se trata de interface e suporte à alguns aplicativos, porem alguns componentes básicos ainda são os mesmos do sistema base.

A figura abaixo tem como objetivo informar quanto a arquitetura do sistema iOS.

Figura 3 - Arquitetura da plataforma iOS.



Fonte: <http://www.purplechaiblogger.com/the-different-layers-of-ios-architecture/> acesso em 19 nov. 2019.

Quando se fala em aplicação mobile, o desenvolvedor pode optar por algumas opções, entre elas: Desenvolvimento de um Web App ou desenvolvimento de um app nativo.

2.2.1 Desenvolvimento de Web app

Segundo Prezotto e Boniati (2014), Web app se trata de aplicações que são destinadas a rodar no browser do dispositivo mobile, evitando que o desenvolvedor se preocupe qual sistema ele deseja utilizar a aplicação, afim de optar entre diferentes linguagem para desenvolver, pois eles utilizam padrões web como HTML, CSS e JavaScript na parte do cliente e na parte do servidor Java ou outras linguagem.

Entretanto Prezotto e Boniati (2014) ressaltam que é importante lembrar que ainda que se utilize a mesma linguagem para diferentes sistemas operacionais, essas aplicações correm o risco de apresentar comportamentos diferentes caso o browser do dispositivo não suporte algum componente da aplicação ou ele seja processado graficamente de outra maneira.

2.2.2 Desenvolvimento de App Nativo

Um app nativo, segundo Prezotto e Boniati (2014) “[...] são as que foram desenvolvidas especialmente para aquela plataforma e somente para ela, utilizando para isso as ferramentas disponibilizadas para a mesma, como a linguagem de desenvolvimento ambiente e emulador”.

Prezotto e Boniati (2014) ainda destacam que os aplicativos desenvolvidos na linguagem nativa tendem a ter um melhor desempenho, mesmo com o avanço de outras tecnologias de desenvolvimento.

2.3 APLICAÇÕES HÍBRIDAS

As aplicações híbridas trazem uma ideia de juntar os app web e os app nativos e partem do princípio de que são desenvolvidas para funcionar em qualquer dispositivo, independente de qual plataforma ele use, através do mesmo código fonte.

Prezotto e Boniati (2014) destacam que normalmente essas aplicações são desenvolvidas utilizando linguagens Web que são interpretadas pelo browser nativo do sistema, porem apesar de serem utilizada as linguagens web, elas não são aplicações web pois ficam instaladas no dispositivo.

2.3.1 Framework

Pressman e Maxim (2016, p. 351) afirmam que “[...] framework é uma “mini arquitetura” reutilizável que serve como base para e a partir da qual outros padrões de projetos podem ser aplicados.”

Ainda segundo Pressman e Maxim (2016) o framework pode ser incorporado ao código e são reutilizados a fim de ajudar e agilizar o desenvolvimento.

De uma maneira mais simples, um framework pode ser considerado como uma estrutura de classe que possui diversas implementações incompletas que podem ser usadas para reduzir o tempo na produção de um software ou aplicação.

A seguir são apresentados alguns frameworks que nos possibilitam e facilitam a criação de aplicações híbridas.

2.3.1.1 Apache Cordova

A ASF (*Apache Software Foundation*) é uma organização de caridade financiada por patrocinadores corporativos e doações individuais, fundada em 1999. Conta com mais de 350 projetos de código aberto incluindo o Apache HTTP Server que é atualmente o servidor Web mais popular do mundo (APACHE, 2019).

Um desses projetos é a ferramenta Cordova, que segundo o próprio site, se trata de um framework *open-source* que se utiliza de HTML5 CSS3 e JavaScript para desenvolvimento multiplataforma (CORDOVA, 2019).

Ainda segundo Cordova (2019), ele é recomendado para quem trabalha com mobile e quer usar em mais de uma plataforma sem ter que reimplementar código para cada uma, também sendo muito útil para quem trabalha com desenvolvimento web e quer implementar em mobile.

O framework é provido de diversos plugins e estes são responsáveis por prover interfaces que permitem a comunicação entre ele e os componentes nativos. Esses plugins ou “Core Plugins” como são conhecidos, permitem às aplicações terem acesso a diversos recursos do dispositivo, como por exemplo câmera, contatos, localização (CORDOVA, 2019).

Além dos “Core Plugins” existem diversos plugins de terceiros ou “third-party” como são chamados, que disponibilizam acesso a recursos não disponíveis em todas as plataformas e podem ser encontrados oficialmente pelos links: “<https://cordova.apache.org/plugins/>” e “<https://www.npmjs.com/search?q=ecosystem%3Acordova>”. O framework também permite criar e utilizar seus próprios plugins.

Atualmente dispõe de dois caminhos na hora de criar um app, mas embora seja possível usar ambos para atingir um mesmo objetivo, cada um é designado em busca de uma vantagem diferente.

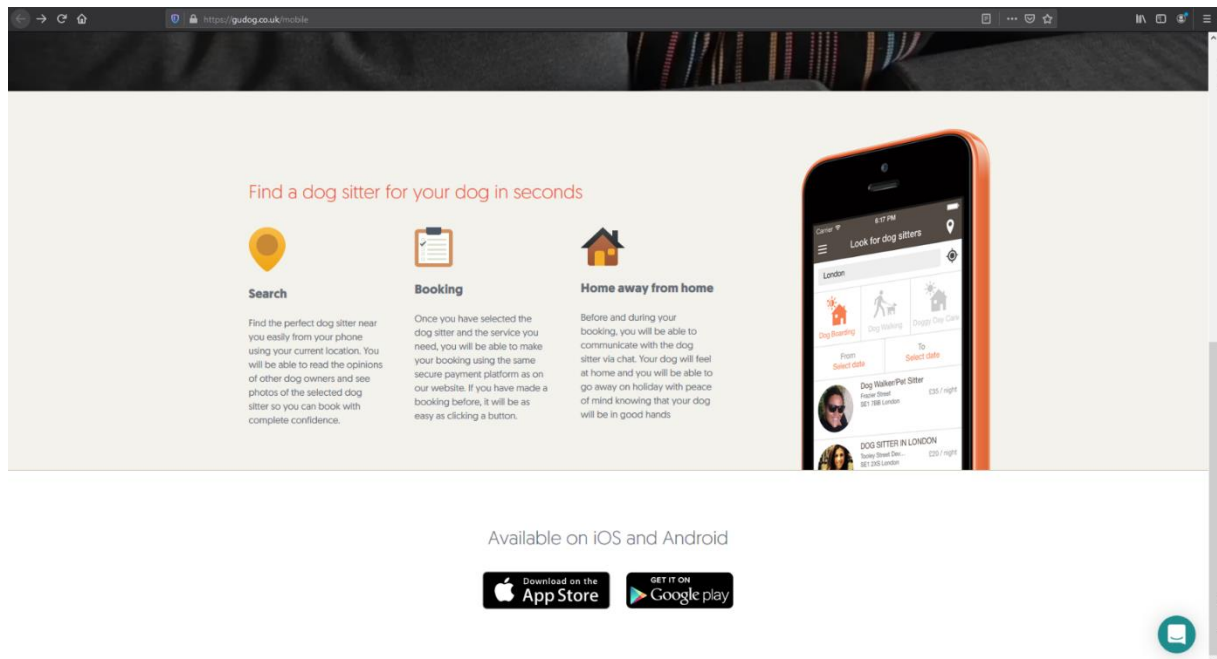
Tem-se o caminho de CLI, que é recomendado para quem quer rodar o app em todos os diferentes tipos de sistemas operacional mobile e sem nada muito específico de cada um deles.

Tem-se também o caminho focado na plataforma, que como o próprio nome já diz, é recomendado para quem quer focar em criar um app para uma plataforma específica somente (CORDOVA, 2019).

É importante ressaltar que o framework, por trabalhar com recursos nativos dos dispositivos e plataformas às quais atende, possui algumas restrições. Como é o caso de sistemas IOS, no que se vê necessário que seja compilado em um Mac e no caso de dispositivo Windows Phone, deve ser compilado em um ambiente Windows. Porém ambos os sistemas operacionais são capazes de compilar para Android (CORDOVA, 2019).

A figura abaixo tem como objetivo mostrar o exemplo da aplicação “GuDog” desenvolvida utilizando o Cordova, onde pode ser observado que atualmente atende as plataformas Browser, Android e iOS).

Figura 4 - Exemplo da aplicação “GuDog”.



Fonte: <https://gudog.co.uk/mobile>, acesso em: 26 out. 2019.

2.3.1.2 Framework 7

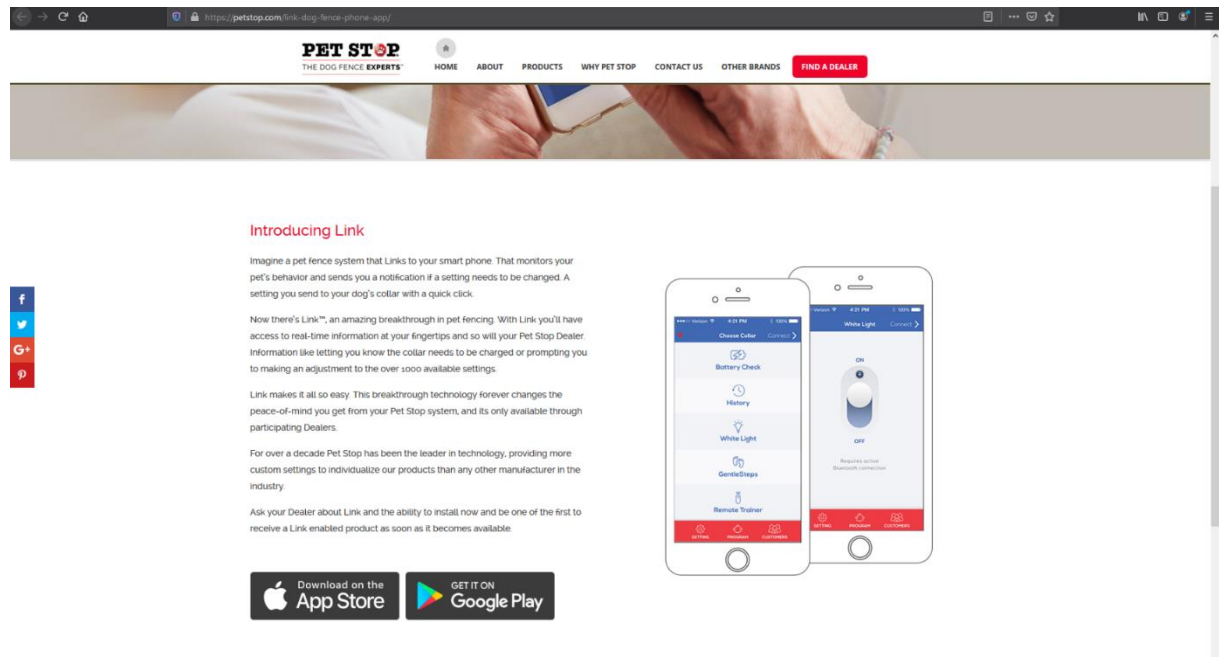
Trata-se de um framework *open-source* para desenvolvimento mobile ou aplicativos web com a aparência de um sistema IOS ou Android. Também atua como uma ferramenta para aplicativos de prototipagem funcionando muito bem em casos de necessidade de mostrar o protótipo do aplicativo em funcionamento o mais rápido possível (FRAMEWORK 7, 2019).

A ideia principal do framework7 é oferecer ao usuário a oportunidade de criar aplicativos Android ou IOS com muita facilidade e clareza, utilizando HTML, CSS e JavaScript. Porém a ferramenta não é compatível com todas as plataformas. Focando somente em iOS e materiais da Google (FRAMEWORK 7, 2019).

Para facilitar ainda mais a criação de projetos utilizando a ferramenta, recomendasse a utilização do Framework 7 CLI, que fornece uma quantidade enorme de combinações com diferentes templates, plataformas, outras ferramentas e algumas das melhores práticas. Um bom exemplo, é o fato de poder integrar o Framework 7 com o Apache Cordova, com React e alguns outros frameworks.

Assim como o Cordova, o Framework 7 é provido de diversos plugins e permite o usuário de criar os seus próprios (FRAMEWORK 7, 2019).

Figura 5 - Exemplo da aplicação “Pet Stop”.



Fonte: <https://petstop.com/link-dog-fence-phone-app/> acesso em: 26 out. 2019.

2.3.1.3 JQuery Mobile

Construído com as bibliotecas jQuery e jQuery UI, o jQuery Mobile é um framework projetado para criar sites e aplicativos de maneira responsiva, a fim de atender todos os dispositivos de smartphone, tablet e desktop. Personalizado para interação por toque, também conta com navegações em Ajax¹ com transição de página, Widgets² e um código leve com aprimoramento progressivo (JQUERY MOBILE, 2019).

Entre algumas de suas características, ele se destaca por seu tamanho pequeno e com pouca necessidade de imagem, o que o torna leve e mais rápido para carregar (TEIXEIRA, 2013).

O framework possibilita que o usuário escreva seus próprios estilos personalizados para criar layouts próprios, ou use templates disponibilizados.

