



Aplicativo para Otimização de Mobilidade Interurbana

Modelagem, Implementação e Prática

Eduardo Machado Lopes

JUIZ DE FORA
JULHO, 2018

Aplicativo para Otimização de Mobilidade Interurbana

Modelagem, Implementação e Prática

EDUARDO MACHADO LOPES

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Orientador: Igor de Oliveira Knop

JUIZ DE FORA
JULHO, 2018

APLICATIVO PARA OTIMIZAÇÃO DE MOBILIDADE
INTERURBANA

Modelagem, Implementação e Prática

Eduardo Machado Lopes

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Igor de Oliveira Knop
D.Sc. em Modelagem Computacional

Stênio Sã Rosário Furtado Soares
D.Sc. em Computação

Luiz Felipe Carvalho Mendes
M.Sc. em Modelagem Computacional

JUIZ DE FORA
09 DE JULHO, 2018

Resumo

O Brasil é um dos países mais caros para se ter um veículo próprio, considerando os gastos com a compra e manutenção do bem. Porém, com a precariedade do transporte público, o transporte privado continua sendo uma das opções mais utilizadas pelos brasileiros. A carona solidária tem sido uma das alternativas utilizadas para quem necessita se locomover frequentemente entre cidades vizinhas. O contato é feito principalmente utilizando-se dos grupos criados em aplicativos de troca de mensagens instantâneas para divulgar os interesses de oferta e procura. Entretanto, esta abordagem apresenta limitações, como por exemplo, existe número máximo de participantes, fraca organização dos anúncios postados, baixo nível de confiança entre os integrantes, falta de controle sobre os assuntos disponibilizados, grupos repetidos de caronas entre mesmas cidades e entre outros problemas menores. Para sanar tais problemas, foi desenvolvido um aplicativo híbrido para viabilizar a criação de grupos únicos entre cada par de cidades, sem limite de participantes, centralizando assim a organização das caronas. Cada grupo possui pelo menos um moderador, responsável por aceitar novos integrantes e controlar reclamações entre os usuários. Com essa nova arquitetura, tornou-se possível uma maior organização das caronas oferecidas e procuradas, além de ter permitido a interação mais ágil entre seus participantes, acelerando o processo de comunicação com outras plataformas de mensagens. O aplicativo apresenta como diferencial a disponibilização de escalas quando não há caronas diretas, realizando a plotagem de grafos em uma plataforma web para visualização das opções ou então exibindo uma listagem no aplicativo.

Palavras-chave: Carona solidária, Dispositivos Móveis, Sistemas de Informação.

Abstract

Brazil is one of the most expensive countries to have your own vehicle considering the purchase cost and maintenance expenses. However, with the precarious nature of public transport system, private transport continues to be most used option by Brazilians. The carpool has been one of the alternatives used for those who need to move around frequently between neighboring cities. The contact is mainly made using the instant messaging groups to broadcast the interest of supply and demand. However, this approach has its limitations and issues: the number of participants, poor organization of posts, low level of trust among members, lack of control over the topics on groups, repeated groups of same-city and other minor problems. To diminish such problems, a mobile hybrid application was developed to enable the creation of unique groups between each pair of cities, without limit of participants, but with centralizing the organization of carpools. Each group has at least one moderator, responsible for accepting new members and controlling complaints among users. With this new architecture, was possible to offer a better organization of offered carpools, allowing agile interaction among its participants and accelerating the process of communication with other messaging tools.

Keywords: Carpool, Mobile Devices, Informations Systems.

Agradecimentos

A toda minha família por acreditar que eu seria capaz e investir na realização do meu sonho.

A minha namorada por me incentivar todo o tempo e compreender minha ausência nos momentos de necessidade.

Ao professor Igor Knop, por me guiar durante todo o processo de pesquisa contribuindo de forma efetiva para um bom desenvolvimento do presente trabalho.

Aos professores do Departamento de Ciência da Computação pelos seus ensinamentos e aos funcionários do curso, que durante esses anos, contribuíram de algum modo para o nosso enriquecimento pessoal e profissional.

“Pare de começar e comece a terminar”.

Filosofia base do Kanban

Conteúdo

Lista de Figuras	6
1 Introdução	7
1.1 Objetivos	8
1.2 Justificativa	8
1.3 Motivação	9
1.4 Organização deste Trabalho	9
2 Revisão Conceitual	10
2.1 Transporte público e privado de pessoas	10
2.2 Dispositivos Móveis	12
2.2.1 Aplicações Nativas	12
2.2.2 Aplicações Híbridas	13
2.3 Grafos e Caminhos	14
2.3.1 Problema do caminho mínimo em Grafos	16
2.3.2 Algoritmo de Moore-Dijkstra	16
3 Arquitetura Conceitual	19
3.1 Arquitetura geral	19
3.2 Modelo de dados	20
3.3 Aplicativo	21
3.4 Aplicação Web e Telemetria	23
3.5 Servidor	24
3.6 Composição de serviços	25
4 Implementação	26
4.1 Aplicativo	26
4.2 Aplicação Web	28
4.2.1 Análise de uso	28
4.2.2 Visualização de grafos	30
4.2.3 Escalas	30
4.3 Serviço de Caronas	31
4.4 Composição de serviços	34
4.5 Coleta de dados	35
5 Análise dos dados	36
5.1 Dados de teste do aplicativo	36
5.2 Análise de dados reais	36
6 Conclusão	38
6.1 Limitações e Trabalhos Futuros	39
Bibliografia	40

Lista de Figuras

2.1	O Grafo A representa um grafo não direcionado e o Grafo B um grafo direcionado.	15
2.2	Exemplo de grafo com arestas ponderadas, onde as selecionadas em verde indicam o caminho mínimo para sair de s e ir até t.	17
3.1	Arquitetura conceitual	19
3.2	Quais dados são coletados e como eles se relacionam.	22
3.3	Layout do aplicativo	23
4.1	Arquitetura do aplicativo	27
4.2	Tabela de distância entre cidades	29
4.3	Gráficos gerados na análise	29
4.4	Números interessantes do aplicativo	30
4.5	Plotagem de grafos web	31
4.6	Geração de escala no aplicativo	33
4.7	33
4.8	Comunicação técnica entre os serviços	34

1 Introdução

O transporte público no Brasil frequentemente é alvo de críticas e protestos. Os problemas comumente retratados se referem às condições dos veículos, superlotação e alto custo das passagens. Isso resulta em 60% de avaliações tidas como péssimas ou ruins, segundo Pena (2016).

Diante deste cenário, quem precisa se locomover diariamente opta pelo transporte privado obtendo certo conforto e praticidade. Porém, os gastos relacionados a essa escolha são ainda maiores, considerando o valor para aquisição do bem e ainda o custo com revisões, impostos, seguros e depreciação do bem.

Tal cenário abriu caminho para um novo tipo de transporte, denominado caronas solidárias. Sua definição consiste na ideia de transportar pessoas com interesse de destino em comum, realizando a divisão dos gastos entre os passageiros.

Tal solução aparentemente resolve os problemas de desconforto gerados pelo transporte público e ameniza os gastos com a utilização do transporte privado. Entretanto, se faz necessário um estudo no que tange a legislação no contexto de caronas para definição de sua categoria, além de ser necessário a criação de ferramentas que viabilizem a organização de ofertas e procuras.

Atualmente, existem inúmeros grupos de caronas em redes sociais e aplicativos de mensagens instantâneas, porém a utilização dessas ferramentas possui limitações quanto ao limite de usuários, organização, crescente quantidade de assuntos não relacionados nos canais, segurança e registro de caronas acertadas.

Em vista dos problemas enfrentados nesse contexto e acompanhando o crescimento significativo na utilização de aplicativos de celular, foi desenvolvida uma ferramenta capaz de não só facilitar a coincidência de interesses, mas também otimizar todo o fluxo englobado em ofertas e procuras de carona.

1.1 Objetivos

O objetivo geral deste trabalho é o desenvolvimento de um aplicativo capaz de centralizar e organizar todo o processo de oferta e procura de caronas.

Para que o objetivo geral seja alcançado, é necessário desenvolver funcionalidades que gerem uma maior segurança e centralize de forma automatizada todo o processo de agendamento de carona. São, portanto, objetivos específicos para a realização deste trabalho:

- Uma revisão da legislação e atuais soluções para o transporte particular no Brasil;
- A modelagem dos principais conceitos e construção de um serviço online;
- A construção de um aplicativo protótipo que permita o consumo do serviço bem como a coleta de dados de utilização;
- A análise dos dados colhidos durante o uso real do aplicativo através de ferramentas de telemetria.
- Implementação de um algoritmo capaz de gerar escalas com janela de tempo, dando ao usuário uma opções de rota além da direta entre a cidade de origem e destino;

1.2 Justificativa

Conforme estudo realizado pelo Bolso (2017), gasta-se por ano com um carro popular cerca de dez mil reais considerando combustível, impostos, manutenção e depreciação. É sabido também que cada veículo privado com capacidade para cinco pessoas, possui ocupação média de 1,5 passageiro (RIBEIRO, 2016). Esses dados levam à conclusão de que ocupar mais os veículos privados pode gerar mais economia aos seus proprietários e comodidade a outros passageiros. Isso, sem contar na contribuição com o meio ambiente e com a diminuição do número de automóveis nas estradas.

Uma abordagem para estimular o uso de viagens compartilhadas seria a criação de uma ferramenta na forma de um aplicativo para aparelhos móveis. Segundo Agrela (2017), existem no mundo 5 bilhões de aparelhos celulares ativos para uma população

de 7,6 bilhões de habitantes, resultando em uma média 0,66 celulares por indivíduo. Acompanhando o exorbitante número de aparelhos móveis, Sá (2017) calcula que hoje existam cerca de 5 milhões de aplicativos, somando os disponibilizados nas lojas da Apple e Google.