Algo que chama muita atenção e diferencia de outros frameworks para desenvolvimento mobile ou multiplataformas é o fato de que ele não necessita que seja feita nenhuma instalação. Basta criar em um HTML uma marcação que referencie à biblioteca jQuery, uma outra ao arquivo JavaScript do framework e uma última para as folhas de estilo padrão do framework,

¹ É uma técnica de desenvolvimento web que tem como objetivo principal tornar as respostas das páginas Web mais rápidas.

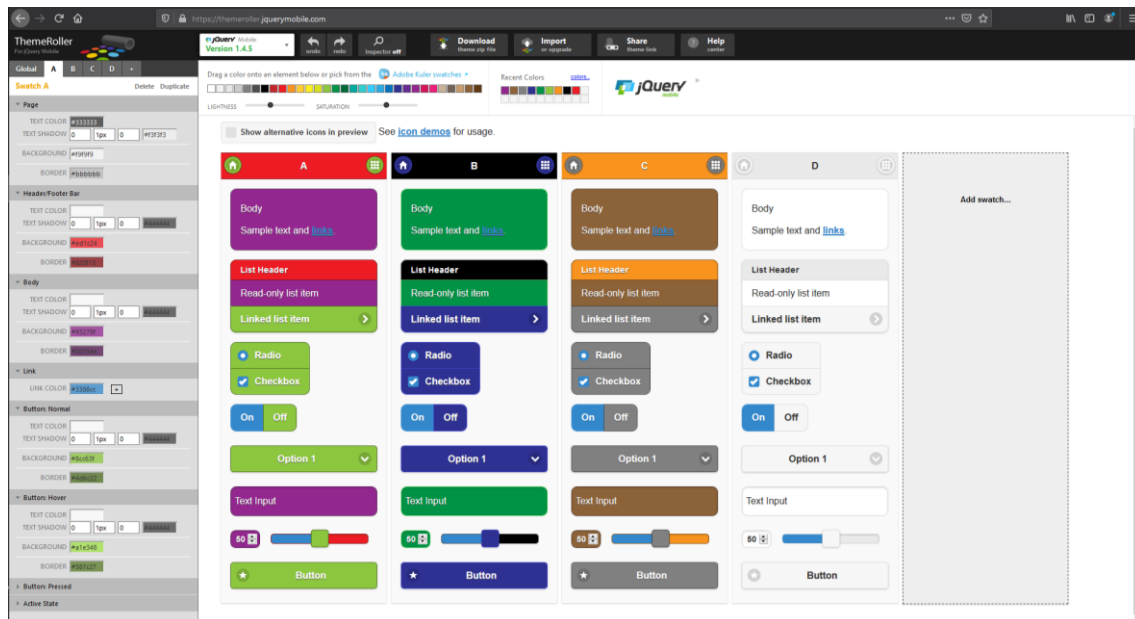
² É um componente para simplificar o acesso a algum programa.

facilitando ainda mais na hora do usuário criar suas aplicações. Esses arquivos podem ser baixados diretamente do site do framework ou como os desenvolvedores sugerem como uma boa prática, referenciá-los aos servidores de distribuição de conteúdo (CDN) (TEIXEIRA, 2013).

jQuery possui uma ferramenta chamada “Theme Roller” que lhe possibilita e facilita ainda mais a criação de temas personalizados. Para gerar um tema novo, basta estar um pouco familiarizado com o framework e “arrastar e dropar” os elementos na tela, e após concluído e no gosto do usuário, é só baixar e importar no seu projeto. Abaixo está um exemplo de alguns elementos e cores disponíveis na ferramenta (JQUERY MOBILE, 2019).

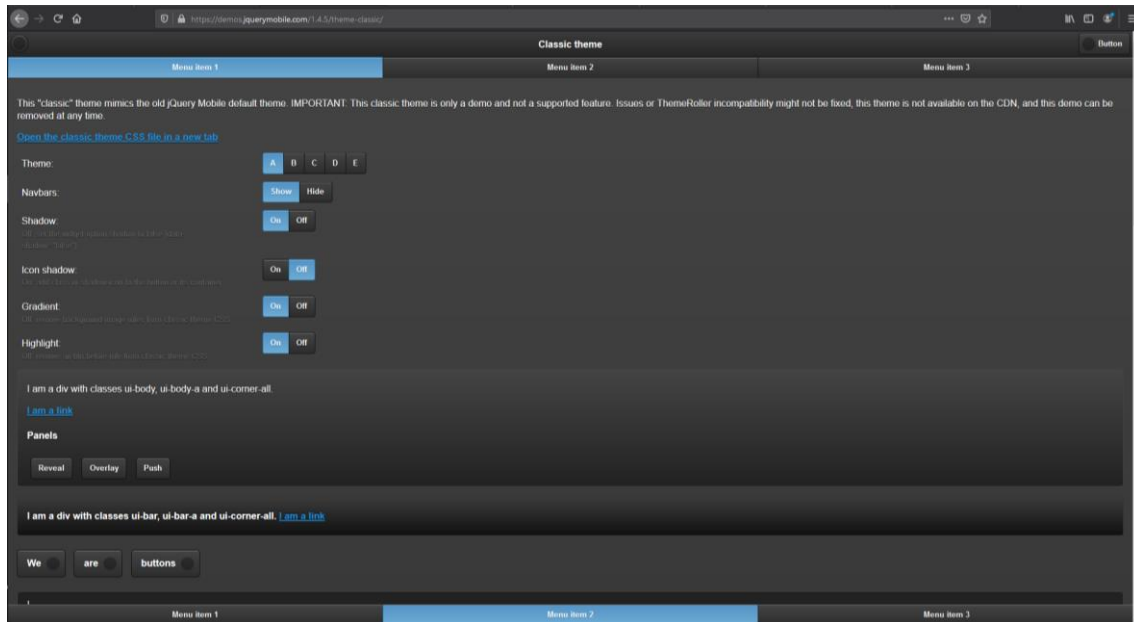
A figura a seguir apresenta um exemplo de como funciona, e é simples, o "Theme Roller", para criar um tema personalizado.

Figura 6 - Exemplo do "Theme Roller".



Fonte: <https://themeroller.jquerymobile.com/> acesso em: 26 out. 2019.

Figura 7 - Exemplo do tema padrão do jQuery Mobile.



Fonte: <https://demos.jquerymobile.com/1.4.5/theme-classic/> acesso em: 26 out. 2019.

2.3.1.4 Adobe PhoneGap

Este framework é uma distribuição open-source do Apache Cordova, o qual tem a mesma proposta de criar aplicativos utilizando HTML, CSS e JavaScript, porém, por ser desenvolvido e pertencer a Adobe, o PhoneGap possui algumas integrações com serviços da empresa (PHONEGAP, 2019).

O PhoneGap possui, assim como o Cordova, alguns caminhos que podem ser percorridos pelo novo usuário. Tem-se a opção de ir pelas linhas de comando (CLI), onde o usuário, para testar e pré-visualizar seu aplicativo, pode fazê-lo através de algum navegador, ou com seu próprio dispositivo, utilizando o “PhoneGap Developer App” sem a necessidade de instalar nenhum SDK (Kit de desenvolvimento de software) específico de alguma plataforma, registrando dispositivo ou compilando o código. O PhoneGap CLI inicia um pequeno servidor com o projeto e informa os dados para serem acessados através do aplicativo do PhoneGap que ao conectar, lhe permite fazer essa pré-visualização (PHONEGAP, 2019).

Existe também o segundo caminho, que seria mais voltado a usuários “visuais” que tendem a se dar melhor com interfaces do tipo “arrastar e dropar”. Para tal, basta instalar o “Desktop App”, que atualmente possui instalador para Windows e Mac OS X.

Em ambos os casos, seja através do desktop app ou do caminho CLI, é aconselhável a instalação do “Developer App” para facilitar a pré-visualização e conseguir fazer ajustes mais precisos (PHONEGAP, 2019).

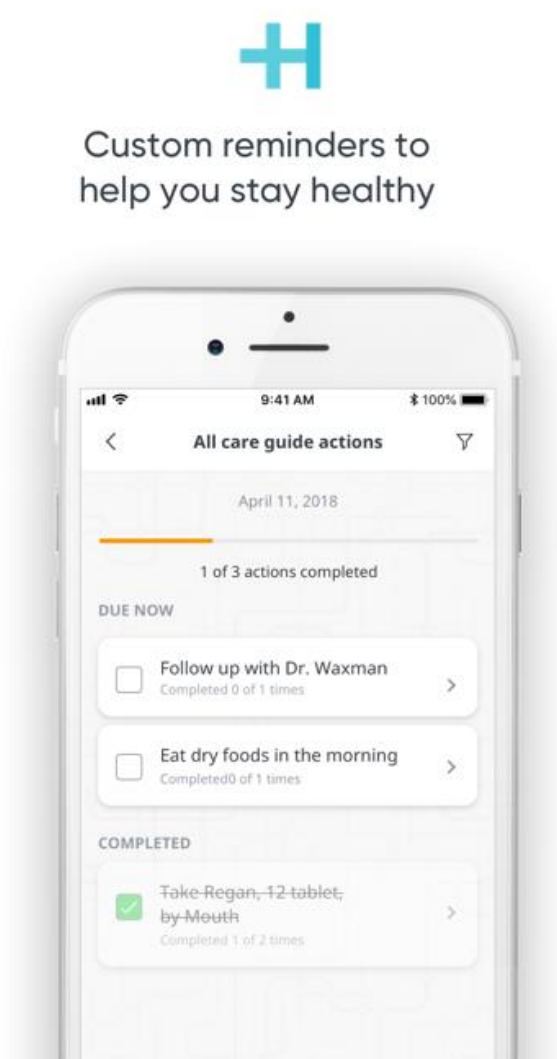
Assim como o Cordova, o PhoneGap possui algumas restrições com relação a compilar o aplicativo para as plataformas destino, porém ele traz uma segunda opção. O usuário pode tanto ir pelo método semelhante ao do Cordova, através da CLI, porém precisará de um sistema Mac para dispositivos IOS e um sistema Windows para Windows phone, e instalar as bibliotecas necessárias para cada plataforma, ou poderá utilizar do serviço “PhoneGap Build”.

O PhoneGap Build trata-se de um serviço em nuvem que lhe permite compilar o aplicativo diretamente do repositório, de qualquer lugar, a qualquer momento, sem precisar instalar nada.

Na questão dos plugins ele se utiliza dos mesmos do Cordova e funciona da mesma maneira caso o usuário queira criar os próprios. Porém ele se destaca por ter vários aplicativos terceiros (Third party) criados pela comunidade e que atendem a exigências e as necessidades dos usuários, como é o exemplo do “ConfiGAP”, que foi desenvolvido exclusivamente para ajudar o usuário a criar o arquivo de configuração utilizado pelo Adobe PhoneGap Build (PHONEGAP, 2019).

Abaixo é apresentado um exemplo da aplicação "HealthTap", que foi desenvolvida utilizando o Adobe PhoneGap.

Figura 8 - Exemplo da aplicação "HealthTap",



Fonte: <https://phonegap.com/app/healthtap-find-doctors-and-free-answers/> acesso em: 26 out. 2019.

2.3.1.5 Ionic

O Ionic é um framework open-source que utiliza tecnologias de desenvolvimento web (Html, Css e JavaScript), para criar aplicativos mobile de alto desempenho e qualidade. O framework é focado na experiência do usuário e de fácil aprendizagem. Ele também possui uma boa integração com outras bibliotecas ou outros frameworks, ou pode ser utilizado de maneira independente (IONIC, 2019).

Possui por padrão um design bem simples, limpo e funcional, que foi projetado para funcionar em qualquer plataforma (IONIC, 2019).

A ferramenta também possui uma CLI que implementa rapidamente aplicativos Ionic e traz vários comandos uteis na hora de desenvolver. Além de instalar o framework a CLI atualiza

e dispõe de um servidor de desenvolvimento interno, algumas ferramentas de compilação e algumas de depuração.

Ionic tem a cada versão se preocupado mais com a compatibilidade, desempenho e extensibilidade da ferramenta. Sendo assim, o Ionic V4+, atual versão, conta com a integração padrão com Angular, porém agora de maneira independente da sua estrutura, o que lhe possibilita a integração com qualquer outra estrutura JavaScript (Vue, React, Preact, entre outros) ou sem nenhuma estrutura integrada (IONIC, 2019).

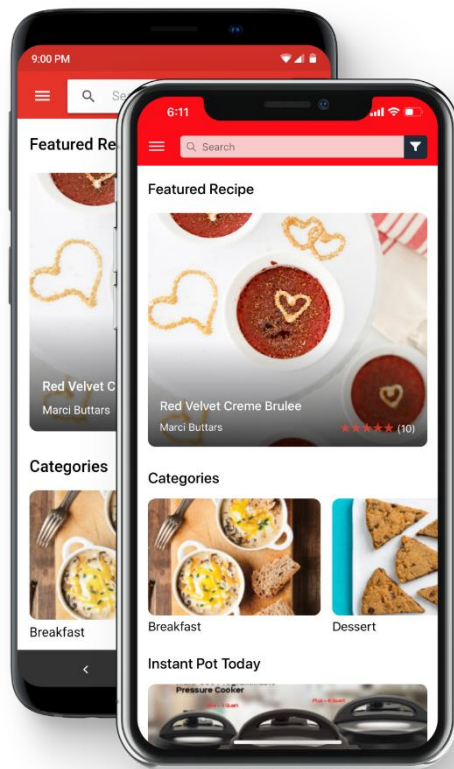
O framework conta com um recurso interno chamado de “Platform Continuity” que adapta a aparência de todo componente Ionic para a plataforma na qual o aplicativo está sendo executado, como por exemplo os dispositivos Android que utilizam a linguagem de design do Google chamada de “Material Design” (IONIC, 2019).