1.3 Motivação

É comum a utilização de aplicativos de mensagens instantâneas e redes sociais para procura e oferta de caronas. Porém, por não terem tal objetivo, deixam a desejar quanto a organização, segurança e centralização das informações.

Após uma experiência pessoal de aproximadamente três anos como moderador de um grupo de carona utilizando um aplicativo de mensagens instantâneas *WhatsApp*¹, foi possível observar todas as situações problema mencionadas, contribuindo para se criar na prática as facilidades que podem auxiliar no processo.

Concomitantemente, o aprendizado em desenvolvimento *mobile*, demandava um projeto como base de estudo. Isso levou ao desenvolvimento do aplicativo aqui apresentado.

1.4 Organização deste Trabalho

O Capítulo 1 apresenta uma introdução ao assunto e explicita os objetivos almejados durante o desenvolvimento deste trabalho. No Capítulo 2 foi realizada uma revisão conceitual, contextualizando todo o problema de transporte público e privado, trazendo informações sobre as tecnologias utilizadas e os problemas encontrados para sua melhoria. O Capítulo 3 apresenta de forma didática como o aplicativo foi montado e a composição de seus serviços, para que no Capítulo 4 o processo técnico de implementação pudesse ser apresentado. O Capítulo 5 apresenta uma análise dos dados obtidos nos testes simulados e colhidos de uso real da ferramenta. Por fim, as considerações finais e sugestões para trabalhos futuros são apresentadas no Capítulo 6.

¹https://www.whatsapp.com/?lang=pt_br

2 Revisão Conceitual

Neste capítulo será feita uma revisão dos conceitos relacionados ao transporte público e privado levando em consideração suas limitações e a legislação que os regem. Adicionalmente, também será feita uma revisão das tecnologias contemporâneas para desenvolvimento de aplicações nativas e híbridas para dispositivos móveis, bem como um levantamento de algoritmos de escalonamento de horários e roteamento que serão úteis para melhorar a qualidade do serviço prestado pelo aplicativo objeto desse estudo.

2.1 Transporte público e privado de pessoas

Rota é um caminho, uma direção ou um rumo que liga um lugar a outro; um itinerário que define o caminho percorrido para chegar a algum lugar. As rotas de trânsito definem o trajeto entre o local de partida e o ponto que se deseja chegar (SIGNIFICADOS, 2015). De especial interesse deste trabalho, uma rota interurbana é composta sempre entre duas cidades. Os parâmetros de decisão utilizados para definir um percurso para cumprir uma rota podem ser distância, tempo, periculosidade, custo e entre outros. A determinação de uma melhor rota depende diretamente da definição dos parâmetros priorizados.

O transporte público pode ser definido como um serviço disponível a qualquer pessoa, podendo ser cobrado ou ofertado com gratuidade caso o passageiro se enquadre em algum programa do governo. Já o transporte privado tem seu uso limitado ao próprio dono e por quem ele conceda acesso ao veículo, não sendo, dessa forma, aberto ao público (SCHLICKMANN, 2013).

A utilização desse tipo de transporte, em grande parte, é feita por estudantes que foram morar fora e possuem interesse de retornar a cidade natal sempre que possível. Devido a isso, priorizam um baixo custo e um curto tempo de percurso, quando possível.

Com a precariedade do transporte público no Brasil e o alto custo de manutenção ao utilizar o transporte individual e a alta do custo praticado pelas empresas de transporte coletivo, surge o transporte privado.

Tal modalidade, ainda nova no país, possui um projeto de Lei aprovado no Senado e que tramita na Câmara, com intuito de regulamentar a modalidade. O projeto de Lei 28/2017 define várias obrigаторiedades ao condutor e/ou empresa responsável pelo serviço, tendo como principais (MIOZZO, 2017):

- CNH deverá ter observação de exercício de atividade remunerada;
- Condutor deverá ter seguro contra acidentes pessoais a passageiros;
- Condutor e empresa deverão ter certidão negativa de antecedentes criminais;
- Condutor deverá realizar inspeção veicular em periodicidade inferior aos demais veículos particulares;
- Empresa deverá enviar ao usuário antes da viagem o nome e fotografia do condutor, além da placa do veículo, e garantir que será o mesmo a conduzir o carro;

Anteriormente, os municípios possuíam autoridade para permitir ou não a utilização desse tipo de transporte, porém com esse novo projeto de lei a decisão é tida em âmbito nacional e cabe as cidades apenas fiscalizar.

Entretanto, tal regulamentação não se enquadra no âmbito de caronas, onde não existe uma empresa por trás das relações de negócio e tão pouco uma estrutura que consolide o processo ou mesmo prática comercial por parte dos envolvidos. O tema que abrange a legalidade em caronas não é recente. Porém, continua gerando frequentes discussões no meio jurídico. O Artigo 736 do Código Civil - Lei 10406/02 consolida, (JUSBRASIL, 2002):

Art. 736. Não se subordina às normas do contrato de transporte o feito gratuitamente, por amizade ou cortesia.

Parágrafo único. Não se considera gratuito o transporte quando, embora feito sem remuneração, o transportador auferir vantagens indireta.

Tal artigo garante então a legalidade na oferta de caronas, desde que não haja favorecimentos provenientes de tal ato. O aplicativo que foi desenvolvido se enquadra nesse contexto, tendo em vista que visa caronas solidárias, sem incentivar ou intermediar qualquer tipo de favorecimento para quem oferece o serviço.

2.2 Dispositivos Móveis

A utilização de dispositivos móveis é recente, tendo seu início ao final da década de 80 e apresentando um crescimento exponencial nos últimos anos segundo (CAPELAS, 2017). Essa tendência evidencia que o uso destes dispositivos não se trata mais de uma tendência, mas sim uma mudança cultural real.

No começo de 2018, já existia em operação no Brasil um aparelho celular para cada habitante (CAPELAS, 2017). Tais números refletem diretamente na utilização dos aparelhos como ferramentas do dia a dia, seja para viabilizar processos de negócio, aproximar pessoas ou simplesmente visando o lazer.

Para atender tais demandas, os dispositivos móveis possuem acesso a internet para permitir a navegação em sites. Mas sua maior utilização é feita através de aplicativos disponibilizados nas lojas virtuais das plataformas, facultando aos usuários personalizar seus aparelhos com a instalação de ferramentas que atenderão suas necessidades cotidianas. Acredita-se que foco nos problemas pequenos do cotidiano contribui ativamente para a expansão do uso de dispositivos móveis.

2.2.1 Aplicações Nativas

Segundo pesquisa divulgada por Ciriaco (2017), as plataformas Android, iOS e Windows Phone, juntas dominam cerca de 99,5% das plataformas mobile disponíveis no mercado atualmente. Tal dado nos permite dizer que ao desenvolver um aplicativo para essas três plataformas, é possível alcançar quase a totalidade dos dispositivos em uso no mundo.

Devido ao fato de cada uma das três plataformas possuir sua arquitetura, configurações e requisitos, seu processo de desenvolvimento necessita de ferramentas específicas capazes de realizar de forma eficiente o acesso aos recursos nativos, mesmo que para desenvolver um aplicativo com as mesmas funcionalidades.

Entretanto, ao optar por realizar o desenvolvimento de aplicações nativas, não existe reaproveitamento de código e conhecimento além do domínio do problema, pois são linguagens e ambientes completamente diferentes. Mas por se tratar de uma linguagem específica para a plataforma de destino, existe um ganho de desempenho e facilidades de acesso aos recursos nativos de forma direta (ILHASOFT, 2016).

2.2.2 Aplicações Híbridas

Sempre houve um grande interesse das empresas em desenvolvimento híbrido eficiente, que possibilitasse, através de um único código fonte (ou com pequenas modificações), gerar aplicações para plataformas diversas.

Um aplicativo híbrido é executado em cima de uma *web view*, ou seja, um navegador *web* especializado, desenvolvido exatamente para receber código em uma linguagem intermediária e realizar acesso aos recursos nativos do dispositivo de forma transparente ao desenvolvedor. (TAQTILE, 2016)

A partir da evolução do *hardware* dos dispositivos móveis e de técnicas de programação, o desenvolvimento híbrido ganhou força, ficando cada vez mais robusto e, muitas vezes, sendo imperceptível a diferença de desempenho quando comparado ao nativo. Porém é preciso realizar uma análise antes de decidir qual tipo de aplicação você irá desenvolver. As aplicações híbridas aceleram o processo de desenvolvimento e facilitam a manutenção, mas existe uma perda de desempenho que pode passar a ser evidente de acordo com a aplicação da ferramenta a ser produzida.

Para realização de uma análise completa é importante saber quais tecnologias existem no mercado, podendo dessa forma comparar suas vantagens e escolher de forma consciente a melhor opção.

- Xamarin: Framework da Microsoft que utiliza-se da tecnologia de *mobile cross-platform* para realizar chamadas nativas do sistema operacional nas plataformas móveis. Na prática, todo código do aplicativo é desenvolvido utilizando a linguagem C sharp, porém não existe um processo de emulação com o intuito de replicar a interface da plataforma de destino. Todas as chamadas são feitas de forma nativa. Comumente o Xamarin é tido como híbrido, porém é um equívoco, pois a linguagem de desenvolvimento é única, porém as chamadas a recursos da plataforma é integralmente nativa.(QUAIATO, 2017)
- PhoneGap/Cordova: Framework integralmente híbrido que permite o desenvolvimento de aplicativos usando html, css e javascript. O código desenvolvido é então compilado para uma plataforma específica e executado em cima de uma *web-*

view que emula o comportamento de um aplicativo nativo dentro de uma página web.(PHONEGAP, 2018)

- Ionic: Framework que tem como principal objetivo fornecer abstrações na utilização dos serviços do Phonegap, funcionando como uma extensão que facilita ainda mais o desenvolvimento híbrido, focado em performance e simplicidade. Uma importante característica do Ionic trata da viabilidade da substituição do javascript pelo typescript, uma linguagem mais confiável e legível, ou seja, existem características no typescript que impedem o programador de realizar desenvolvimento incorreto, deixando o código mais limpo e inteligível.(IONIC, 2017)

Dessa forma, pode-se fazer uma relação objetiva entre os frameworks. O Xamarin oferece interação nativa com o dispositivo móvel, tendo ganho em performance. Porém, mesmo garantindo o desenvolvimento em uma única linguagem, existe o retrabalho ao criar telas para todas as plataformas de destino. Em contrapartida, o PhoneGap permite desenvolvimento onde o código é integralmente reaproveitado, porém utiliza-se de javascript, uma linguagem pouco confiável. Por fim, o Ionic otimiza as chamadas de recursos nativos do aparelho e, através da utilização do Angular como ferramenta responsável por organizar a arquitetura do projeto, garante maior organização e confiabilidade da aplicação.

Com base nos dados levantados e análise realizada, o Ionic Framework foi escolhido como ferramenta de desenvolvimento do aplicativo aqui criado, devido a facilidade, agilidade e confiabilidade de desenvolvimento, sendo condizente com o objetivo aqui definido.

2.3 Grafos e Caminhos

Grafos podem ser definidos como uma estrutura matemática constituída de vértices e arestas, valorados ou não, que são comumente usados para representar ou modelar soluções e problemas do mundo real.(BIGGS, 1986)

Na grande maioria das vezes sua representação é feita de forma visual, facilitando o entendimento de problemas, onde um vértices representa elementos do contexto e as

arestas definem uma relação entre tais elementos.

Formalmente, um grafo G é representado por $G = (V, E)$, onde V é um conjunto de vértices e E um conjunto de arestas, sendo que cada aresta possui em suas extremidades um dos elementos do conjunto V , conforme definido em SZWARCFITER (1983).

A partir de um de grafo bem definido é possível executar algoritmos em sua estrutura com diversos objetivos práticos que proporcionam ganho em performance e consequentemente eficiência de processamento.

Imagine por exemplo um problema que consiste na necessidade de se saber quais são os roteadores mais importantes de uma rede interna, que se falharem desconectam toda a infraestrutura. Tal problema pode ser resolvido com o conceito de vértice de articulação, quando um vértice de articulação é removido do grafo, torna-o desconexo, ou seja, não é possível a partir de qualquer nó gerar um caminho que alcance qualquer outro nó do grafo. (SZWARCFITER, 1983)

Um grafo pode ser direcionado ou não direcionado, tal característica é definida de acordo com a característica da aresta. Considere que existam os vértices X e Y pertencentes ao conjunto V e uma aresta Z pertencente ao conjunto E e composta por X e Y . Se a aresta Z representar uma relação apenas de X para Y , dizemos que o grafo é direcionado, porém se a representação indicar uma relação dupla, ou seja, de X para Y e de Y para X , então o grafo é não direcionado. Como exemplo, na figura 2.1 o Grafo A representa um grafo não direcionado e o Grafo B um direcionado.

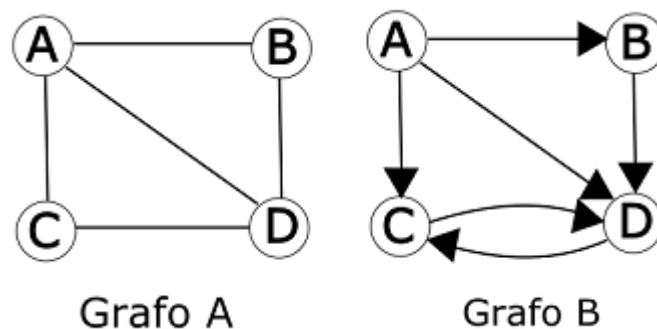


Figura 2.1: O Grafo A representa um grafo não direcionado e o Grafo B um grafo direcionado.

Um outro conceito utilizado nesse trabalho diz respeito a formação de ciclos dentro de um grafo. Um ciclo ocorre quando há um caminho em que o primeiro e o último

vértice coincidem, sem que haja repetição de nenhum outro vértice. (SZWARCFITER, 1983)

2.3.1 Problema do caminho mínimo em Grafos

Dado dois vértices de um grafo, o problema do caminho mínimo consiste em obter o conjunto de arestas cujos custos somados totalizem o menor valor possível para sair do nó de origem até o destino. O custo atribuído às arestas representam o valor a ser considerado ao escolher aquela aresta como solução, onde o objetivo é que tal valor seja minimizado.

Formalmente, considere um grafo valorado como um conjunto V de vértices, um conjunto A de arestas e uma função de peso $f : A \rightarrow B$. Segundo SZWARCFITER (1983), dado qualquer elemento v de V , encontrar um caminho P de v para cada v' de V tal que:

$$\sum_{p \in P} f(p)$$

é mínimo entre todos os caminhos conectando n a n' .

A solução mais imediata é percorrer todos os caminhos possíveis saindo da origem até o destino, armazenando os somatórios e retornar o caminho de menor valor, porém em um grafo de porte médio tal abordagem já se torna inviável.

2.3.2 Algoritmo de Moore-Dijkstra

O algoritmo de Dijkstra surgiu em 1957 como solução para problemas de caminho mínimo, porém sendo aplicável apenas quando não existem ciclos negativos no grafo a ser avaliado. Um ciclo negativo é definido quando ao realizar o somatório do peso das arestas que compõe o ciclo, o valor obtido é negativo.

O algoritmo de dijkstra realiza iterações partindo de um nó definido como origem até alcançar o nó de destino, onde a cada iteração ele opta pelo caminho que gere um menor valor acumulado, considerando sempre o trecho anterior já percorrido, dessa forma tendo poder de arrependimento podendo sempre optar pela melhor solução.

Ao término do algoritmo, o caminho dado da origem até o destino representa

não somente o caminho mínimo entre eles, mas também o caminho mínimo da origem até qualquer outro nó que faça parte desse caminho. (SZWARCFITER, 1983) A figura 2.2 mostra um exemplo de um grafo com arestas ponderadas, destacando em verde o caminho mínimo do nó s até o nó t .

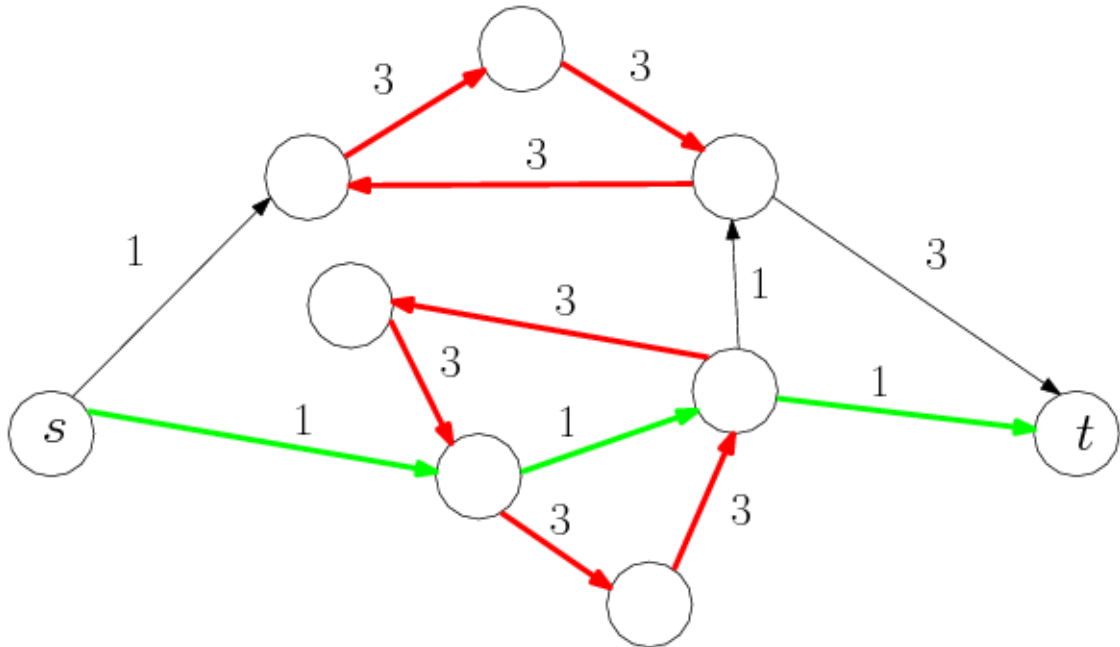


Figura 2.2: Exemplo de grafo com arestas ponderadas, onde as selecionadas em verde indicam o caminho mínimo para sair de s e ir até t .

No problema aqui abordado, o peso das arestas utilizado pelo algoritmo são definidos pela distância entre as cidades que a compõem ou pelo tempo rodoviário necessário para percorrer essa distância.

Com base nos conceitos até então apresentados, o próximo capítulo propõe uma arquitetura conceitual para a construção de uma solução para implementação de um aplicativo para compartilhamento de caronas.

A implementação de dijkstra foi construída a partir do Algoritmo 1 descrito abaixo:

Algoritmo 1: Pseudocódigo utilizado na implementação do dijkstra.

Entrada: Nó inicial e nó final.