Quando se trata de rodar e testar o aplicativo criado, o Ionic não deixa a desejar, possibilitando que isto seja feito de várias maneiras, conforme o usuário se sinta mais à vontade.

Visando facilitar na hora de testar, Ionic disponibiliza o “Ionic DevApp” que oferece uma visualização em tempo real das alterações e com uma série de plugins a fim de testar os recursos nativos do aplicativo. Para isto, não se faz necessária nenhuma instalação de nenhum SDK. Basta que o usuário baixe o DevApp na loja do seu dispositivo, e esteja conectado na mesma rede do computador. Após verificado isso, basta um simples comando e o App já aparecerá no dispositivo para o teste (IONIC, 2019).

A figura a seguir apresenta um exemplo da aplicação "Instant Pot" desenvolvida utilizando o Ionic.

Figura 9 - Exemplo da aplicação "Instant Pot".



Fonte: <https://ionicframework.com/enterprise/customers> acesso em: 27 out. 2019.

2.3.1.6 Xamarin

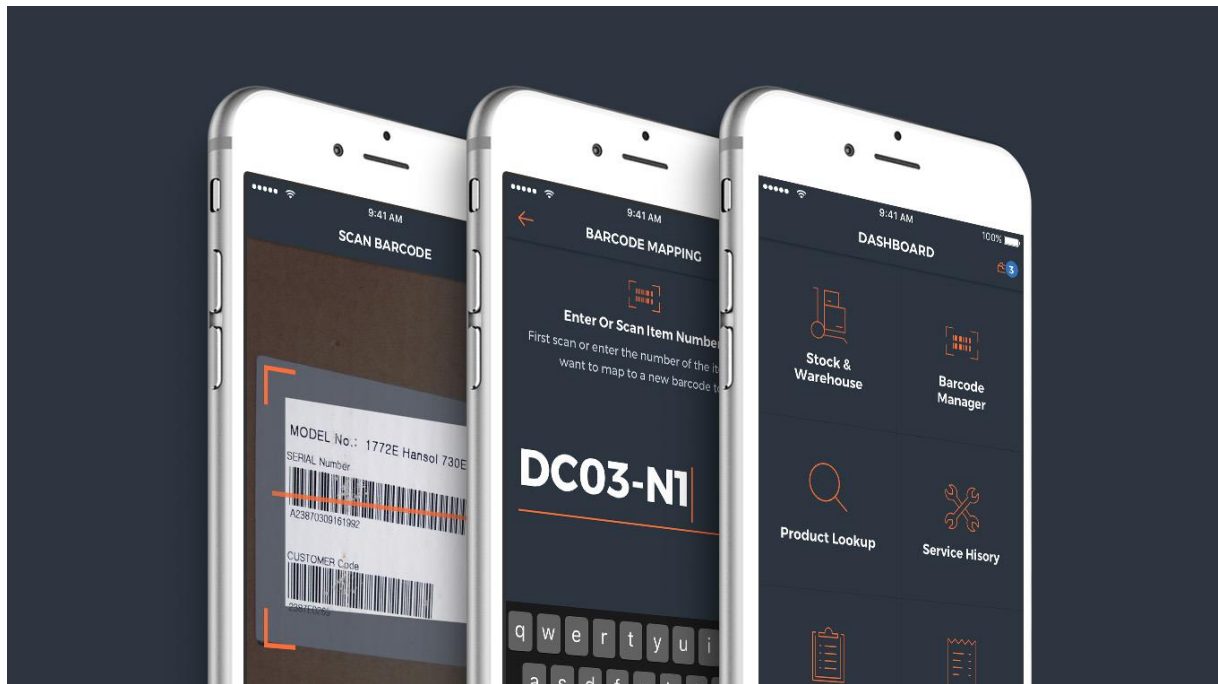
O Xamarin é um framework open-source para criar aplicativos IOS, Android e Windows utilizando a tecnologia “.NET”. Ele é ideal para desenvolvedores que queiram fazer aplicativos multiplataforma utilizando a linguagem C#, com o Visual Studio (XAMARIN, 2019).

A ferramenta combina as habilidades das plataformas nativas e adiciona vários recursos como por exemplo: Fornece recursos para chamar diretamente bibliotecas “Objective-C, Java ou C / C ++” e também permite utilizar o .NET BCL (Base class library) que se trata de uma grande coleção de classes que possuem recursos otimizados como suporte a “XML”, banco de dados e serialização (XAMARIN, 2019).

Ele ainda conta com o “Xamarin Essentials”, uma biblioteca que fornece APIs para recursos nativos dos dispositivos, como as informações do dispositivo, seu sistema de arquivo, bloqueio de tela, bateria, câmera e muitos outros.

A figura abaixo apresenta o aplicativo “Aggreko” e algumas de suas telas, que foram desenvolvidos utilizando o Xamarin.

Figura 10 - Exemplo da aplicação "Aggreko".



Fonte: <https://devblogs.microsoft.com/xamarin/customer-showcase-aggreko/> acesso em: 27 out. 2019.

2.3.1.7 React Native

O React Native é um framework desenvolvido pela Facebook para criar aplicativos Andoird ou iOS, combinando o melhor do desenvolvimento nativo com uma biblioteca de JavaScript chamada React. Ela também pode ser utilizada em aplicativos e projetos já criados (REACT NATIVE, 2019).

O usuário pode criar versões e componentes específicos para cada plataforma pois elas serão renderizadas na interface nativa de cada uma, fazendo com que o aplicativo tenha apenas um código base e seja compartilhado entre plataformas sem que seja comprometida a experiência dos usuários (REACT NATIVE, 2019).

Uma das maneiras mais simples de se utilizar e desenvolver com React Native, é usando o “Expo CLI”. Ele é um conjunto com várias ferramentas. Porém ao optar por essa abordagem o usuário precisará ter instalado uma versão recente de “Node.js” e de um dispositivo com a plataforma desejada. Caso queira experimentar diretamente do navegador antes é recomendável que se dê uma olhada na ferramenta Snack” <https://snack.expo.io/>” (REACT NATIVE, 2019).

Após instalar o expo no computador basta instalar o “Expo client app” no dispositivo iOS ou Android e conectar na mesma rede de internet que está conectado ao computador. Feito isso, caso seu sistema seja Android basta abrir o aplicativo no dispositivo, use o mesmo para

escanear o QR code que estará no computador e abrirá o projeto. Caso seja iOS basta seguir as instruções que aparecem na tela, para pegar o link para o projeto (REACT NATIVE, 2019).

O Expo CLI foi projetado para permitir que o usuário utilize do React Native sem precisar de um ambiente de desenvolvimento. Caso o usuário deseje utilizar um emulador aconselha-se que utilize configure o ambiente para ele utilizando o React Native CLI e instalando suas dependências e bibliotecas necessárias (REACT NATIVE, 2019).

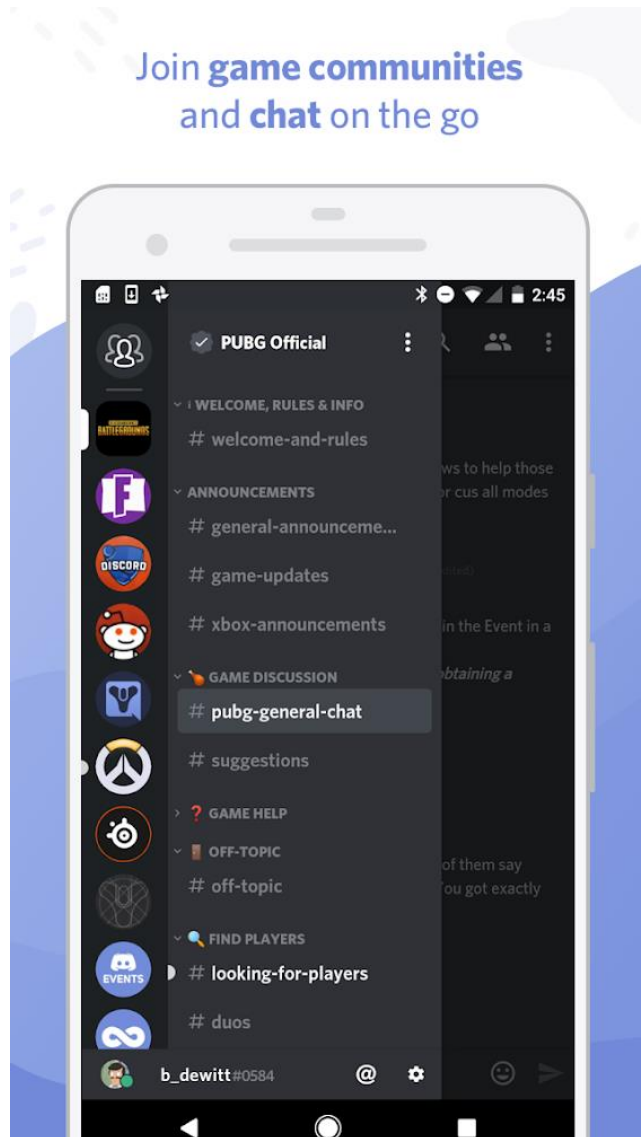
Outra característica que se destaca é o “Fast Refresh” ou atualização rápida, em português. Trata-se de um recurso que permite visualizar praticamente de imediato as mudanças nos componentes. Ela é ativa por padrão, mas pode ser desabilitada no menu do React Native. Com ela ativa, o usuário deve ter suas edições atualizadas alguns segundos após fazê-las (REACT NATIVE, 2019).

React Native foi lançado em 2015 e em 2018 teve o segundo maior número de contribuidores entre todos os repositórios do GitHub (REACT NATIVE, 2019).

React native é utilizado por dezenas de empresas e pessoas ao redor do mundo todo como por exemplo o próprio Facebook, o Instagram, Pinterest, Uber, Tesla, Discord, Walmart e muitas outras (REACT NATIVE, 2019).

A figura abaixo apresenta o aplicativo “Discord” na visão de um dispositivo Android e foi desenvolvido utilizando o React Native.

Figura 11 - Exemplo da aplicação “Discord”.



Fonte - <https://play.google.com/store/apps/details?id=com.discord> acesso em: 09 nov. 2019.

2.3.1.8 Flutter

O Flutter é um framework open-source da Google, para criar belas aplicações nativas para mobile, web e desktop, através de um único código. Ele funciona com códigos já existentes e tem sido utilizado por empresas ao redor do mundo (FLUTTER, 2019).

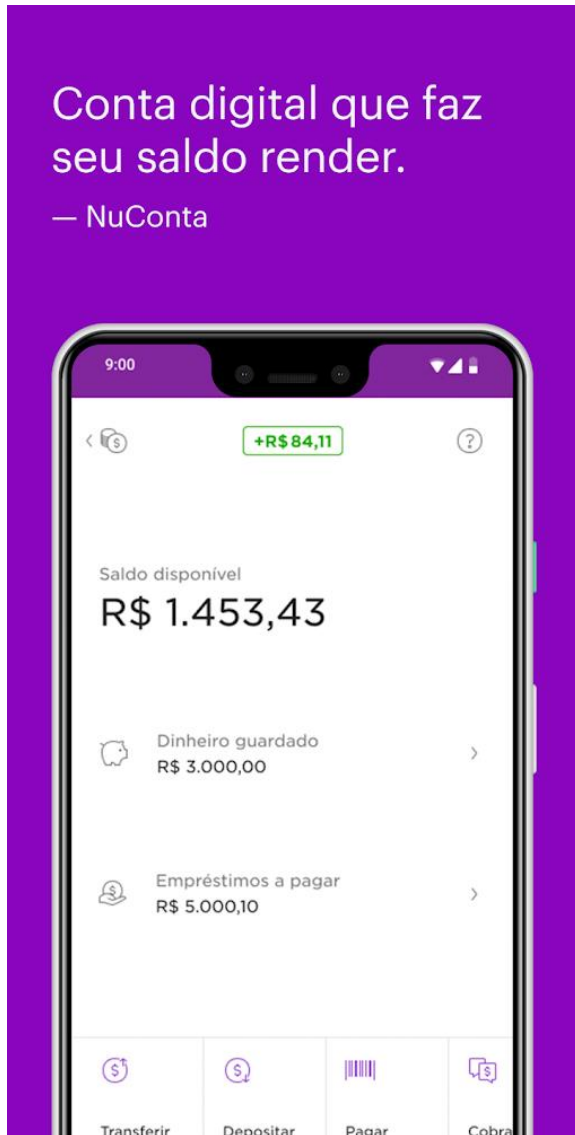
A ferramenta ainda promete, aos olhares dos designers, auxiliar a entregar a versão original da visão do designer, sem perdas de fidelidade ou comprometimento e atuando como uma produtiva ferramenta de prototipagem. E para os desenvolvedores, ela promete acelerar o desenvolvimento e reduzir a complexidade de produção nas plataformas (FLUTTER, 2019).