Saída: Caminho a ser percorrido na ordem inversa

```

1 início
2   instanciar Array “distancias”, onde para o nó inicial terá valor zero e
   infinito para os demais;
3   instanciar Array “nós”, com todos os nós do grafo;
4   instanciar Array “anteriores”, vazio; while lista de nós não for vazia
   do
5     ordena nós pela menor distância;
6     defino NoMenor = nó com a menor distância;
7     removo NoMenor da lista de nós;
8     se NoMenor = nó final então
9       defino Caminho = lista vazia de nós;
10      while anteriores possuir NoMenor do
11        adiciono NoMenor ao Caminho;
12        NoMenor = anterior ao NoMenor;
13      end while
14      saia do enquanto;
15    fim se
16    se distancia do NoMenor = infinito então
17      saia do enquanto;
18    fim se
19    para cada extremidade do NoMenor faça
20      defina novaDistancia = distancias[NoMenor] +
      extremidade.distancia;
21      se novaDistancia < distancias[extremidade] então
22        distancias[extremidade] = novaDistancia;
23        anteriores[extremidade] = NoMenor;
24      fim se
25    fim para
26  end while
27  retorna Caminho
28 fim

```

3 Arquitetura Conceitual

Neste capítulo será feita a descrição da arquitetura planejada para a construção do aplicativo e serviços que o constituem. Assim será possível entender como as suas partes constituintes se relacionam e entregam as funcionalidades para os usuários.

3.1 Arquitetura geral

Arquitetura geral sugerida se baseia em um serviço web central, o qual é acessado por aplicativos e aplicações web enquanto realiza a persistência de dados em um banco de dados externo. A Figura 3.1 apresenta a arquitetura sugerida para a implantação do sistema de caronas.

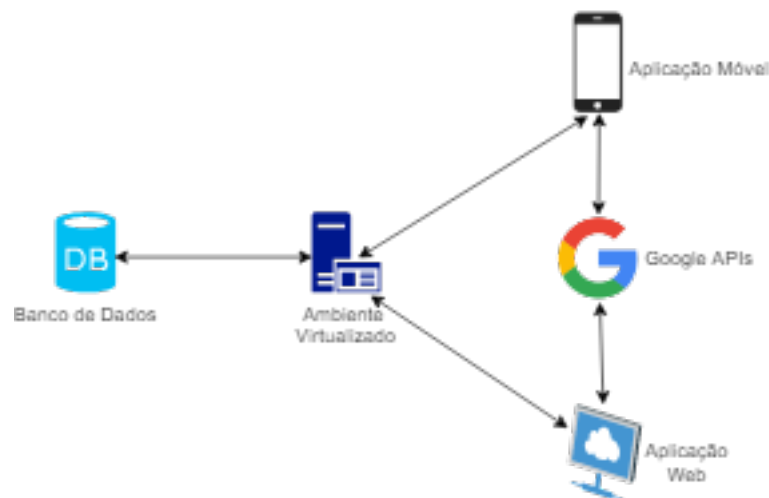


Figura 3.1: Arquitetura sugerida para o desenvolvimento do sistema de caronas: serviços de computação na nuvem sob demanda para hospedagem de serviço, aplicação web e banco de dados e os clientes com aplicativos em individuais.

A solução proposta prevê a implementação na forma de um serviço web, responsável por toda lógica de negócio e realiza a persistência em um sistema gerenciador de banco de dados. Recomenda-se que esse serviço seja hospedado pela contratação de um sistema de computação na nuvem, baseado na virtualização de servidores que permitem o fácil escalonamento sob demanda, além de agregar maior estabilidade à aplicação.

O aplicativo é o principal meio de acesso ao sistema por parte dos usuários, sendo executado nos seus respectivos dispositivos móveis. Para que os usuários possam instalá-lo, ele se encontra disponível nas principais lojas das respectivas plataformas.

Adicionalmente, uma interface web será disponibilizada para tarefas de gestão e acompanhamento dos dados, além de prover algumas das funcionalidades do sistema para o público em geral. Também recomenda-se o uso de contratação de um serviço de computação na nuvem para hospedagem.

3.2 Modelo de dados

Esta seção descreve os principais conceitos e dados capturados para a modelagem do problema.

Um **Participante** modela os dados necessários para registro dos usuários no sistema. No momento do cadastro o usuário cria uma senha e informa o número de telefone, que atua como identificador único, além de informações complementares como nome, e-mail, cidade e opcionalmente o perfil do *facebook*. O usuário informa também se seu interesse é procurar ou oferecer caronas, o que será importante para dimensionar as ofertas e procuras dentro do grupo. Quando o participante realiza uma primeira oferta de carona é capturado qual veículo ele colocou como próprio, dessa forma nas ocasiões futuras tal campo já vem preenchido facilitando na usabilidade da ferramenta. Existe também uma funcionalidade de curtidas, onde outros usuários podem recomendar participantes, ou então através da funcionalidade de reclamação, relatar problemas em caronas anteriores, deixando visível para todos os usuários, gerando assim maior segurança entre os membros e confiabilidade nas caronas oferecidas. Ainda sobre segurança, é previsto que nas próximas versões seja implementado um processo mais rígido de acesso, onde um moderador da aplicação tenha que permitir o cadastro de novos usuários, fazendo-o apenas após receber certidão negativa de antecedentes criminais e foto da CNH do motorista.

Um **Grupo** define uma ligação entre duas cidades, armazenando as cidades de suas extremidades, seu nome público e seu status.

Quando um Participante cria um grupo, a relação participante-grupo é feita em “ParticipanteGrupoRelacional”, onde é setado que esse participante é moderador e se

encontra liberado dentro desse grupo. Quando existe uma solicitação de ingresso é armazenado se existe indicação de algum outro membro e inicialmente o status é não liberado, até que um moderador aprove a solicitação, liberando o acesso às funcionalidades para o usuário. Se um usuário for excluído de um grupo, o indicador de excluído é alterado para verdadeiro e esse participante não consegue solicitar novo ingresso. Porém se o usuário resolve sair por vontade própria, ele desativa esse vínculo com o grupo, mas pode futuramente solicitar novo ingresso.

A Figura 3.2 apresenta quais dados são coletados e como eles se relacionam, cada um dentro de seu contexto.

Os dados são coletados durante a utilização do aplicativo e nenhum registro é excluído. As caronas oferecidas e procuradas permanecem armazenadas mesmo quando sua vigência acaba, sendo possível realizar análises futuras em cima de tais dados. O mesmo acontece com a criação de grupos e participantes.

3.3 Aplicativo

O aplicativo tem como objetivo ser a interface com o usuário, o qual é capaz de capturar as ações e exibir os resultados de forma intuitiva e mais direta possível.

Durante todo o desenvolvimento sempre foi planejado deixar o aplicativo bem atrativo, com uso de cores fortes e sensação de boa navegabilidade, gerando uma boa experiência do usuário. Foi tomado um cuidado especial para manter as características da plataforma na qual ele estaria sendo executado. Então, quando é identificado um Android, os botões e menus se refletem de acordo com a plataforma. O mesmo ocorre quando o iOS é detectado.

Ainda pensando em uma boa experiência de uso, foi criado um controle de sessão, onde o usuário precisa realizar login apenas no primeiro acesso, ficando registrado sua sessão até que o usuário a encerre.

A ferramenta possui três funcionalidades principais: “Mais grupos”, “Meus grupos” e “Opções” de grupos que serão discutidas a seguir.

A tela “Mais grupos” exibe todos os grupos criados no aplicativo que o usuário ainda não participe e estejam disponíveis para solicitar participação, trazendo as in-

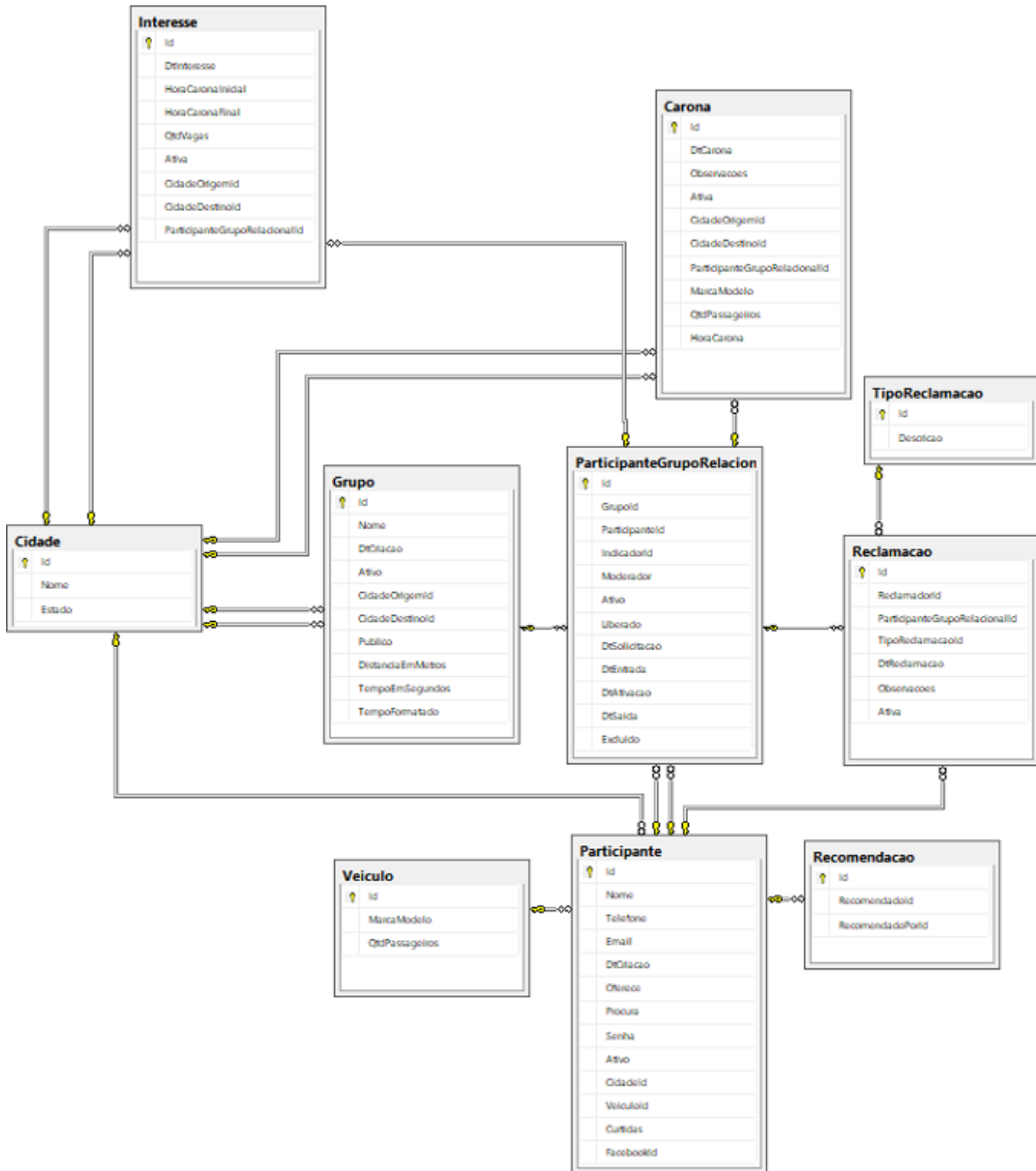


Figura 3.2: Quais dados são coletados e como eles se relacionam.

formações básicas necessárias para identificação, como nome do moderador, as duas cidades que o compõem e a data de criação.

Ao solicitar participação em um grupo, tal grupo passa a ser exibido na tela “Meus grupos” divididos em subcategorias que indicam os quais o usuário modera, participa, foi

excluído, resolveu sair, foi desativado ou simplesmente esta aguardando aprovação do moderador.

Para interagir com as funcionalidades de cada grupo, basta selecioná-lo na tela de “Meus grupos” que as opções do grupo serão listadas, existindo então um controle de acesso que verifica se o usuário desempenha papel de moderador ou usuário simples do grupo escolhido, determinando dessa forma quais opções deverão ser exibidas para escolha. A partir daí o usuário consegue procurar ou oferecer caronas, ver membros, indicar pessoas, realizar reclamações, sair do grupo e entre outras.

A Figura 3.3 ilustra a tela de “Meus Grupos” e exibe um exemplo da listagem de caronas oferecidas.

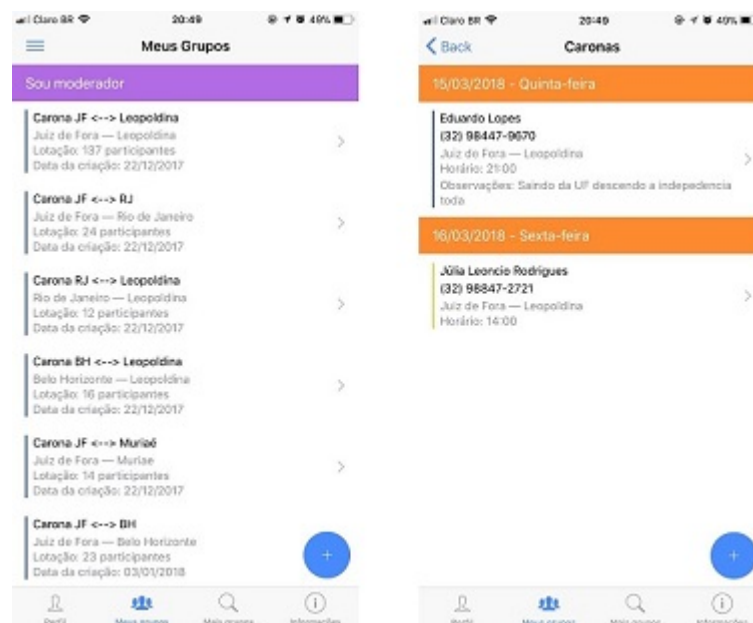


Figura 3.3: A tela “Meus Grupos” lista todos os grupos que eu participo, sendo a entrada para acesso das funcionalidades disponíveis. A tela “Caronas” apresenta todas as caronas futuras disponíveis, ao selecionar alguma o usuário é direcionado para o contato do participante que a ofereceu.

3.4 Aplicação Web e Telemetria

Foi criada uma ferramenta web com o intuito de compilar os dados que constantemente são coletados através do aplicativo e exibi-los em um *dashboard* que centralize as informações em uma só visualização através de gráficos, facilitando o acompanhamento do crescimento de cadastros, a distribuição entre as cidades, quais plataformas móveis possuem

mais usuários, além de dados de uso, como por exemplo número de caronas oferecidas e procuradas.

Para uma visualização ainda mais completa do alcance que os grupos de carona permitem, existe a possibilidade de plotagem de grafos parametrizados que considera as características dos grupos, gerando uma eficiente ferramenta de avaliação.

A funcionalidade principal do aplicativo é focada em oferecer caronas apenas entre duas cidades organizadas pelos grupos. Entretanto, a partir da plotagem dos grafos tornou-se viável a possibilidade do usuário chegar a seu destino através de escalas. Ou seja, ele realiza uma ou mais caronas intermediárias que o destino final é o desejado.

Na prática, quando o usuário pede carona ele define qual a cidade de origem e qual o destino, além da janela de horário que ele esta disponível para iniciar a viagem. Com essas informações é feita uma requisição para que a API possa consultar o banco e retornar as caronas disponíveis nesse intervalo de tempo. Quando não existe nenhuma carona com esses parâmetros, outro método na API é requisitado, onde os cálculos de escalas disponíveis serão feitos.

A ideia do algoritmo é inicialmente montar um grafo com as rotas viáveis dentro da janela de tempo informada pelo usuário, limitando o tempo total de viagem ao dobro do tempo caso a viagem fosse feita de forma direta. Tal implementação se baseou no algoritmo da Listagem 2.

A partir daí, executamos o algoritmo de dijkstra para obter duas rotas possíveis, uma que indica o caminho a ser percorrido pelo usuário para chegar mais cedo no destino e outra que fornece o caminho utilizando o menor número de escalas. A implementação de dijkstra foi baseada na Listagem 1.

3.5 Servidor

O servidor é responsável por receber as requisições vindas do aplicativo e aplicação web através de protocolos http e acessar o banco de dados quando preciso para tratar as informações.

O papel do servidor é fundamental para o correto funcionamento do ecossistema montado, pois todas as interações entre os componentes passam por ele, principalmente

as validações de acesso. Devido a tal importância, ele fica hospedado em um ambiente estável na nuvem que possa garantir acesso e escalabilidade sempre que necessário.

3.6 Composição de serviços

Os serviços interagem de forma transparente para o usuário, gerando impressão de que tudo ocorre no dispositivo móvel e plataforma web, entretanto existe comunicação frequente entre o aplicativo, aplicação web, servidor e banco de dados.

No contexto do aplicativo por exemplo, quando um usuário se cadastra é realizada uma requisição ao servidor para que faça a persistência do novo participante no banco de dados, que realiza a inserção e retorna o sucesso para o servidor, que por sua vez retorna a mensagem para o aplicativo finalmente a exibir para o usuário.

Quando observado na aplicação web o processo de construção dos gráficos de análise dos dados, uma requisição é agora iniciada do componente web e novamente chega ao servidor, que irá realizar consultas no banco para tratar os dados e retornar para a aplicação.

Na prática os componentes se comunicam sempre com o servidor de forma centralizada, tratando os dados da forma solicitada e os retornando para o cliente.

4 Implementação

Este capítulo visa discriminar quais ferramentas técnicas foram utilizadas para o desenvolvimento de cada componente da aplicação como um todo, explicando o processo de implementação, justificando as decisões tomadas e a avaliação de cada etapa.

4.1 Aplicativo

Para desenvolvimento do aplicativo foi utilizado o *Ionic Framework* em sua versão 3, sendo a versão mais atual. O *framework* viabiliza um desenvolvimento híbrido tornando possível a implementação em uma única linguagem, onde a portabilidade é feita para as plataformas móveis de destino no momento da publicação do aplicativo.

Foi utilizado também o *Angular* na versão 4, assumindo o papel de codificar as telas no aplicativo, além de fornecer a implementação do protocolo http para comunicação com o servidor.

Durante o desenvolvimento a documentação do *Ionic Framework Documentation* (2017), oferecida de forma detalhada e com apresentação de exemplos, facilitou o entendimento e principalmente a utilização dos *plugins*, responsáveis por fazer a interação com serviços nativos do dispositivo móvel.

A arquitetura em que o aplicativo se baseia é subdivida em três principais camadas:

1. *Pages*: Cada tela do aplicativo é encapsulada como uma página, possuindo seu próprio arquivo de elementos html, personalização de layout(css) e codificação(typescript). Juntos eles são auto suficientes e capazes de realizar todas as tratativas dentro de seu contexto.
2. *Models*: Trata-se de uma representação do que irá transitar entre a aplicação e o servidor. Basicamente quando trata-se de um método *POST*, uma *model* é inicialmente instanciada em uma página, preenchida pelo usuário e enviada para o servidor. O

caso reverso é quando em um método *GET* um objeto é retornado e mapeado para exibição no aplicativo.

3. *Services*: Responsável por fazer a interação direta entre aplicação e servidor. Utiliza-se de chamadas *http* para se comunicar com a *Web API* via *end points* conhecidos e acessar os métodos disponibilizados. O retorno dos métodos são tratados com um *subscribe* que permite controlar a requisição e determinar o exato momento do retorno à aplicação.

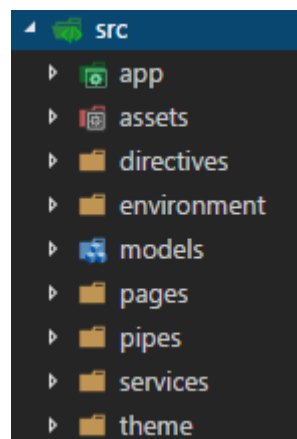


Figura 4.1: Arquitetura base do aplicativo com as camadas principais de pages, models e services

Através de *plugins* intermediados pelo *ionic-native* (IONIC, 2018), um facilitador na utilização de funcionalidades do *Cordova*, torna-se possível realizar o acesso as funcionalidades do dispositivo móvel.