Algo que torna o Flutter um framework atrativo e que também é uma das suas principais características, é o fato que a ferramenta trabalha com seu próprio mecanismo de renderização

de alto desempenho para criar widgets e não utiliza os fornecidos com o aparelho. Ele também se difere por usar uma linguagem chamada “Dart” com uma camada de código em “C/C++” (FLUTTER, 2019).

A figura a seguir é um exemplo da aplicação do “NuBank” desenvolvida utilizando o Flutter

Figura 12 - Exemplo da aplicação do "NuBank".



Fonte: https://play.google.com/store/apps/details?id=com.nu.production&hl=pt_BR 09 nov. 2019.

2.4 CONSIDERAÇÕES FINAIS.

Para as considerações finais e melhor escolha do framework foram levados mais em consideração, alguns fatores do que outros. Estes fatores se encontram no quadro 1 que se encontra a seguir.

Quadro 1 - Quadro de comparação dos frameworks.

Framework	Plataforma	Linguagem	Suporte para Localização	Suporte para SMS
Apache Cordova	Multiplataforma	Html5, Css3 e JavaScript	Sim	Sim
Framework 7	Multiplataforma	Html, Css, JavaScript	Somente com plugins	Sim
jQuery Mobile	Multiplataforma	Html, Css e JQuery	Sim	Não
Adobe PhoneGap	Multiplataforma	Html, Css e JavaScript	Sim	Sim
Ionic	Multiplataforma	Html, Css e JavaScript	Sim	Sim
Xamarin	Multiplataforma	C#	Sim	Sim
React Native	Multiplataforma	React native	Sim	Sim
Flutter	Multiplataforma	Dart, C e C++	Sim	Sim

Fonte: Elaborado pelo autor (2019).

Porém ao observar esses fatores, como por exemplo a questão de suporte para localização, é perceptível que não é muita restrição e diferença entre todos eles. A maioria se diferencia em questão de linguagem ou questão de algumas características da empresa desenvolvedora ou de algumas funções que facilitam ainda mais o aprendizado e desenvolvimento utilizando a ferramenta, ficando assim por gosto pessoal a decisão de se utilizar a ferramenta React Native.

3 MÉTODO

Neste capítulo será abordado uma explicação mais detalhada do planejamento do desenvolvimento da aplicação para solucionar a problemática encontrada nos bares de karaokê bem como suas delimitações, escolha do framework a se trabalhar e detalhes do seu funcionamento.

Também será abordado quanto ao tipo de pesquisa realizada, métodos e técnicas que foram utilizados para realização.

3.1 CARACTERIZAÇÃO DO TIPO DE PESQUISA

Quanto a natureza da pesquisa realizada neste trabalho, segundo Silva e Menezes (2001), é considerada como uma pesquisa aplicada, pois está dirigida a solução de um problema e gera conhecimento para aplicação prática.

Também segundo Silva e Menezes (2001, p. 20) a abordagem retratada nesta pesquisa se encaixa na definição de uma pesquisa qualitativa pois “[...]considera que há uma relação dinâmica entre o mundo real e o sujeito, isto é, um vínculo indissociável entre o mundo objetivo e a subjetividade do sujeito que não pode ser traduzido em números.”

Em relação aos seus objetivos, esta pesquisa pode ser classificada como exploratória, que segundo Silva e Menezes (2001) pois se utiliza uma pesquisa bibliográfica para apresentar informações e melhor entendimento de como solucionar o problema apresentado.

E por fim, segundo Silva e Menezes (2001), pode ser considerada como uma pesquisa bibliográfica, por ser elaborada através de materiais publicados como alguns artigos, e principalmente material que se encontra disponibilizado na internet.

3.2 ETAPAS METODOLÓGICAS

As etapas metodológicas deste trabalho estão organizadas em:

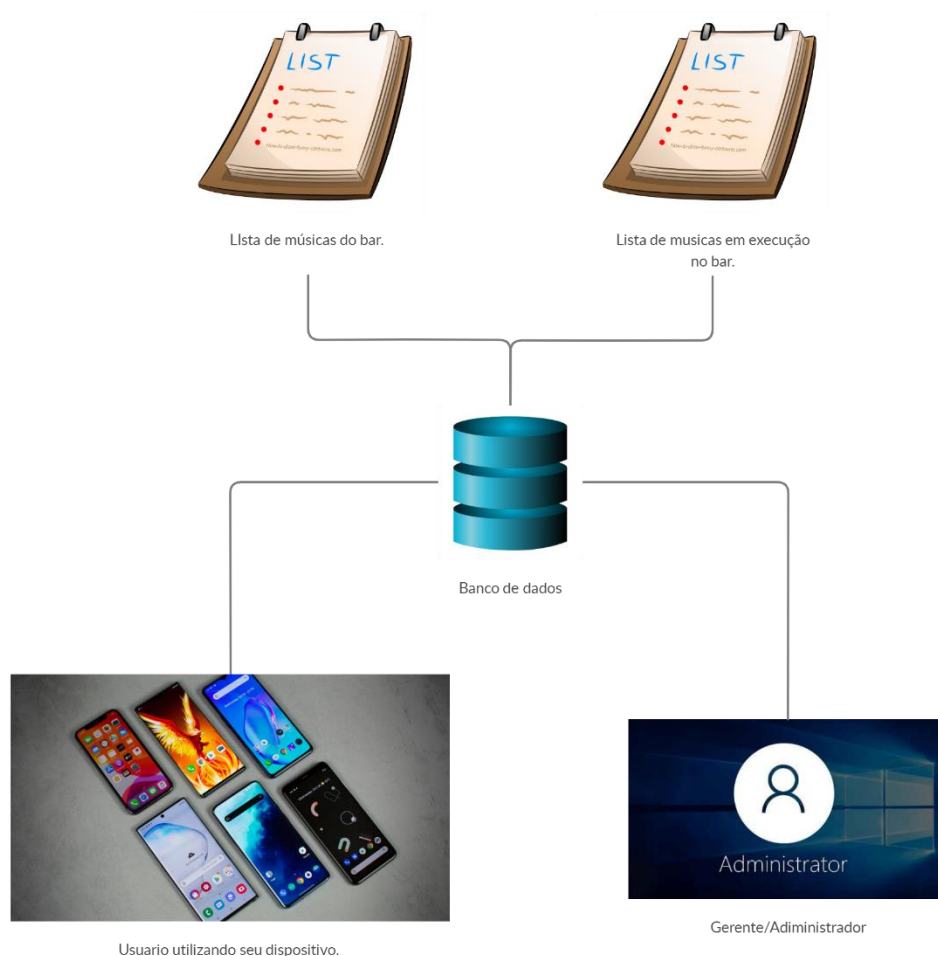
- a) Leitura, redação e revisão bibliográfica.
- b) Estudo e escolha do framework para auxiliar no desenvolvimento da aplicação para a solução da problemática.
- c) Aprendizado da tecnologia a ser utilizada
- d) Preparar o ambiente para o desenvolvimento
- e) Modelagem da aplicação

- f) Desenvolver protótipo da aplicação
- g) Testes da aplicação
- h) Elaboração dos resultados obtidos através dos testes

3.3 ARQUITETURA DA SOLUÇÃO PROPOSTA

A fim de atender os objetivos pretendidos, a solução proposta é um aplicativo com uma tecnologia híbrida, funcionando em ambos os sistemas, Android e iOS, que permita que o usuário consiga ter acesso a listagem de músicas do bar em que se encontra e adiciona-las na lista de músicas a ser executada, de maneira simples e pratica. A solução também permite que o administrador do bar tenha acesso a edição de todos os dados, seja da lista de músicas em execução ou do próprio acervo do seu bar. Abaixo segue uma figura onde é descrito a arquitetura do aplicativo.

Figura 13 - Arquitetura da solução proposta.



3.4 DELIMITAÇÕES

Este trabalho será realizado utilizando o framework React Native e visa a solução da problemática encontrada através de um aplicativo desenvolvido com o auxílio da ferramenta para que possa rodar tanto pelo browser dos aparelhos ser acessado via tal ou se o usuário preferir, funcione também instalando o aplicativo disponível na loja do dispositivo.

Os itens a seguir não são implementados na solução proposta:

- a) A implantação do sistema em um bar.
- b) Não será implantada nenhum tipo de requisito segurança para a aplicação.
- c) Não é função da aplicação armazenar e rodar as músicas. Estas devem ser feitas pelo sistema do bar que quiser implementar a solução, já utiliza.
- d) É necessária a conexão com a internet para executar as funções.
- e) Para a execução e apresentação deste trabalho, não será compilado e adicionado a loja dos dispositivos iOS.
- f) Não será disponibilizado o aplicativo na loja PlayStore do Android, pois para tal, há um custo na necessidade de um perfil de desenvolvedor para a plataforma.
- g) Não é foco deste aplicativo o tratamento de outras funções do bar como comanda, pedidos de comida e bebida e outros.

4 MODELAGEM

Este capítulo, apresenta as etapas da modelagem do sistema proposto, sendo enfatizados os requisitos do sistema, seus casos de uso principais os atores, suas regras de negócio e seus diagramas de classe.

Segundo Booch, Jacobson e Rumbaugh (2006) a modelagem é construída com diversos propósitos sendo alguns deles visualizar e controlar a arquitetura do sistema, compreender melhor o sistema e poder ter um melhor gerenciamento, se tornando então uma parte central das atividades que são necessárias para implantação de um bom software.

4.1 REQUISITOS

Booch, Jacobson e Rumbaugh (2006, p. 246) definem requisito como “[...] é uma característica do projeto, uma propriedade ou um comportamento de um sistema.”

Funcionando de maneira semelhante ao contrato, os requisitos servem para que se tenha uma boa visão das funcionalidades e as restrições que o sistema ou projeto terá, facilitando assim seu desenvolvimento (BOOC, JACOBSON e RUMBAUGH, 2006).

Estes requisitos geralmente estão divididos em duas partes e são conhecidos como requisitos funcionais e não funcionais.

4.1.1 Requisitos funcionais

O PMI (2017) define como requisito funcional aqueles que “[...] descrevem os comportamentos do produto. Exemplos incluem ações, processos e dados.”

Abaixo são apresentados os requisitos funcionais do sistema.

RF01 – O sistema deve permitir que o cliente acesse o bar específico.

Descrição – O cliente deve acessar o bar específico no qual ele se encontra ou deseja consultar as músicas, através de uma seleção quando for entrar no app/site.

Restrição – Necessária a escolha de um local para seguir a diante.

RF02 – O sistema deve permitir que o usuário autenticado tenha acesso a listagem das músicas do bar escolhido.

Descrição – Para se autenticar no sistema basta o usuário inserir o nome na tela inicial e selecionar o bar desejado.

Restrição – O sistema não deve permitir que um usuário use um nome já em uso.

RF03 – O sistema deve permitir que o usuário adicione uma música na lista de próximas a serem cantadas.

Descrição – O cliente deve acessar a listagem de músicas e adicionar uma, e somente uma, música a lista de execução, a fim de esperar sua vez de performar a música que ele escolheu.

Restrição – O sistema não deve permitir que seja adicionada mais do que uma música, até que ela seja performada e retirada da lista de músicas em execução.

RF04 – O sistema deve permitir a inserção de novas músicas pelo administrador.

Descrição – O sistema deve permitir que o administrador do sistema (dono do bar ou algum funcionário dele), adicionem músicas novas a listagem de músicas do bar.

Restrição – O id da música deve ser gerado automático de maneira aleatória.

RF05 – O sistema deve permitir que seja removida músicas da listagem de músicas.

Descrição – O sistema deve permitir que o administrador do sistema (dono do bar ou algum funcionário dele), remova uma ou mais músicas da listagem de músicas do bar.

Restrição – O sistema deve verificar através de uma mensagem de confirmação se o administrador deseja realmente remover a(s) música(s) selecionada(s).

RF06 – O sistema deve permitir que o administrador edite as músicas da listagem de músicas.

Descrição – O sistema deve permitir que o administrador do sistema (dono do bar ou algum funcionário dele), edite uma ou mais músicas da listagem de músicas do bar evitando assim de que seja necessário deletar e adicionar novamente caso algum erro ocorra.

RF07 – O sistema deve permitir que o administrador tenha total acesso as músicas a serem executadas.

Descrição – O administrador deve possuir acesso a lista de músicas em execução de forma que possa adicionar, editar ou remover músicas na lista.

4.1.2 Requisitos não funcionais

Como requisito não funcional o PMI (2017) define como sendo algo que “[...] complementam os requisitos funcionais e descrevem as condições requeridas para que o produto seja eficaz. Exemplos incluem: confiabilidade, proteção e desempenho.”

Abaixo são apresentados os requisitos não funcionais do sistema.