O *cordova-plugin-camera*(v2.4.1)² é responsável por realizar o acesso as funcionalidades da câmera, permitindo não somente tirar fotos e vídeos, mas também acessar o álbum para seleção de mídias da galeria. Outro plugin utilizado foi o *cordova-plugin-contacts*(v2.3.1)³ que permite acesso a agenda telefônica, sendo possível realizar busca por contatos e até mesmo salvar novos números. A integração com o facebook para utilização da foto de perfil é feita a partir do *cordova-plugin-facebook4*(v.1.9.1)⁴, que disponibiliza uma comunicação simplificada com a API pública da rede social facilitando o login e consulta de dados. Para o envio de notificações o *onesignal-cordova-plugin*(v2.2.5)⁵ é

²<https://ionicframework.com/docs/native/camera/>

³<https://ionicframework.com/docs/native/contacts/>

⁴<https://ionicframework.com/docs/native/facebook/>

⁵<https://ionicframework.com/docs/native/onesignal/>

utilizado para registrar os identificadores de cada aparelho e enviar efetivamente as mensagens de acordo com as demandas. Por fim, o `cordova-plugin-x-socialsharing(v5.3.2)`⁶ permite o envio de SMS e ligações direto pelo aplicativo e principalmente é capaz de abrir o whatsapp exatamente na conversa de algum membro do grupo.

A união de todos esses processos viabilizou a construção final do aplicativo desenvolvido.

4.2 Aplicação Web

Toda aplicação foi desenvolvida utilizando Angular 5 como linguagem *frontend* e se comunicando com a mesma API responsável pelos tratamentos do aplicativo.

Para o desenho dos grafos foi utilizado o componente do *cytoscape(v3.2.14)* (CYTOSCAPE, 2017) e para exibição dos gráficos o componente utilizado foi o *Google Charts(v3.1.12)* (GIRETTI, 2017).

A aplicação web possui três tipos de análises realizadas sobre os dados constantes no banco de dados de uso do aplicativo: Análise de uso; Análise em Grafos e Escalas. cada uma delas será vista nas próximas seções.

4.2.1 Análise de uso

Esta página centraliza os dados coletados pelo aplicativo, viabilizando uma análise direcionada em indicativos que permitam um acompanhamento de uso dos usuários.

Esta ferramenta visual mostra inicialmente uma tabela para definir a distância e tempo entre as cidades de um grupo conforme mostrando na 4.2. Tais valores serão adicionados no momento da criação dos grupos no aplicativo, porém como existem versões desatualizadas da ferramenta ainda em uso, tal tabela funciona de forma complementar para definir tais valores. A inserção pode ser feita de forma manual ou utilizando a API *Distance Matrix* do Google, que retorna via chamada `https` os dados solicitados.

Na segunda parte da tela, quatro gráficos são plotados como pode ser observado na 4.3, indicando a quantidade de grupos que os usuários participam, as cidades onde há

⁶<https://ionicframework.com/docs/native/socialsharing/>

Parâmetros de distância e tempo dos grupos				
Grupo	Distância	Tempo formatado	Tempo em segundos	Ações
Carona Santos <-> Sao Paulo	74,1 km	1 hora 4 minutos	3837	▼
Carona Florianopolis <-> Itajai	96,4 km	1 hora 24 minutos	5018	▼
Carona Juiz de Fora <-> Barbacena	96,1 km	1 hora 28 minutos	5293	▼
Carona Juiz de Fora <-> Carandai	127 km	1 hora 45 minutos	6318	▼
Carona Kta <-> Viçosa	116 km	2 horas 2 minutos	7319	▼
Carona Belo Horizonte <-> Mariana	118 km	2 horas 3 minutos	7405	▼
Carona JF <-> Leopoldina	116 km	2 horas 4 minutos	7439	▼
Carona Blumenau <-> Sao Jose	145 km	2 horas 4 minutos	7448	▼
Carona Juiz de Fora <-> Uba	131 km	2 horas 14 minutos	8069	▼
Carona Lafaiete <-> JF	162 km	2 horas 17 minutos	8190	▼

Figura 4.2: Informação de distância e tempo entre duas cidades, podendo ser obtida via API Distance Matrix do Google ou inserida manualmente

maior aceitação, qual é a plataforma de dispositivo móvel mais utilizada e por fim uma análise de crescimento de cadastros por mês.

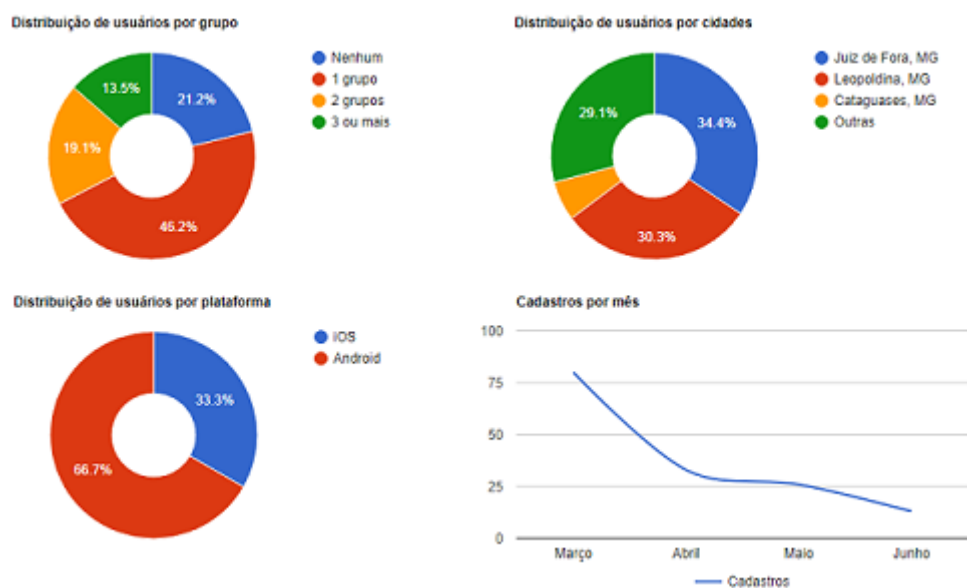


Figura 4.3: Gráficos de usuário por grupo e cidades, análise das plataformas utilizadas pelos usuários e da quantidade de cadastros realizados nos últimos meses

Ao final da página existe uma outra tabela exibindo dados de controle conforme mostrado na Figura 4.4, apenas para identificar a quantidade total de usuários cadastrados, pedidos de ingresso pendentes em grupos, uma análise de caronas oferecidas e procuradas que estão ativas e por fim a quantidade de grupos que foram excluídos.

Números interessantes				
Usuários cadastrados	Pedidos pendentes	Caronas oferecidas ativas/total	Caronas procuradas ativas/total	Grupos excluídos
340	8	1/386	0/263	5

Figura 4.4: Informações de usuários cadastrados, pedidos de ingresso pendentes, caronas oferecidas futuras pelas totais, caronas procuradas futuras pelas totais e total de grupos excluídos.

4.2.2 Visualização de grafos

A ferramenta web disponibiliza uma página para plotagem de grafos, com o objetivo de realizar uma análise visual do alcance dos grupos e seus encadeamentos, dando uma visão na qual é possível observar o alcance do aplicativo em relação aos Grupos criados.

Cada cidade que pertença a pelo menos um grupo do aplicativo vira um nó. A partir dela, uma aresta não direcionada é criada indicando a existência de um grupo que a liga até uma outra cidade. Assim, cada aresta é um par de cidades conectadas por um grupo.

É possível realizar filtros que consideram apenas grupos com caronas em período específicos. Outra possibilidade é parametrizar as informações das arestas, podendo indicar a distância, tempo ou ambas, além de colorir de acordo com as ofertas do grupo.

A Figura 4.5 representa a plotagem de um grafo com todos os grupos existentes no aplicativo, com as arestas indicando a distância entre as cidades.

4.2.3 Escalas

Quando não há possibilidade de uma viagem direta da origem até o destino, o participante tem a opção de visualizar sugestões de escalas, ou seja, passar por cidades que o possibilite da mesma forma alcançar seu destino final. Os resultados das escalas geradas são exibidos de formas diferentes no aplicativo e na versão web. No aplicativo, é exibido apenas as duas rotas geradas de forma sequencial, devido a limitação da tela do dispositivo móvel conforme exemplificado na Figura 4.6.