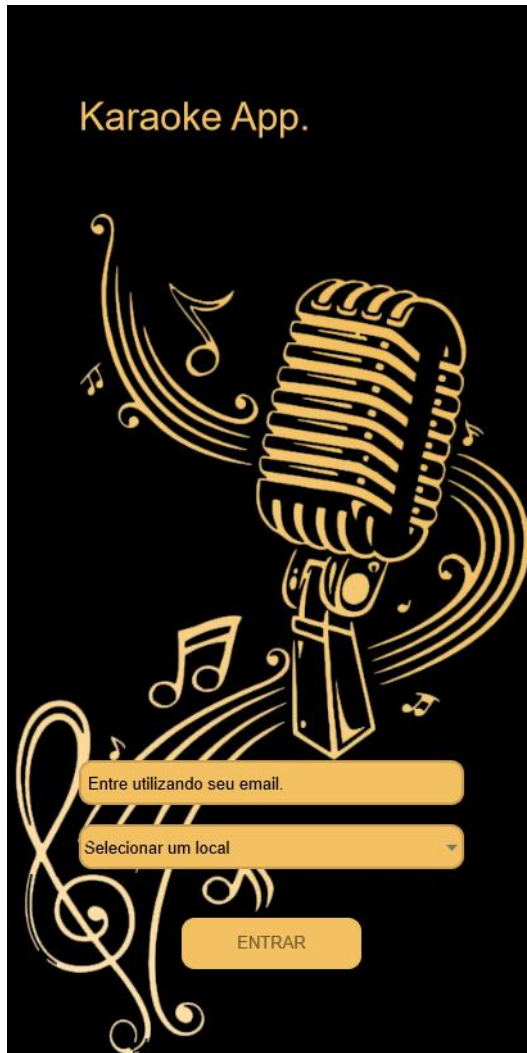
RNF01 – O sistema deverá ser desenvolvido em react native.

RNF02 – O sistema devera rodar em qualquer plataforma mobile

4.2 PROTOTIPAGEM

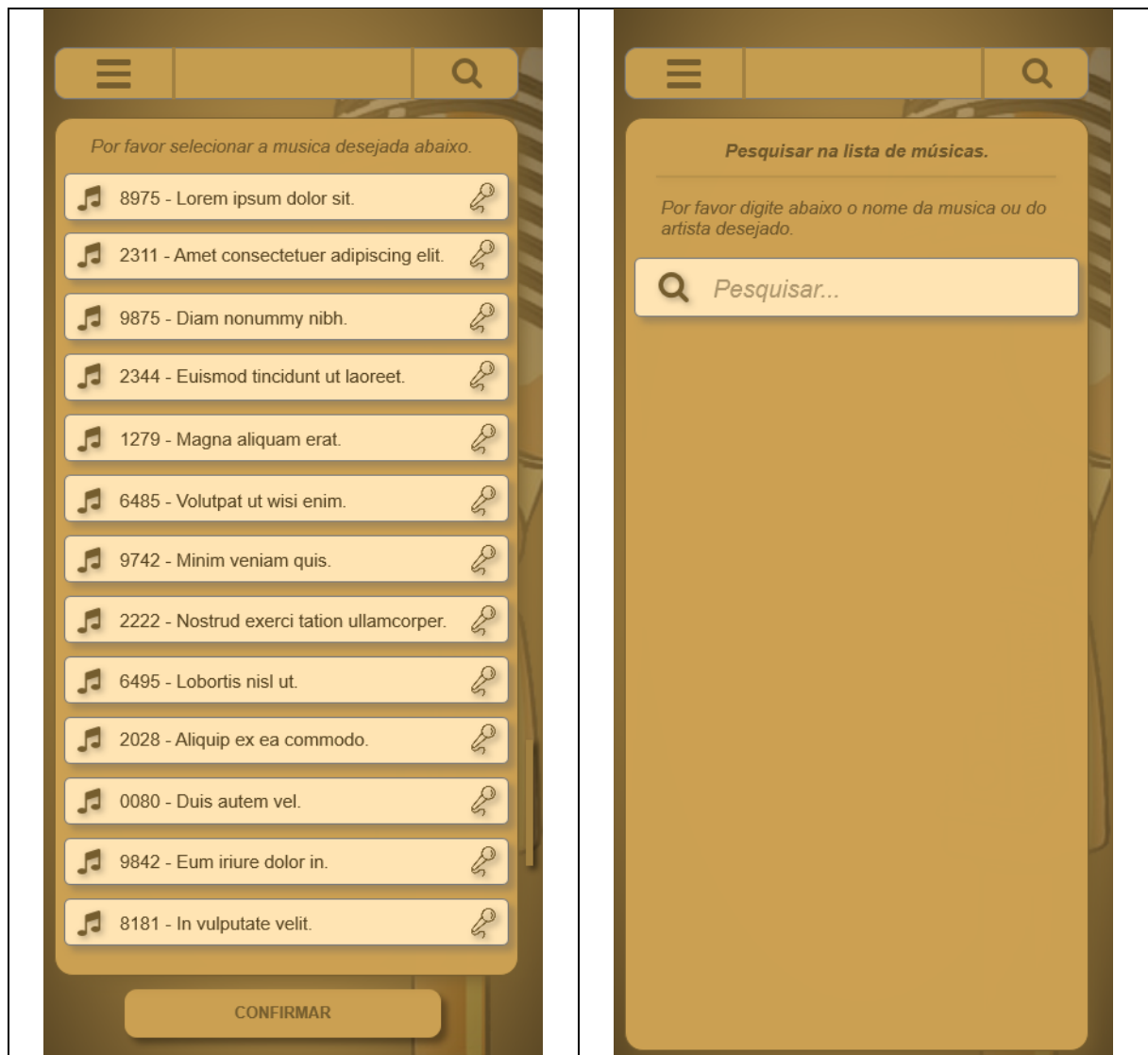
Para facilitar e agilizar o desenvolvimento foi utilizada uma ferramenta de prototipação na criação do modelo das telas, fazendo assim com que no desenvolvimento já se tenha uma base de como o sistema deverá ficar e se comportar. A primeira tela do sistema, é a de login onde o usuário entra com seu e-mail e seleciona o local desejado.

Figura 14 - Protótipo da tela de login



Uma vez logado no sistema, o usuário tem acesso a lista de músicas do bar selecionado e pode então realizar a escolha ou ir para a página de pesquisa procurar pela música desejada.

Quadro 2 – Protótipo das telas de lista de música e pesquisa respectivamente.

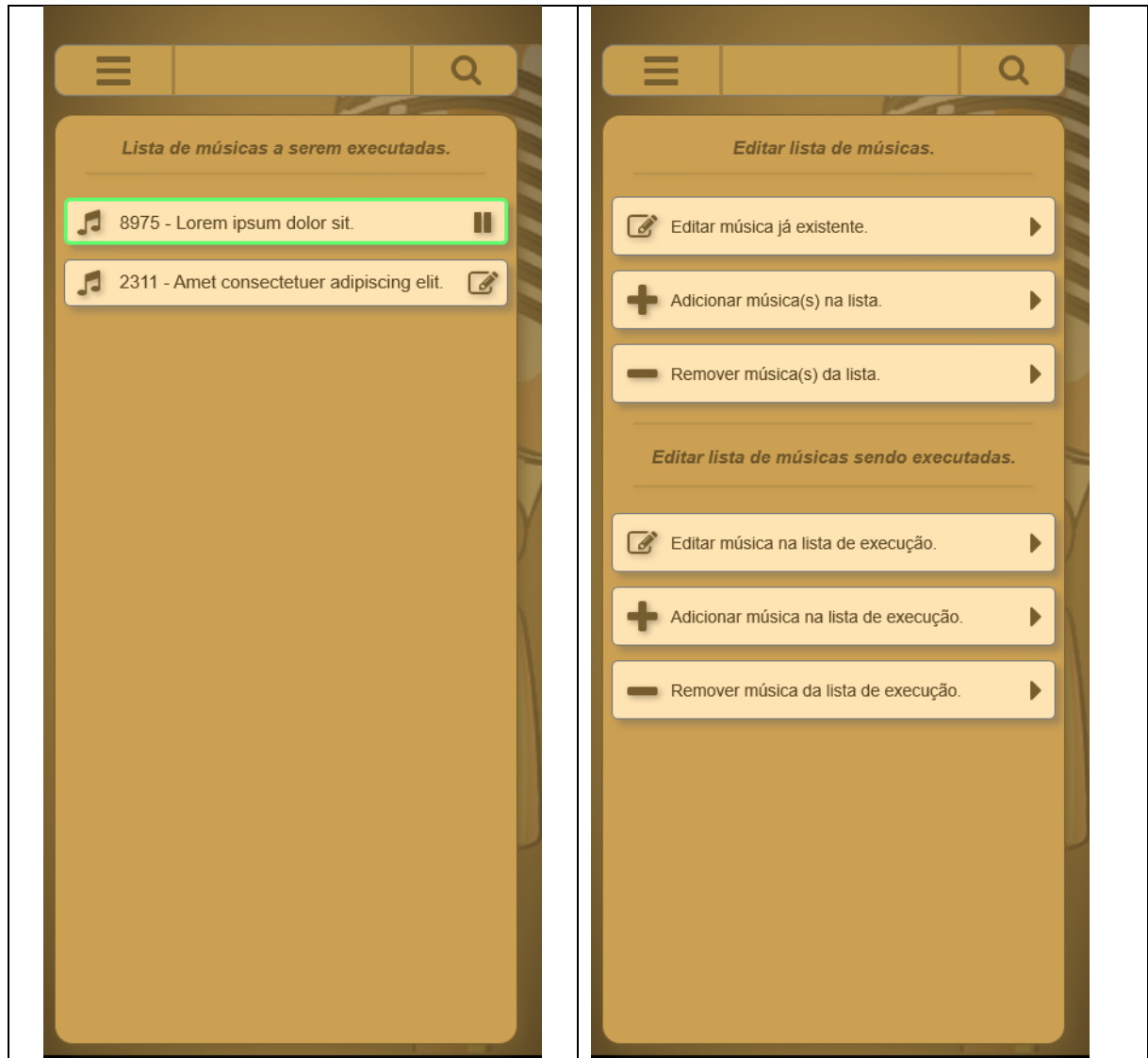


Fonte: Elaborado pelo autor (2020).

Caso o perfil não seja do usuário e sim do administrador do sistema, ele então teria acesso a outras duas telas, sendo elas a da lista de execução, e a tela de painel do administrador.

A tela de lista de execução serviria apenas para visualização da mesma e a de painel de administrador, para editar tanto a lista de execução, como a lista de acervo de músicas.

Quadro 3 - Protótipo das telas de lista de execução e painel de administrador.



Fonte: Elaborado pelo autor (2020).

4.3 CASOS DE USO

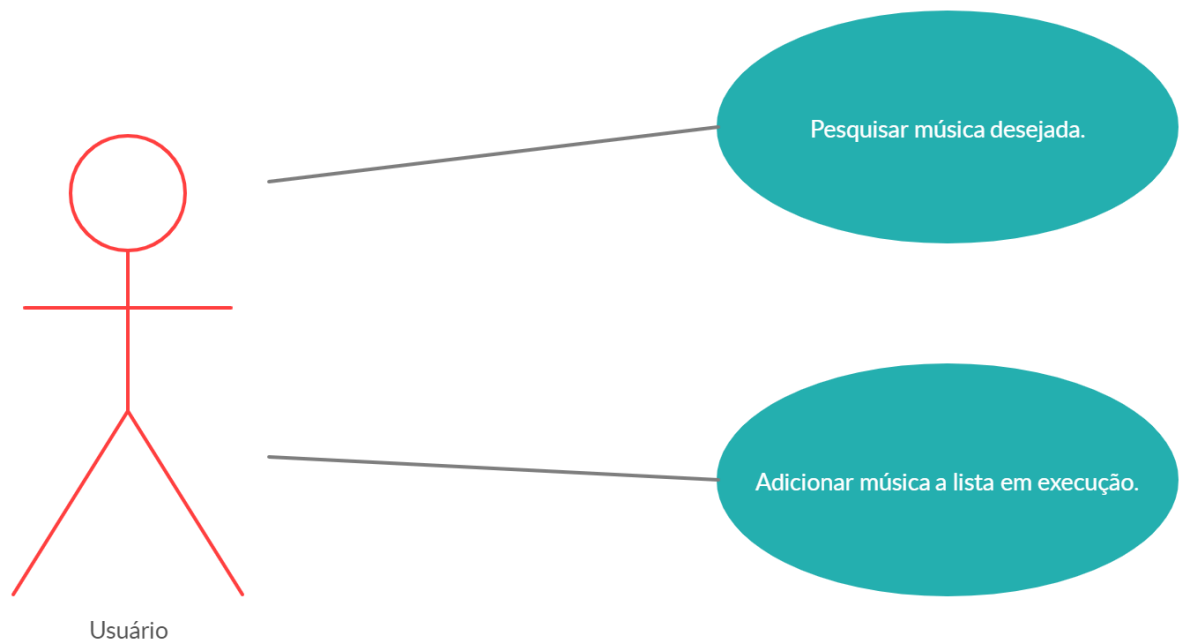
Casos de uso servem para auxiliar a determinar características e funcionalidades que o sistema irá possuir na visão do usuário. (PRESSMAN; MAXIM, 2016)

Pressman e Maxim (2016, p. 149) definem que ao escrever um caso de uso “[...] o primeiro passo é definir um conjunto de atores são as diferentes pessoas que usam o sistema no contexto da função e do comportamento a ser descrito. “

4.3.1 Casos de uso do sistema

A seguir tem-se os principais casos de uso do sistema.

Figura 15 - Diagrama de caso de uso - UC1 - Cadastrar música.

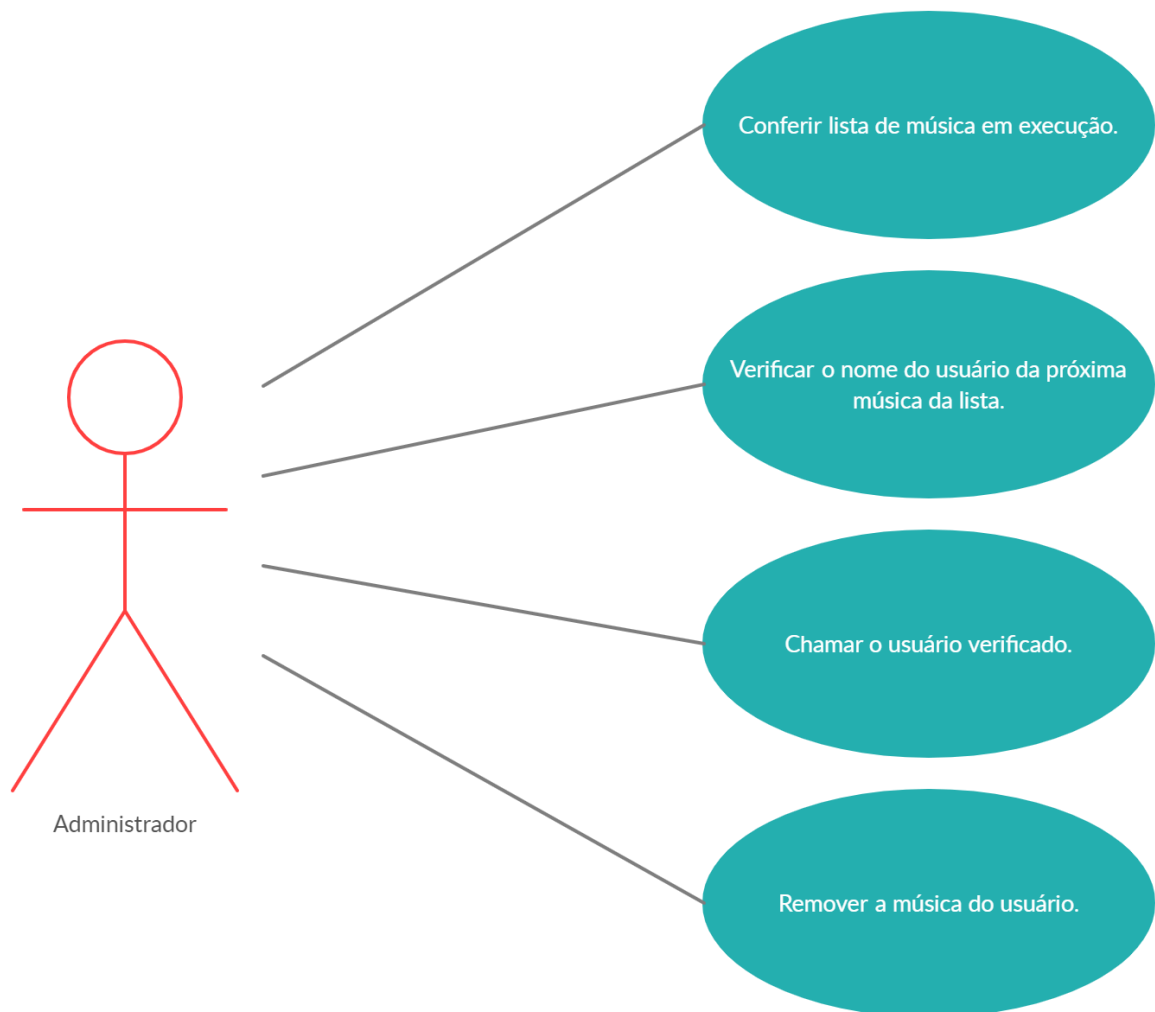


Fonte: Elaborado pelo autor (2020).

O usuário após acessar o sistema, pesquisa a música desejada, seja por nome, autor ou somente observando a listagem das músicas, e a adiciona na lista de músicas em execução afim de aguardar sua vez para realizar a apresentação.

Figura 16 - Diagrama de caso de uso - UC2 - Gerenciar a lista de execução

©



Fonte: Elaborado pelo autor (2020).

O Administrador do sistema, seja ele um funcionário ou o próprio proprietário do bar, possui a função de gerenciar a listagem de músicas e as pôr em execução. Sendo assim ele uma vez dentro do sistema, confere a lista de músicas em execução afim de saber qual será a próxima música, verifica o nome do usuário que cadastrou aquela respectiva música e chama o usuário. Após a música acabar, ele remove a música da lista de músicas em execução e repete o processo.

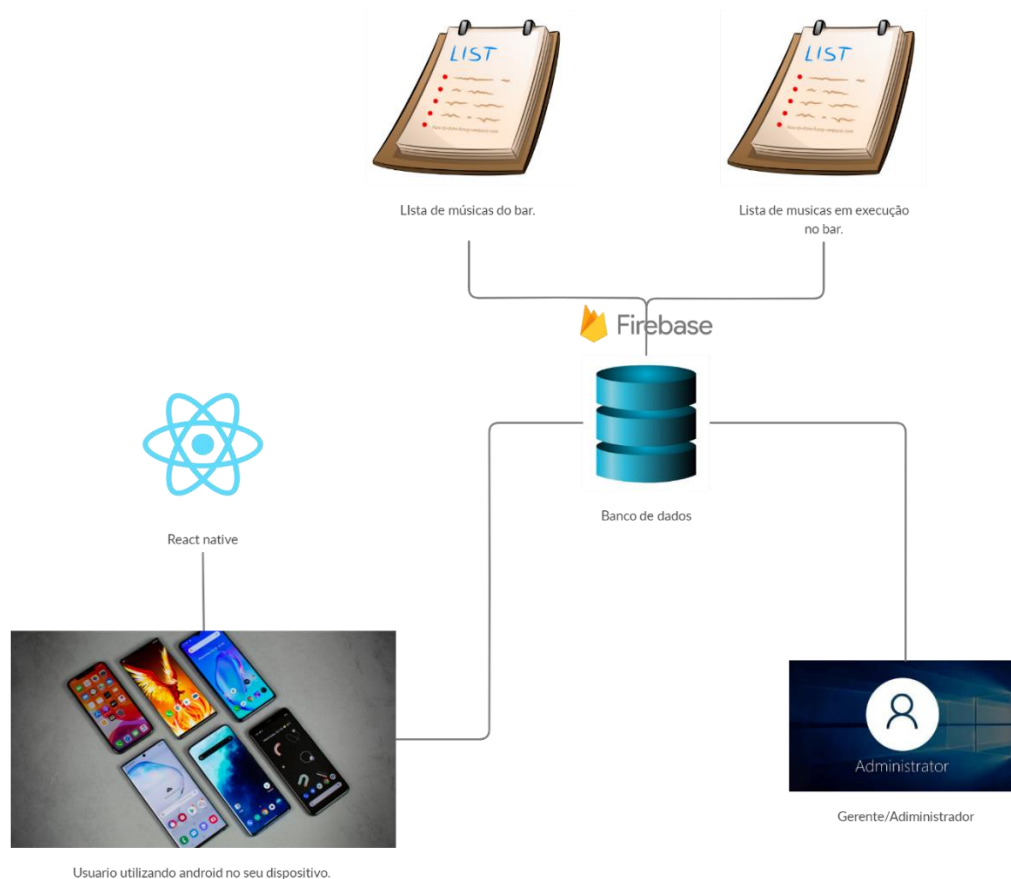
5 DESENVOLVIMENTO

Este capítulo apresenta o desenvolvimento da solução proposta, as tecnologias que foram utilizadas, a apresentação das telas e comportamentos do sistema e os resultados que foram obtidos com a implementação.

5.1 ARQUITETURA E TECNOLOGIAS

Abaixo são apresentadas as tecnologias utilizadas no processo de desenvolvimento da solução proposta e uma figura exemplificando a arquitetura do software.

Figura 17 - Arquitetura do sistema.



Fonte: Elaborado pelo autor (2020).

5.1.1 Visual Studio Code

Trata-se de um editor de código fonte muito leve e de fácil uso. Possui diversas plugins e extensões para diversos idiomas e diversas funções diferentes. Por padrão já vem com suporte para JavaScript, Typescript e Node.js. (VSCODE, 2020).

5.1.2 SDK Plataform-Tools

Trata-se de um componente do Android que traz ferramentas necessárias para o desenvolvimento de apps para a plataforma (SDK PLATAFORM-TOOLS, 2020).

5.1.3 Genymotion

Trata-se de um emulador para Android que possui uma grande variedade de modelos com um conjunto grande de recursos de interação com o ambiente virtual facilitando o desenvolvimento, testes e demonstrações. Apesar de ser um software pago, ele possui uma versão gratuita para uso pessoal (GENYMOTION, 2020).

5.1.4 JavaScript

Para Flanagan (2006), é uma linguagem de programação para Web que foi criada nos primórdios da mesma. É utilizada para definir o comportamento das páginas Web.

5.1.5 React Native

É uma linguagem que combina desenvolvimento nativo com React.js facilitando a criação de app's multiplataforma e nativos. Para utilizá-la é necessária a instalação de algumas dependências como por exemplo: Python e Node.js. O react native permite trabalhar com funções em JavaScript, com componentes de React.js e elementos nativos tudo ao mesmo tempo e de maneira que rode em ambas as plataformas (Android e IOS).

5.1.6 Firebase Real Time Database

É um banco de dados NoSQL hospedado em nuvem que trabalha com sincronização de dados em formato JSON em tempo real. (FIREBASE, 2020)

5.1.7 React Native Paper

Trata-se de uma biblioteca de componentes personalizados e customizáveis de fácil uso para React Native (React native paper, 2020).

5.1.8 React Navigation

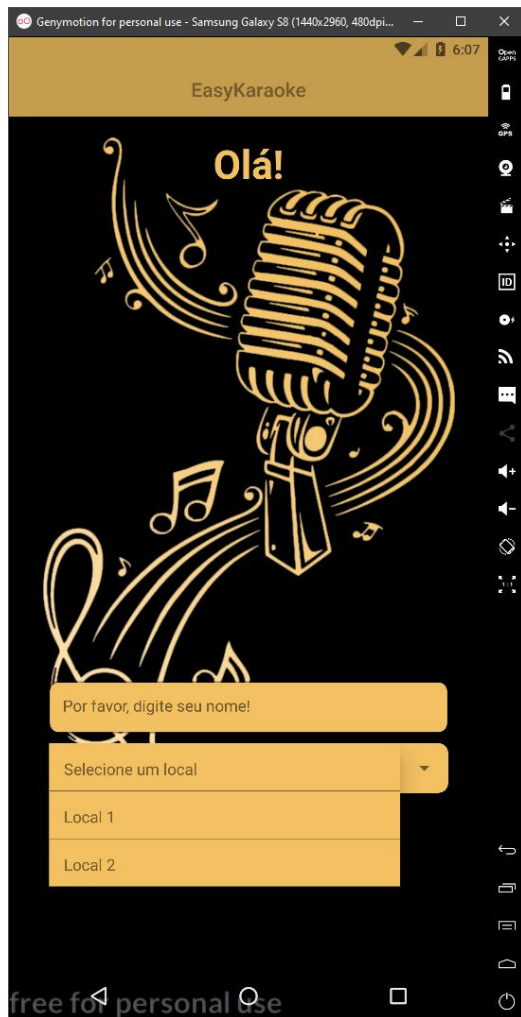
É uma biblioteca para trabalhar com roteamento e navegação entre telas de diversas formas diferentes e personalizáveis (React navigation, 2020).

5.2 APRESENTAÇÃO DO PROTOTIPO

Nesta seção, o sistema móvel é apresentado e é possível observar todas as telas do mesmo.

O aplicativo móvel traz uma tela principal simples que solicita a entrada do nome do usuário e a escolha do local como mostra na figura abaixo.

Figura 18 - Tela de entrada do aplicativo.



Fonte: Elaborado pelo autor (2020).

Uma vez feito isso, o usuário terá acesso a listagem de músicas como mostrado na figura 19 logo abaixo.

Figura 19 - Tela de lista de músicas.



Figura 20 - Tela de pesquisa.