Quando a consulta é feita na versão web, existe uma flexibilidade maior para exibir os resultados, dessa forma utilizamos grafos direcionados para delimitar as respostas,

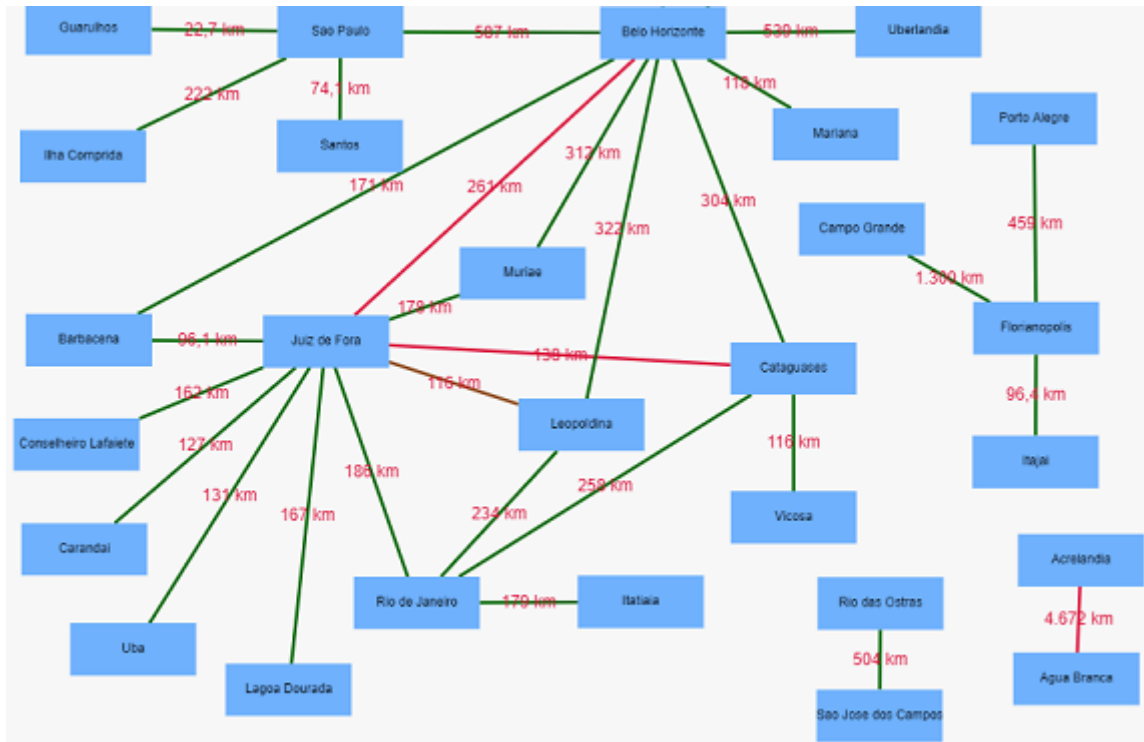


Figura 4.5: Plotagem de grafos indicando grupos entre cidades, sendo possível parametrizar as arestas e filtrar apenas por grupos com caronas vigentes em períodos específicos.

onde suas arestas são coloridas de forma a indicar os caminhos a serem percorridos para alcançar uma viagem mais rápida ou com menor número de escalas, um exemplo pode ser observado na figura 4.7.

A preparação dos dados para a plotagem do grafo na versão web e exibição das escalas no dispositivo móvel é integralmente feita na API REST, centralizando o processamento no componente capaz de efetuar de forma mais eficiente. A exibição das escalas permite ao usuário um maior poder de decisão, não ficando limitado apenas as caronas oferecidas nos grupos que ele pertence.

O algoritmo 2 realiza a instanciação dos elementos principais e chama o Algoritmo recursivo 3 que efetivamente realiza as consultas de rotas possíveis a partir de cada origem.

4.3 Serviço de Caronas

Para definição da escolha do servidor foi preciso levar em conta o custo do serviço e a escalabilidade prevendo um potencial sucesso no uso do aplicativo.

A partir dessa análise a melhor opção foi alocar uma máquina virtual na AWS

Algoritmo 2: Pseudocódigo utilizado para construção do grafo indicando as rotas possíveis para o usuário.

Entrada: data de interesse da carona com hora inicial, cidade de origem, cidade de destino e duração da viagem feita sem escalas

Saída: Grafo direcionado com as opções de rota

```

1 início
2   instanciar “grafo” vazio;
3   instanciar “tempoMaximo” = 2 * (tempo de viagem sem escalas);
4   chamar Algoritmo 3 com comportamento recursivo;
5   remover arestas sumidouros onde o nó de destino não seja a cidade de
   destino informada pelo usuário.;
6 fim

```

Algoritmo 3: Algoritmo recursivo usado no algoritmo 2 para obter os destinos possíveis a partir de uma cidade de origem.

Entrada: data de interesse da carona com nova hora inicial, cidade de origem, cidade de destino, tempo gasto atual e tempo máximo permitido

Saída: sem retorno

```

1 início
2   instanciar “caronas” = todas as caronas saindo da cidade de destino
   no dia de interesse, onde o destino seja uma cidade diferente da origem;
3   para cada carona faça
4     instanciar “tempoNecessario” = tempo necessário para ir da
   origem até o destino da carona selecionada;
5     instanciar “tempoEspera” = tempo desde a chegada na cidade de
   origem até o horário da carona;
6     se  $tempoNecessario + tempoEspera + tempoGastoAtual \leq$ 
    $tempoMaximoPermitido$  então
7       adicionar carona selecionada ao grafo;
8       chamar Algoritmo 3 passando a data de interesse como a data
   de chegada no destino, a cidade de origem como a cidade de
   destino, a cidade de destino continua a mesma, o tempo gasto
   atual vira o somatório feito no “Se” e o tempo máximo permitido
   continua o mesmo;
9     fim se
10  fim para
11 fim

```

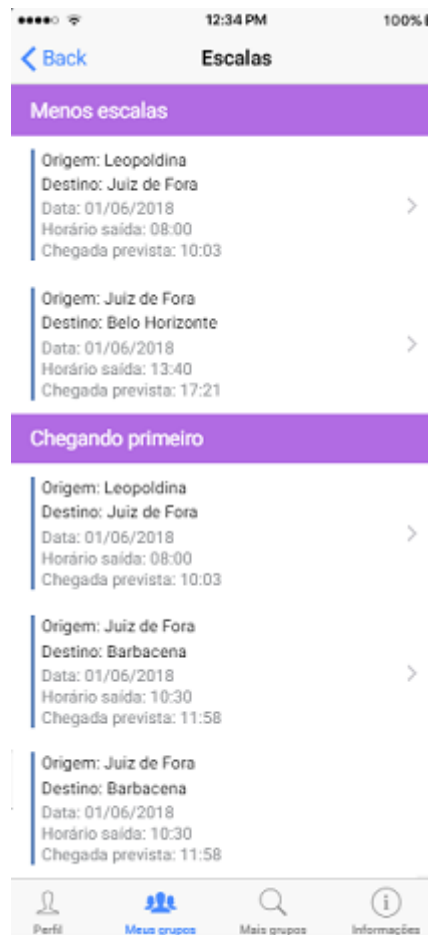


Figura 4.6: Sugestão de escalas no aplicativo quando não há carona direta. É listada a opção mais rápida e outra com menor número de escalas.



Figura 4.7: [Geração de escala web via grafo] Sugestão de escalas na versão web quando não há carona direta. Um grafo direcionado é plotado onde a cor das arestas representa o caminho a seguir, nas arestas existe a informação de horário de partida/chegada prevista.

da Amazon (AMAZON, 2018), que oferece um ano de serviço gratuito e total capacidade de correspondência de acordo com a utilização dos serviços dinamicamente.

Foi construída uma Web Api Rest em .Net Framework 4.6, com sua infraestrutura organizada em camadas. Sendo elas:

1. Presentation: Camada responsável pela interação com os serviços, é nela onde os end points são encontrados;
2. Application: É aonde as ViewModels são definidas e o processo de mapeamento de entidade para ViewModel é configurado;
3. Domain: Trata-se do núcleo desse tipo de infraestrutura, sendo aonde as entidades do banco são definidas;
4. Infra: Responsável por fazer os acessos ao banco;

4.4 Composição de serviços

Por fim, com o banco de dados modelado, aplicativo desenvolvido e servidor configurado, foi realizado o processo de comunicação entre os serviços baseado no protocolo cliente-servidor, conforme ilustrado na Figura 4.8.



Figura 4.8: Comunicação técnica entre os serviços

Vale ressaltar que o banco de dados se encontra no mesmo ambiente do servidor, agilizando o processo de troca de informação entre os componentes focando em manter a aplicação fluida para o usuário.

Por se tratar de um aplicativo que pode utilizar redes instáveis, existiu uma constante preocupação com o tempo de resposta entre os componentes, devido a isso foi implementado uma abordagem que utiliza-se de cache no aparelho móvel, com o objetivo de diminuir as trocas de requisições e tratamentos.

A aplicação, podendo ser na plataforma Android ou iOS, se mantém em constante interação com o servidor hospedado na *Amazon*, que por sua vez se comunica com o banco relacional *SQL Server* buscando ou inserindo registros.

4.5 Coleta de dados

O processo de divulgação do aplicativo foi realizado através de redes sociais, focando-se em grupos específicos de carona do Facebook e WhatsApp, o que permitiu a coleta de dados reais da ferramenta. Os registros obtidos foram armazenados no banco de dados relacional SQL SERVER 2014 da Microsoft, instalado em uma máquina virtual na Amazon.

O serviço que engloba a máquina virtual na Amazon é denominado *Elastic Compute Cloud (EC2)*⁷, que funciona como um *web service* fornecedor de capacidade computacional segura e redimensionável na nuvem. Foi alocado para o CaronAdmin uma instância t2.micro, com 40GB de HD e 2GB de memória RAM.

O aplicativo se encontra disponível para Android e iOS e pode ser baixado de forma gratuita pesquisando por “CaronAdmin” nas respectivas lojas virtuais.

A plataforma web pode ser acessada $\langle \text{http://ec2-54-207-109-126.sa-east-1.compute.amazonaws.com/#/} \rangle$, não é recomendado acesso via dispositivo móvel, tendo em vista que não houveram tratamentos responsivos neste contexto.

O aplicativo encontra-se em produção, porém para exemplificação das escalas é possível fazer uma busca por caronas no grupo de “Leopoldina - Belo Horizonte” na data de 01 de Junho de 2018 entre 08:00 e 12:00. Tal dia é recomendado pois as caronas que foram oferecidas permitem gerar uma escala de Leopoldina para Belo Horizonte fazendo escala em Juiz de Fora e Barbacena.

O próximo capítulo irá definir a metodologia dos testes e fazer uma análise dos dados coletados pelo aplicativo, focando em seu crescimento e utilização.

⁷<https://aws.amazon.com/pt/ec2/>

5 Análise dos dados

Neste capítulo definimos como as instâncias de teste foram montadas e realizamos uma análise dos dados reais de uso da ferramenta em produção.

5.1 Dados de teste do aplicativo

O processo de desenvolvimento do aplicativo foi gradativo. Desta forma cada etapa que era concluída passava por seu respectivo processo de testes. Inicialmente era realizado a checagem das funcionalidades via a própria aplicação, garantindo que cumpria com seus requisitos, posteriormente através de *scripts* de banco de dados era realizados testes de carga e estresse.

5.2 Análise de dados reais

O aplicativo foi disponibilizado inicialmente para os dispositivos Android em Dezembro de 2017 e em Fevereiro de 2018 para usuários do iOS. Após seis meses do lançamento é possível observar 352 usuários cadastrados, alcançando seis estados brasileiros, onde dois terços acessam através da plataforma Android e o restante iOS.

Houve uma maior concentração em Juiz de Fora, Leopoldina e Cataguases, sendo justificado pela divulgação realizada nas redes sociais, onde o meio social era coincidência com tais regiões. Porém, vale ressaltar o cadastro de usuários de cidades distantes, como por exemplo Salvador na Bahia e Bom Jesus do Sul no Paraná.

Observando o sistema operacional do dispositivo móvel utilizado pelo usuário, vale ressaltar que a liberação do uso para Android foi feita dois meses antes quando comparado ao iOS, nos permitindo dizer que há uma aceitação satisfatória também por parte do sistema operacional da Apple.

Como esperado, foi observado uma maior utilização de dispositivos da Google, simplesmente pelo fato de representarem atualmente cerca de 85% do mercado de apare-

lhos celulares.

Outra observação interessante diz respeito a 73 usuários que se cadastraram no sistema, porém não solicitaram participação em nenhum grupo. Tal fato indica possivelmente uma falha na interface que de alguma forma desmotivou o usuário a seguir com o uso ou a não entender o conceito do aplicativo. Porém, ainda sim não houveram avaliações negativas nas lojas virtuais das plataformas móveis. Vale mencionar inclusive que usuários de outros estados usam a ferramenta e chegaram ao aplicativo através de buscas realizadas nas lojas das plataformas.

6 Conclusão

Tendo em vista o precário cenário do transporte público no Brasil e o alto custo com transporte privado individual, a ideia de caronas vem como opção viável e concreta na busca por soluções nesse contexto.

O maior problema encontrado para quem opta por tal meio é exatamente encontrar pessoas que ofereçam ou estejam procurando caronas para lugares que possam coincidir com seus interesses, tudo isso de forma rápida e segura.

Com o aplicativo desenvolvido, tal problema tende a ser minimizado e posteriormente sanado, pois é possível realizar divulgação de ofertas e procuras de forma rápida e gratuita, possibilitando interação frequente entre os envolvidos e formas de obter segurança nos contatos realizados.

A partir de apenas seis meses de uso, o aplicativo de forma geral se mostrou bem aceito pelos usuários, registrando um uso considerável e constante retorno de sugestões, gerando uma melhoria contínua nas funcionalidades.

A possibilidade de oferecer escalas de caronas aos usuários deixa o aplicativo com um importante diferencial, gerando ainda mais facilidades e deixando os participantes com maior capacidade de decisão, além de permitir uma visão mais abrangente dos locais que consegue chegar apenas com caronas.

Os objetivos mencionados na subseção 1.1 foram integralmente atendidos, onde uma análise da legislação no âmbito de transporte particular no Brasil foi realizada e implementamos todo o processo de criação de serviço online, modelando seus conceitos e implementando na prática as características abordadas. Os dados colhidos são amplamente analisados através dos gráficos gerados e disponibilizados de forma clara para os usuários.

6.1 Limitações e Trabalhos Futuros

Com a conclusão do projeto foi possível obter um aplicativo sólido e consistente, que executa de forma eficiente o que se propôs a resolver. Entretanto uma maior aceitação dos usuários ainda pode ser alcançada, principalmente através de divulgação, pois quanto maior o uso, maior o número de publicações de oferta e procura, gerando assim um crescimento contínuo.

Vale ressaltar que mesmo existindo dificuldades na utilização de grupos de mensagens instantânea, eles ainda são os preferidos, principalmente por não haver a necessidade da instalação de um novo aplicativo. Ao conseguir romper tal barreira, a ferramenta aqui construída pode crescer ainda mais.

Existem novas tecnologias também no que diz respeito ao desenvolvimento híbrido, como por exemplo o React Native (REACT, 2018), que apresenta uma nova proposta dentro desse tipo de desenvolvimento sendo uma oportunidade interessante em ganho de performance.

O desenvolvimento para aplicações móveis é recente e oferece um mundo de soluções ainda não exploradas, onde o mais importante é focar na experiência do usuário, sendo capaz de gerar soluções rápidas, eficientes e que prenda cada vez mais a atenção de quem a utiliza.

Bibliografia

- AGRELA, L. *5 bilhões de pessoas têm smartphones*. Exame, 2017. Visitado em Fev. 2018. Disponível em: <https://exame.abril.com.br/tecnologia/5-bilhoes-de-pessoas-tem-smartphones/>.
- AMAZON. *AWS AMAZON*. 2018. Visitado em Jul. 2018. Disponível em: <https://aws.amazon.com/pt/free>.
- BIGGS, N. *Graph Theory*. [S.l.]: Oxford University Press, 1986.
- BOLSO, G. de. *Qual o gasto anual com um carro? O seu dinheiro vale mais*, 2017. Visitado em Mar. 2018. Disponível em: <http://www.oseudinheirovalemais.com.br/qual-o-gasto-anual-com-um-carro/>.
- CAPELAS, B. *Até o fim de 2017, Brasil terá um smartphone por habitante, diz FGV*. 2017. Visitado em May. 2018. Disponível em: <http://link.estadao.com.br/noticias/gadget,ate-o-fim-de-2017-brasil-tera-um-smartphone-por-habitante-diz-pesquisa-da-fgv,70001744407>.
- CIRIACO, D. *Android cresce iOS diminui e Windows Phone quase some aponta relatório*. TecMundo, 2017. Visitado em Mar. 2018. Disponível em: <https://www.tecmundo.com.br/dispositivos-moveis/119411-android-ios-windows-phone-mercado.htm>.
- CYTOSCAPE. *Cytoscape.js*. 2017. Visitado em Jun. 2018. Disponível em: <http://js.cytoscape.org/#getting-started/including-cytoscape.js>.
- DOCUMENTATION, I. *Ionic Documentation*. 2017. Visitado em Mar. 2018. Disponível em: <https://ionicframework.com/docs/>.
- GIRETTI, A. *Using Google Charts in Angular 4 project*. 2017. Visitado em Jun. 2018. Disponível em: <http://anthonygiretti.com/2017/10/12/using-google-charts-in-angular-4-project-part-1/>.
- ILHASOFT. *O que são apps nativos e como eles funcionam?* 2016. Visitado em Mar. 2018. Disponível em: <http://ilhasoft.com.br/o-que-sao-apps-nativos-e-como-eles-funcionam/>.
- IONIC. *PhoneGap - Concepts*. 2017. Visitado em Jun. 2018. Disponível em: <https://ionicframework.com/docs/intro/concepts/>.
- IONIC, N. *Ionic, Native*. 2018. Visitado em May. 2018. Disponível em: <https://ionicframework.com/docs/native/>.
- JUSBRASIL. *Art. 736 do Código Civil - Lei 10406/02*. 2002. Visitado em May. 2018. Disponível em: <https://www.jusbrasil.com.br/topicos/10688972/artigo-736-da-lei-n-10406-de-10-de-janeiro-de-2002>.
- MIOZZO, J. *Senado aprova Projeto de Lei que regulamenta Uber e Cabify no Brasil com emendas*. Infomoney, 2017. Visitado em Mar. 2018. Disponível em: <http://www.infomoney.com.br/negocios/grandes-empresas/noticia/7046586/senado-aprova-projeto-lei-que-regulamenta-uber-cabify-brasil-com>.

PENA, R. A. *Problemas No Transporte Público. Transporte Público No Brasil*. Brasil Escola, 2016. Visitado em Mar. 2018. Disponível em: brasilecola.uol.com.br/geografia/problemas-no-transporte-publico.htm.

PHONEGAP. *PhoneGap - Reviews, News and Analysis*. 2018. Visitado em Jun. 2018. Disponível em: <http://www.businessofapps.com/marketplace/phonegap/>.

QUAIATO, V. *O que é Xamarin?* 2017. Visitado em Jun. 2018. Disponível em: <https://www.lambda3.com.br/2016/10/o-que-e-xamarin/>.

REACT. *React Native*. 2018. Visitado em Jun. 2018. Disponível em: <http://www.reactnative.com/>.

RIBEIRO, R. *Seja racional, vá de carona*. Mobilize Brasil, 2016. Visitado em Mar. 2018. Disponível em: <http://www.mobilize.org.br/noticias/9471/seja-racional-va-de-carona.html>.

SCHLICKMANN, M. P. *Alguns conceitos básicos do transporte urbano de passageiros*. 2013. Visitado em Apr. 2018. Disponível em: <https://transportacao.com/2013/12/28/4-alguns-conceitos-basicos-do-transporte-urbano-de-passageiros-1/>.

SIGNIFICADOS. *Definição de Rota*. 2015. Visitado em Apr. 2018. Disponível em: <https://www.significados.com.br/rota/>.

SZWARCFITER, J. *Grafos e Algoritmos Computacionais*. [S.l.]: Editora Campus, 1983.

Sá, L. C. *Existem mais de 5 milhões de aplicativos no mundo*. Infotec, 2017. Visitado em Fev. 2018. Disponível em: <http://www.infotecblog.com.br/2017/07/03/existem-mais-de-5-milhoes-de-aplicativos-no-mundo/>.

TAQTILE. *Híbrido vs Nativo*. 2016. Visitado em May. 2018. Disponível em: <https://medium.com/taqtilebr/h%C3%A9brido-vs-nativo-c8591df0dce6>.