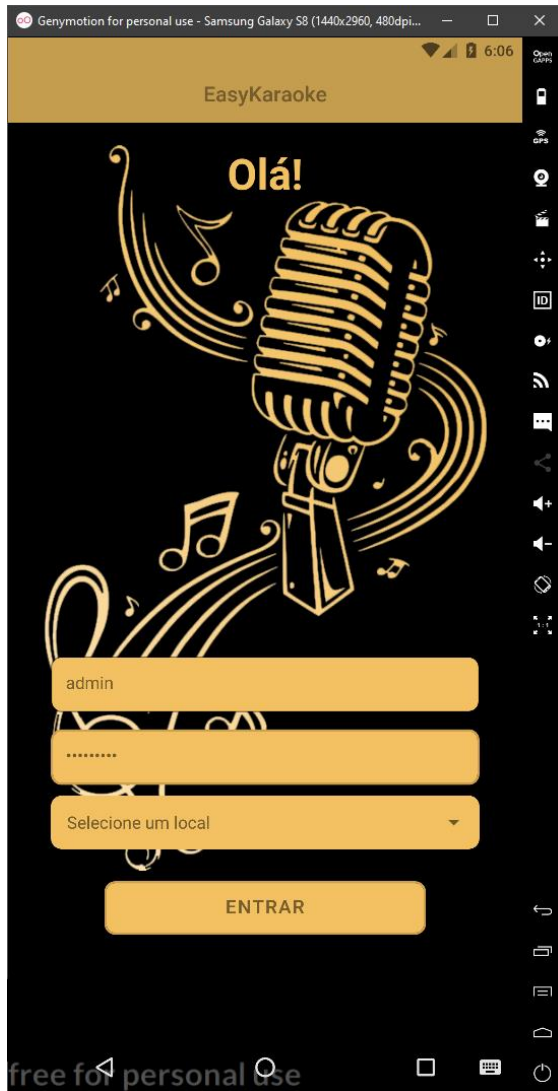
Fonte: Elaborado pelo autor (2020).

Uma vez que o usuário se encontre nesta tela basta selecionar a música que seja clicando em cima do container dela. Caso o usuário prefira, ele pode optar por pesquisar a música desejada diretamente utilizando a tela de pesquisa, como mostrado com a figura 20.

O usuário também poderá optar por ordenar sua pesquisa em ordem alfabética, por nome do artista ou por nome da música sendo o padrão da ordem por nome da música.

Caso não se trate do usuário, e sim do administrador do sistema, ele deverá entrar com o nome de administrador (Inicialmente escolhido como admin) clicar em entrar, que irá aparecer um campo para o mesmo digitar a senha do sistema como mostra a figura abaixo.

Figura 21 - Tela de entrada do administrador.



Fonte: Elaborado pelo autor (2020).

Após digitar a senha e selecionar o local, basta clicar em entrar que ele irá realizar o login como administrador e terá acesso ao sistema, com algumas funcionalidades que os usuários não têm acesso e não possuem visualização assim como consta na figura abaixo.

Figura 22 - Lista de músicas pelo perfil de admin.



Figura 23 - Tela de inserção de música nova.



Fonte: Elaborado pelo autor (2020).

Na visão do administrador é possível adicionar músicas a lista, remover ou editar as já existentes caso haja necessidade.

Para adicionar basta clicar no símbolo de mais na parte inferior da tela, que abrirá uma janela aonde o administrador deve entrar com o nome do artista e o nome da música que deseja inserir como mostra na figura 23.

Caso ele queira remover ou editar alguma música da lista, ele deverá pressionar a música que deseja e segurar por dois ou mais segundos, abrindo um menu aonde poderá ver e escolher uma destas opções como mostrado abaixo.

Figura 24 - Menu de editar ou remover músicas.



Figura 25 - Menu de edição de música.



Fonte: Elaborado pelo autor (2020).

Caso ele queira remover, basta clicar no botão “remover” e automaticamente excluirá e organizará todos os demais elementos presentes em tela.

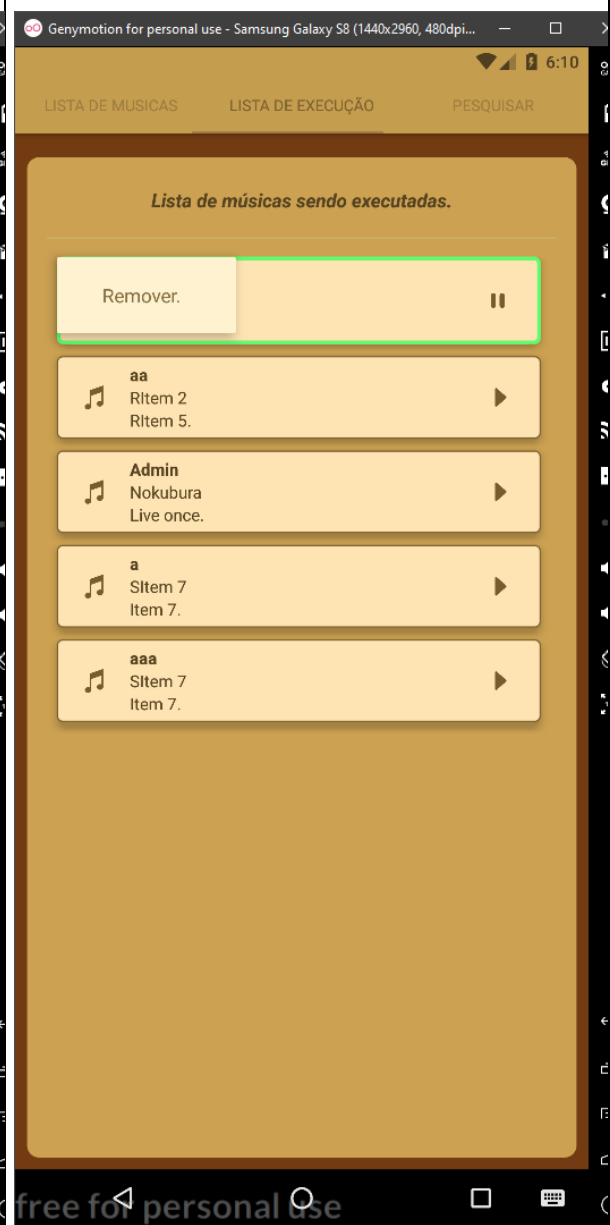
Caso clique em editar, ele verá uma tela semelhante à de adicionar, porém com os dados da música que ele selecionou para editar, já carregado nos campos, facilitando a edição assim como mostrado no exemplo abaixo.

Por último e não menos importante temos a tela da lista de músicas em execução, onde traz a informação do nome do usuário, e as informações da música que o usuário selecionou como mostra na figura 26, logo abaixo.

Figura 26 - Lista de músicas em execução.



Figura 27 - Remover da lista de execução.



Fonte: Elaborado pelo autor (2020).

Após executar a música o administrador deverá fazer o mesmo processo de segurar por dois ou mais segundos em cima da música afim de abrir um menu e remover ela da lista.

Apresentado o sistema por completo, abaixo será mostrada a avaliação do sistema.

5.3 VALIDAÇÃO DO PROTÓTIPO

A validação do protótipo ocorreu através de um questionário em que foi solicitado a alguns usuários, que realizassem o login e a inserção de uma música na lista de execução e após isto o respondesse.

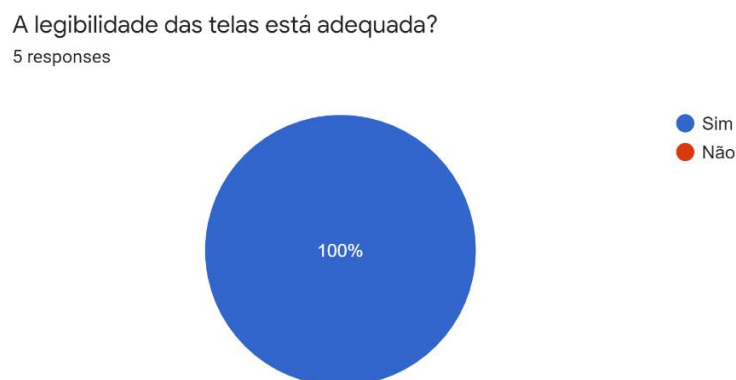
O questionaria foi elaborado com o objetivo de analisar e avaliar a experiencia de alguns usuários que frequentam tais tipos de bares.

5.3.1 Análise dos resultados obtidos

O questionário possuía um prazo para sua realização e dentro do mesmo, foram obtidas cinco respostas.

Para a primeira pergunta, foi abordado o quesito de legibilidade das telas do sistema. Os resultados foram extremamente positivos como é mostrado no gráfico abaixo.

Gráficos 1 - Legibilidade das telas.

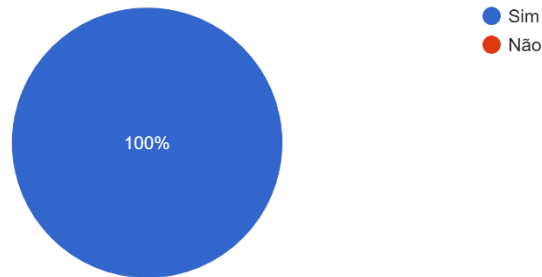


Fonte: Elaborado pelo autor (2020).

Logo após foi abordado quanto aos títulos e rótulos das telas e dos botões. Novamente os resultados foram extremamente positivos como pode ser observado no gráfico a seguir.

Gráficos 2 – Títulos de botões e telas.

Os títulos e rótulos das telas em campos e botões são significativos?
5 responses

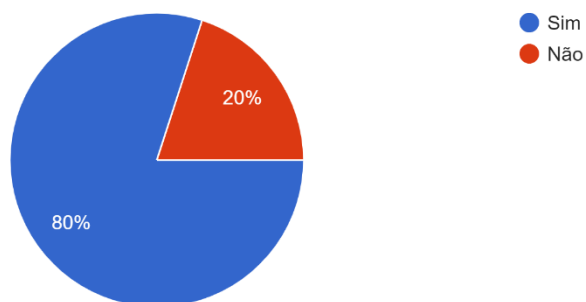


Fonte: Elaborado pelo autor (2020).

Em relação ao feedback das ações, ao clicar nos botões foi notado que o aproveitamento não foi extremamente positivo, porém possui uma boa aceitação. Mais detalhes no gráfico abaixo.

Gráficos 3 - Feedback das ações.

O feedback das ações ao clicar nos botões apoiou sua interação?
5 responses

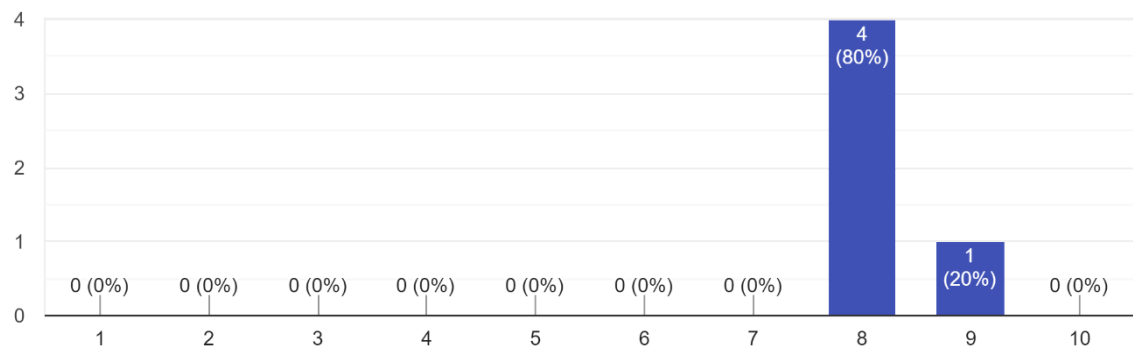


Fonte: Elaborado pelo autor (2020).

Após isso foi feita uma avaliação que ia de 0 a 10, com relação a nota para as telas do aplicativo e sua usabilidade. Os resultados podem ser visualizados com o gráfico abaixo.

Gráficos 4 - Avaliação das telas.

2. Avaliando as telas do aplicativo você considera que a usabilidade possa ser avaliada com a nota:
5 responses

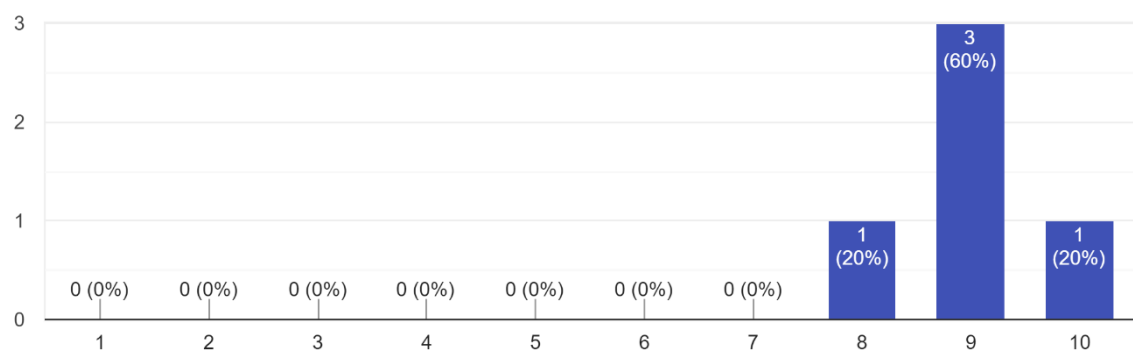


Fonte: Elaborado pelo autor (2020).

Uma outra avaliação de 0 a 10 foi realizada, analisando todas as telas e questionando se as funcionalidades apresentadas pelo aplicativo atendiam as necessidades do usuário. Como pode notar no gráfico abaixo as respostas foram muito positivas.

Gráficos 5 - Avaliação das necessidades atendidas.

3. Avaliando as telas do aplicativo você considera que as funcionalidades do aplicativo atendem suas necessidades, sendo avaliado com a nota:
5 responses



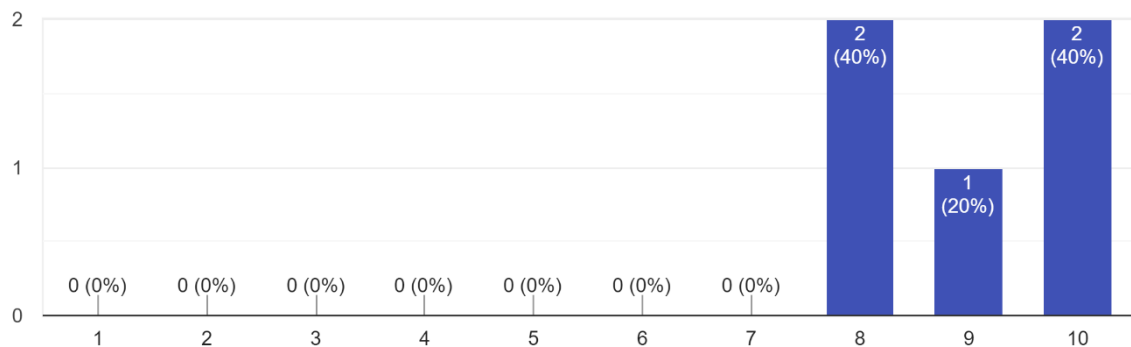
Fonte: Elaborado pelo autor (2020).

Outro ponto importante que foi abordado, foi em relação ao tempo de resposta do aplicativo nas funcionalidades apresentadas e executadas. Ao analisar os resultados foi verificado uma grande satisfação a respeito.

Gráficos 6 - Avaliação do tempo de resposta.

4. Avaliando as telas do aplicativo você considera que o tempo de resposta do aplicativo atendem suas necessidades, sendo avaliado com a nota:

5 responses



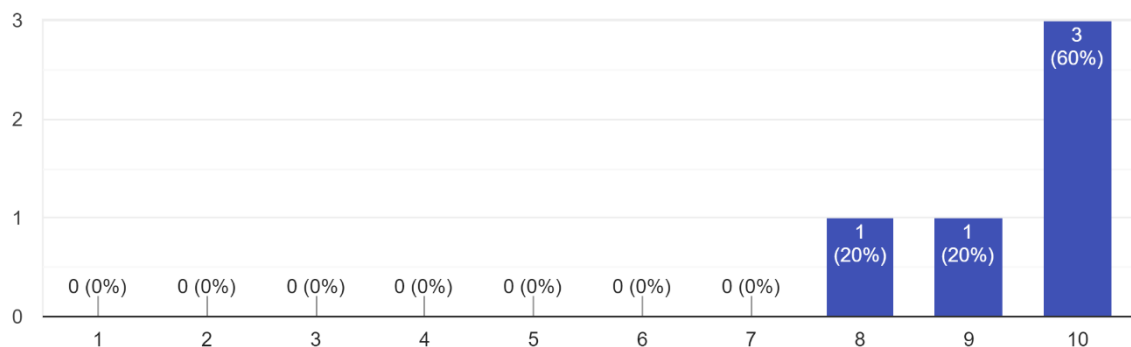
Fonte: Elaborado pelo autor (2020).

Em relação a confortabilidade do usuário ao utilizar o aplicativo, as respostas também foram muito positivas.

Gráficos 7 - Avaliação do conforto do usuário.

5. Você se sentiu confortável ao utilizar o aplicativo?

5 responses



Fonte: Elaborado pelo autor (2020).

Por último foram abordadas perguntas com relação a sugestões de melhorias e pontos positivos do aplicativo.

Vale destacar que em relação a sugestão de melhorias para o sistema, foram obtidas algumas respostas já esperadas e algumas novas que serão abordadas a seguir.

5.4 CONCLUSÕES DO PROTÓTIPO

Considera-se que a recepção do aplicativo, perante o perfil dos usuários e as respostas obtidas, foi muito boa e que de maneira geral atendeu muito bem as expectativas e necessidades dos usuários.

Foram dadas algumas sugestões de melhorias como por exemplo: Exibir a lista de execução para o usuário ressaltando sua posição na fila e algumas modificações no tema (escolha das cores, background e mais personalização para o aplicativo).

Em relação aos pontos positivos do aplicativo alguns usuários destacaram que: “A responsividade é ótima e as funcionalidades são satisfatórias”, o fato de facilitar em muito a gestão do bar, o filtro de pesquisa e a praticidade que o aplicativo traz com as escolhas da música de maneira tão simples e rápida.

6 CONCLUSÃO E TRABALHOS FUTUROS.

6.1 CONCLUSÕES

Ao final do trabalho apresentado, pode-se concluir que a proposta, que era de trazer uma maior comodidade e praticidades aos diversos clientes de barzinhos de karaokê e fazer com que os bares em questão tivessem um atrativo a mais tendo algo diferenciado que fosse levado em consideração por um cliente na escolha de um lugar para ir ou frequentar, foi atendida com sucesso.

Para o desenvolvimento do protótipo foram avaliados e cogitados diversos frameworks, onde a ideia principal era um sistema multiplataforma, para atender todos os usuários possíveis, e de fácil implementação e aprendizagem. Também foi levado em consideração na escolha do framework, sua relação com o mercado de trabalho e aceitação no mesmo, onde o React Native se mostrou uma boa escolha tendo em vista a quantidade de empresas que o utilizam e a quantidade e qualidade de material disponível para o aprendizado.

No início houve alguns problemas com a configuração do ambiente, porem foram solucionados rapidamente e no desenvolvimento também ocorreram alguns pequenos erros inesperados, porém mais uma vez o framework se mostrou a escolha certa devido ao grande volume de material de suporte tanto por parte da empresa quanto por parte da comunidade que o utiliza.

Sendo assim o aplicativo possui um código para ambas as plataformas Android e iOS, e possui as funcionalidades desejadas para atender os objetivos do trabalho de maneira simples e muito eficiente. Essa eficiência pode ser vista através dos testes nos quais o protótipo foi submetido.

Quando realizado os testes e aplicado o questionário a respeito das funcionalidades, expectativas e legibilidade, as respostas dos usuários que testaram o aplicativo, foram extremamente positivas.

6.2 TRABALHOS FUTUROS

Em relação a trabalhos futuros pretende-se criar perfis de usuário com e-mail e senha para armazená-los no sistema, a fim de se ter uma aba de favoritos, sugestões de acordo com o

que a pessoa geralmente escolhe através de tags de artista ou estilo musical, quantidade de vezes que a pessoa escolheu as músicas, histórico de escolha e melhorias na segurança do aplicativo.

Também assim como alguns usuários sugeriram no questionário, possuir uma forma de pré-visualização das músicas, o usuário possuir acesso a tela de músicas em execução para saber qual sua posição na fila, um visualizador de tempo estimado até sua vez, melhorias no layout do sistema e possuir alguma maneira do usuário sugerir músicas para o bar inserir no seu acervo.

Outro ponto a se ressaltar é a ideia de trazer este aplicativo para web, a fim de facilitar para os usuários que não desejarem instalar o aplicativo e para os próprios administradores acessarem o sistema de maneira mais simples, através do computador ao invés do smartphone.

REFERÊNCIAS

APACHE. Disponível em: <https://www.apache.org/foundation/>. Acesso em: 26 out. 2019.

BERTAGNOLLI, Silvia de Castro; MILETTO, Evandro Manara. **Desenvolvimento de Software II: Introdução ao Desenvolvimento Web com HTML, CSS, JavaScript e PHP - Eixo: Informação e Comunicação - Série Tekne**. 2014. Disponível em: https://books.google.com.br/books?id=lcLFAwAAQBAJ&dq=Desenvolvimento+web&lr=&hl=pt-BR&source=gbp_navlinks_s Acesso em: 17 nov. 2019.

BOOCH, Grady; JACOBSON Ivar; RUMBAUGH James. **UML: Guia do usuário**. 2006. Disponível em: https://books.google.com.br/books?id=ddWqxcDKGF8C&hl=pt-BR&source=gbp_navlinks_s Acesso em: 15 maio. 2020.

CORDOVA, Apache. Disponível em: <https://cordova.apache.org/>. Acesso em: 26 out. 2019.

COSTA, Carlos J. **Desenvolvimento para Web**. Disponível em: https://books.google.com.br/books?id=Jn6dTDF-wcsC&dq=Desenvolvimento+web&lr=&hl=pt-BR&source=gbp_navlinks_s. Acesso em: 17 nov. 2019.

FIREBASE, Google. Disponível em: <https://firebase.google.com/products/realtime-database>. Acesso em: 08 jun. 2020.

FLANAGAN, David. **JAVASCRIPT: The definitive guide, fifth edition**. Disponível em: https://books.google.com.br/books?id=k0CbAgAAQBAJ&dq=Javascript&lr=&hl=pt-BR&source=gbp_navlinks_s. Acesso em: 09 jun. 2020.

FLATSCHART, Fábio. **HTML 5 - Embarque Imediato**. Disponível em: https://books.google.com.br/books?id=_cgsCgAAQBAJ&dq=html+css+e+javascript&lr=lan_g_pt&hl=pt-BR&source=gbp_navlinks_s. Acesso em: 19 nov. 2019.

FLUTTER, Google. Disponível em <https://flutter.dev/docs>. Acesso em: 09 nov. 2019.

FRAMEWORK 7. Disponível em <https://framework7.io/docs/introduction.html>. Acesso em: 26 out. 2019.

GABRIEL, Martha. **Marketing na Era Digital: Conceitos, Plataformas e Estratégias**. 2010. Disponível em: https://books.google.com.br/books?id=Cp2YDQAAQBAJ&hl=pt-BR&source=gbp_navlinks_s. Acesso em: 17 nov. 2019.

IONIC. Disponível em <https://ionicframework.com/docs/>. Acesso em: 27 out. 2019.

JQUERY MOBILE. Disponível em <https://api.jquerymobile.com/>. Acesso em: 26 out. 2019.

MORIMOTO, Carlos Eduardo. **Smartphones, Guia Prático**. 2009. Disponível em: <https://www.hardware.com.br/livros/smartphones/>. Acesso em: 17 nov. 2019.

PEREIRA, Lucio Camilo Oliva; SILVA, Michel Lourenço da. **Android para Desenvolvedores**. 2009. Disponível em:

<https://books.google.com.br/books?id=8u9wJowXfdUC&dq=Android&lr=&hl=pt-BR&source=gbs_navlinks_s>. Acesso em: 17 nov. 2009.

PHONEGAP, Adobe. Disponível em: <http://docs.phonegap.com/>. Acesso em: 26 out. 2019.

PMI. **Minha biblioteca Meu histórico Livros no Google Play A Guide to the Project Management Body of Knowledge (PMBOK® Guide) - Sixth Edition (BRAZILIAN PORTUGUESE)**. 2017. Disponível em:

https://books.google.com.br/books?id=4IXDDwAAQBAJ&dq=pmbok+requisitos+funcionais&hl=pt-BR&source=gbs_navlinks_s. Acesso em: 15 maio 2020.

PRESSMAN, Roger; MAXIM, Bruce. **Engenharia de Software - 8ª Edição**. 2016. Disponível em:

<https://books.google.com.br/books?id=wexzCwAAQBAJ&dq=Engenharia+de+software+Framework&lr=lang_pt&hl=pt-BR&source=gbs_navlinks_s>. Acesso em: 18 nov. 2019.

PREZOTTO, Ezequiel Douglas; BONIATI, Bruno Batista. **Anais do EATI - Encontro Anual de Tecnologia da Informação e Semana Acadêmica de Tecnologia da Informação: Estudo de Frameworks Multiplataforma Para Desenvolvimento de Aplicações Mobile Híbridas**. 2014. Disponível em: <<http://www.eati.info/eati/2014/assets/anais/artigo8.pdf>>. Acesso em: 18 nov. 2019.

REACT NATIVE, Facebook. Disponível em: <https://facebook.github.io/react-native/>. Acesso em: 09 nov. 2019.

REACT NATIVE PAPER. Disponível em: <https://callstack.github.io/react-native-paper/>. Acesso em: 08 jun. 2020.

REACT NAVIGATION. Disponível em: <https://reactnavigation.org/>. Acesso em: 08 jun. 2020.

SDK PLATAFORM-TOOLS, Google. Disponível em: <https://developer.android.com/studio/releases/platform-tools>. Acesso em: 09 jun. 2020.

TEIXEIRA, José Ricardo. **Introdução ao framework jQuery Mobile**. 2013. Disponível em <https://www.devmedia.com.br/introducao-ao-framework-jquery-mobile/27213>. Acesso em: 26 out. 2019.

VISUAL STUDIO CODE, Microsoft. Disponível em: <https://code.visualstudio.com/docs>. Acesso em: 08 jun. 2020.

XAMARIN, Microsoft. Disponível em <https://docs.microsoft.com/en-us/xamarin/>. Acesso em: 27 out. 2019.